

Copyright

by

David Ryan Ottesen

2022

The Dissertation Committee for David Ryan Ottesen
certifies that this is the approved version of the following dissertation:

**Spacecraft Trajectory Optimization using Many
Embedded Lambert Problems**

Committee:

Ryan P. Russell, Supervisor

Maruthi R. Akella

Brandon A. Jones

David Fridovich-Keil

Yanping Guo

**Spacecraft Trajectory Optimization using Many
Embedded Lambert Problems**

by

David Ryan Ottesen

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2022

We are made of stardust.

Acknowledgments

I would like to express my gratitude to the people in my life. Austin has been my home for a long time now. I have have grown as much as it has. I hope my journey has positively affected the people around me.

First and foremost, I would like to thank Dr. Ryan P. Russell, without him none of this work is possible. His invaluable technical instinct and breadth enabled my success. He listened to my ideas and gave me the space needed to pursue my research, all the while making sure my efforts were productive. He is a world-class advisor and researcher. I am a expert in astrodynamics because of him.

I would like to thank my mother, father, and sister, who supported me throughout my studies. Without them, I would not have made it through. I would like to thank my girlfriend Tracy who is a light in the dark and her family is dear to me: Greg, Colleen, and Ezra, as well as Kakii and Clancy. They have welcomed me without hesitation.

My many supportive friends over the years, both new and old, deserve praise too: Lee, Kat, and Jackie; Laura and James; Ana and Justin; Bonnie; Columbia; Sean; Sarah and Dave; Som and Carol; Vivek and Ashley; Etienne and Sam; Ash; Chun; Shubham; Eric; Ben; Arjun; Enrico and Courtney; Kristen; Erika; Vishnu; Trevor; Scott and Anne; Nick; Peter; Brandon; Pedro; Cory and Ashley; Joel and Raquel; Nikki; Krystal; Chris; Sonia and Ricardo; Jackie and Matt; and Emma and Cole. I hope to never lose touch with any of them. Life flies by.

Finally, I would also like to thank my committee for their time and effort in reviewing my dissertation and participating in my defense: Dr. Maruthi R. Akella, Dr. Brandon A. Jones, Dr. David Fridovich-Keil, and Dr. Yanping Guo.

DAVID RYAN OTTESEN

The University of Texas at Austin

July 2022

Spacecraft Trajectory Optimization using Many Embedded Lambert Problems

David Ryan Ottesen, Ph.D.
The University of Texas at Austin, 2022

Supervisor: Ryan P. Russell

Improvement of spacecraft trajectory optimization approaches, methods, and techniques is critical for better mission design. Preliminary low-fidelity analysis precedes high-fidelity analysis to efficiently explore the space of a problem. The work of this dissertation extends an embedded boundary value problem (EBVP) technique for preliminary design in the two-body problem. The EBVP technique is designed for direct, unconstrained optimization using many, short-arc, embedded Lambert problems that discretize the trajectory. The short arcs share terminal positions to implicitly enforce position continuity and the instantaneous velocity discontinuities in between segments are the control. These coasting arcs and impulsive maneuvers in between segments are defined collectively as a coast-impulse model, similar to the well-known Sims-Flanagan model.

Use of EBVPs is not new to spacecraft trajectory optimization, extensively used in primer vector theory, flyby-tour design, direct impulsive-maneuver optimization, and more. Lack of fast and accurate BVP solvers has prevented the use of the EBVP technique on problems with more than dozens of segments. For the two-body problem, a recently-developed Lambert solver, complete with the necessary partials, enables the extension of the EBVP technique to many hundreds to thousands of segments and hundreds of revolutions. The use of many short arcs guarantees existence and uniqueness for the Lambert problem of each segment. Furthermore, short arcs simultaneously approximate low thrust and eliminates the need to know the structure of a high-thrust impulsive-maneuver solution. A set of examples show the EBVP technique to be efficient, robust, and useful. In particular, an example using 256 revolutions, 6143 segments, and a constant flight time per segment, optimizes in 5.5 hours using a single processor.

After this initial demonstration, the EBVP technique is improved by a function which enables variable flight time per segment. Guided by the well-known Sundman transformation, these piecewise Sundman transformation (PST) functions divide the total flight time of the trajectory into spatially-even arcs, importantly not modifying the dynamics. Flight-time functions and their dynamical regularization counterpart are shown to share similar behavior for Keplerian orbit propagation. The PST functions are also shown to extend the EBVP technique to a large design space, where a runtime-feasible transfer with 512 revs and 12287 segments is presented that significantly changes semimajor axis, eccentricity, and inclination. Moreover, another example is presented that transfers through the numerically challenging parabolic boundary, i.e. a transfer from a circular to hyperbolic orbit. Both these examples use an exponent of $3/2$ for the PST to enforce the spatially-even arcs or equal steps in eccentric anomaly.

Lastly, an optimal control problem is formulated to solve a class of many-

revolution trajectories relevant to the EBVP technique. For transfers that minimize thrust-acceleration-squared, primer vector theory enables the mapping of direct, many-impulsive-maneuver trajectories to the indirect, continuous-thrust-acceleration equivalent. The mapping algorithm is independent of how the direct solution is obtained and the mapping computations only require a solver for a BVP and its partial derivatives. For the two-body problem, a Lambert solver is used. The mapping is simple because the impulsive maneuvers and co-states share the same linear space around an optimal trajectory. For numerical results, the direct coast-impulse solutions are demonstrated to converge to the indirect continuous solutions as the number of impulses and segments increase. The two-body design space is explored with a set of three many-revolution, many-segment examples changing semimajor axis, eccentricity, and inclination. The first two examples change either a small amount of semimajor axis or eccentricity, and the third example is a transfer to geosynchronous orbit. Using a single processor, the optimization runtime is seconds to minutes for revolution counts of 10 to 100, while on the order of one hour for examples with up to 500 revolutions. Any of these thrust-acceleration-squared solutions are good candidates to start a homotopy to a higher-fidelity minimization problem with practical constraints.

Contents

Acknowledgments	v
Abstract	vii
List of Tables	xiv
List of Figures	xv
Chapter 1 Introduction	1
1.1 Structure of Trajectory Optimization Problems	5
1.1.1 Historical Context	5
1.1.2 Trajectory Model	7
1.1.3 Formal Problem Statement	10
1.1.4 Solution	11
1.2 Outline	15
1.3 Summary of Contributions	16
Chapter 2 Spacecraft Trajectory Model Discretization	19
Chapter 3 Unconstrained Direct Optimization of Spacecraft Trajectories Using Many Embedded Lambert Problems	24
3.1 Introduction	24

3.2	Problem Formulation	32
3.3	The Many-Segment Method	37
3.4	Performance Results	41
3.4.1	Runtime vs. Number of Nodes	42
3.4.2	Parallelization Improvements: Speed Up vs. Threads	46
3.4.3	Search Direction Demonstration: Examples 1.a and 1.b	47
3.4.4	EBVP Solver Demonstration: Examples 1.c and 1.d	48
3.4.5	Initial Guess Demonstration: Examples 1.e and 1.f	49
3.4.6	Cost Homotopy Demonstration: Example 1.g	50
3.4.7	Large Number of Nodes Demonstration: Example 2	54
3.5	Conclusions	58
3.6	Appendix	60
3.6.1	Performance Tables for Examples 1 and 2	60

Chapter 4 Piecewise Sundman Transformation for Spacecraft Trajectory Optimization using Many Embedded Lambert Problems 63

4.1	Introduction	63
4.2	Piecewise Sundman Transformation	67
4.2.1	Differential Sundman Transformation	71
4.2.2	Density Functions	71
4.2.3	Differential of Eccentric Anomaly	73
4.3	Trajectory Propagation: Piecewise Functions vs. Differential Transformations	76
4.3.1	Regularization on Two-Body Dynamics	77
4.3.2	Propagation Results	78

4.4	Trajectory Optimization: Piecewise Sundman Transformation	82
4.4.1	Problem Formulation	84
4.4.2	Optimization Results	87
4.5	Conclusions	102
Chapter 5 Direct-to-Indirect Mapping for Optimal Low-Thrust Trajectories		104
5.1	Introduction	104
5.2	Models	111
5.3	Optimal Control Problems	112
5.3.1	Cost	113
5.3.2	Augmented Cost, First Differential, and Two-Point Boundary Value Problem	114
5.4	Direct-to-Indirect Mapping	118
5.5	Results	124
5.5.1	Results: Relative Error of the Direct Solution to the Indirect Solution	125
5.5.2	Results: Initial Guess Options for Indirect Problem	131
5.5.3	Results: Many-Rev, Multi-Shooting Problems	134
5.6	Conclusions	137
5.7	Appendix	139
Chapter 6 Conclusions		140
6.1	Future Work	142
Appendices		146

Appendix A Optimization Algorithms	147
Appendix B Sensitivities	152
B.1 Cost Partial and Differentials	152
B.1.1 Variational State Transition Matrix	160
B.1.2 Mass-Leak	161
Appendix C List of Publications	162
C.1 Journal Publications	162
C.2 Conference Proceedings	163
Bibliography	163
Vita	200

List of Tables

3.1 Overview of embedded boundary value problem solvers.	31
3.2 Example overview for comparisons of optimization characteristics. .	42
3.3 Performance of examples 1.a, 1.b, 1.c, and 1.d, comparing search direction and EBVP solver.	60
3.4 Performance of examples 1.e and 1.f for the initial guess demonstrations.	61
3.5 Performance of ex. 2 that demonstrates a continuation method for a large number of nodes.	62
4.1 Period in τ_p for Sundman exponents $\alpha = 0, 1, 3/2,$ and 2. The τ_p 's are plotted in Fig. 4.2.	76
4.2 Regularization scenarios for the forward propagation of an orbit. (\mathbf{a}, \mathbf{e}) = (5.0 DU, 0.5).	79
4.3 Segment count of initial and final rev of example 3 shown in Figs. 4.9 and 4.10.	97
4.4 Runtimes for the continuation method for the third sub-example of example 3 in Figs. 4.11 to 4.13.	99
B.1 Segment $i(i + 1)$ partials for \mathbf{r}_i and τ_i for the PST.	159
B.2 Auxiliary segment $i(i + 1)$ partials for α and r_* of the PST.	159
B.3 EBVP partials in terms of the variational STM	161

List of Figures

2.1	Two representative portions of a spacecraft trajectory are both discretized into two segments and three nodes. $r_a(t)$ (top) and $r_b(t)$ (middle) visualize an uncontrolled and controlled trajectory, respectively. The impulsive maneuvers for the control of $r_b(t)$ are the blue, upwards-pointing arrows. The axis (bottom), from left to right, points forward in time and illuminates the variable flight times, $\Delta t_{(i-1)i}$ and $\Delta t_{i(i+1)}$	20
2.2	Position \mathbf{r}_i differential for node i and its effect on the neighboring differentials of velocity (left side) as well as impulsive maneuvers (right side).	22
2.3	Visualization of the position discretization of a spacecraft trajectory for both the direct model (left) and indirect model (right).	23
3.1	Circular-orbit to circular-orbit optimal transfers. Solid and hollow blue circles are the nodes for the 20-node, and 2- or 3-node examples, respectively; blue lines are the integrated segments; the red lines are impulsive $\Delta \mathbf{v}$ maneuvers, and the gray lines are the 2D projections of the trajectory. Note for the middle sub-figure, the 11 nodes after the second impulsive $\Delta \mathbf{v}$ maneuver are on the final orbit, i.e. a coasting phase	39

3.2	Histories of Cartesian positions and impulsive $\Delta\mathbf{v}$ maneuvers for the two- and three-impulse examples (left and right, respectively)	39
3.3	FMINUNC _{UT} runtime of the cost and cost plus partials function calls (left), and runtime per iteration of the search direction as well as the combined cost and partials (right). Examples are plotted.	43
3.4	Speed up of three symmetric BLAS subroutines and the ivLam EBVP.	47
3.5	Cost vs. runtime for two examples using either the BFGS or HZ search direction for FMINUNC _{UT} . Total major iteration count is 209, 56336, 8821, and 1314720 for 1.a.i, 1.a.ii, 1.b.i, and 1.b.ii, respectively.	48
3.6	Initial guesses: min fuel, zero-rev: 1.a, 1.c, 1.e (top) and min energy, seven-rev: 1.b, 1.d, 1.f (bot). z -axis not to scale. Solid blue circles are the nodes. Blue lines connect the nodes, but are not the integrated segments. Red lines are impulsive $\Delta\mathbf{v}$ maneuvers.	51
3.7	Optimal solutions from the initial guesses of Fig. 3.6: min fuel, zero-rev: 1.a, 1.c, 1.e (left) and min energy, seven-rev: 1.b, 1.d, 1.f (right). z -axis not to scale. Solid blue circles are the nodes. Blue lines connect the nodes, but are not the integrated segments. Red lines are impulsive $\Delta\mathbf{v}$ maneuvers.	52
3.8	Optim. solutions: exs. 1.g.i and 1.g.ii from 1.g.iii. z -axis not to scale. Solid blue circles are the nodes; blue lines connected the nodes, but are not the integrated segments; the red lines are impulsive $\Delta\mathbf{v}$ maneuvers.	52
3.9	Mass and thrust history: 1.g.i, $k = 1.0$, $m_{120+} = 0.4695$ MU 1.g.ii, $k = 0.025$, $m_{120+} = 0.4427$ MU 1.b, 1.d, 1.f, 1.g.iii, $k = 0$, $m_{120+} = 0.4129$ MU. $c = 0.6325$ DU/TU for 1.g.	53

3.10	a and e history: 1.g.i, $k = 1.0$ 1.g.ii, $k = 0.025$ 1.b, 1.d, 1.f, 1.g.iii, $k = 0$	53
3.11	i, Ω , and ω history: 1.g.i, $k = 1.0$ 1.g.ii, $k = 0.025$ 1.b, 1.d, 1.f, 1.g.iii, $k = 0$	54
3.12	Optimal solutions: trajectories of select examples 2.a, 2.c, and 2.i of the large number of nodes problem for some of the stages of the continuation method. The blue-green lines are the trajectory. The grey lines are the 2D projections of the trajectory. The color gradient is consistent with the mass and thrust history of Fig. 3.13.	57
3.13	Thrust and mass history: example 2.i contains 6144 nodes and 256 revs, and uses $c = 0.6325$ DU/TU for a final mass $m_{6144+} = 0.3843$ MU. The smaller time domain of the left side reveals the smooth behavior of the thrust profile. The color gradient is consistent with the optimal solution in Fig. 3.12.	58
4.1	Discretization of a trajectory into segments bounded by nodes defined by either equal flight times (a,c) or equal ΔE (b,d) for a near-circular and eccentric Keplerian orbit.	65
4.2	τ_p vs. a and e: τ_p for varying a and e using the expressions in Table 4.1. For varying a (left), $e = 0.0$ for $\alpha = 3/2$ and 2. For varying e (right), $a = 1.0$ DU for $\alpha = 0, 1,$ and 2; and the curves for $\alpha = 0$ and 1 overlap.	76
4.3	ΔE vs. Segment Number Percentage: various differentials and their equivalent piecewise flight time functions for a Keplerian orbit. (a, e) = (5.0 DU, 0.5); (1 rev, $10\pi\sqrt{5}$ TU TOF).	81

4.4	Energy Error vs. Segment Number Percentage: various differentials and their equivalent piecewise flight time functions for a Keplerian orbit. $(\mathbf{a}, \mathbf{e}) = (5.0 \text{ DU}, 0.5)$; (1 rev, $10\pi\sqrt{5}$ TU TOF).	82
4.5	Example 1: ΔE vs. Segment Number Percentage: Discretization effect of three position coefficients r_* of the PST. $(\mathbf{a}, \mathbf{e}) = (5.0 \text{ DU}, 0.5)$; (24 segments, 1 rev, $10\pi\sqrt{5}$ TU TOF).	88
4.6	Example 2: Performance of the IVP and EBVP technique for an energy optimal transfer. Runtime, iteration count, and penalty coefficient updates are tracked vs. varying segment or rev counts. The particular example is illustrated in Figs. 4.7 and 4.8. The transfer is from $(1.0 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.6)$, using (\mathbf{a}, \mathbf{e})	92
4.7	Example 2: Discretization comparison of the IVP and EBVP technique for a 240-segment or 10-rev energy optimal solution. The two discretizations of the trajectory (left) and the plots (right) for ΔE vs. energy and eccentricity are nearly identical. Energy here is $\mathcal{E} = -\mu/2\mathbf{a}$. The transfer is defined by (\mathbf{a}, \mathbf{e}) : $(1.0 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.6)$ with a specified 211.0356 TU TOF.	92
4.8	Example 2: Cost and constraints vs. iteration count for a 240-segment or 10-rev, energy optimal solution using the IVP and EBVP technique. Penalty stages are labeled and magnifications (top) of the cost for the last three stages are included. The transfer is defined by (\mathbf{a}, \mathbf{e}) : $(1.0 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.6)$ with a specified 211.0356 TU TOF.	93

4.9	Example 3: Trajectories that primarily vary a only (top) and e only (bottom) for various PST exponents α at a fixed time of flight. $(a, e) = (1.0 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.0)$ [top], $(5.5 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.6)$ [bottom]; (240 segments [both], 10 revs [both], 220 TU TOF [top], 820 TU TOF [bottom]).	96
4.10	Example 3: Histories of ΔE , e , and \mathcal{E} for the spacecraft trajectories of the top row (left) and bottom row (right) of Fig. 4.9.	96
4.11	Example 3: Time-free, energy optimal spacecraft trajectory that significantly changes a , e , and i . $(a, e, i): (1.0 \text{ DU}, 0.0, 30^\circ)$ to $(5.5 \text{ DU}, 0.6, 0^\circ)$; (12288 nodes, 512 revs, 12814 TU TOF).	97
4.12	Example 3: History of ΔE and e vs. \mathcal{E} (left), as well as a , e , and i vs. t (right) for Fig. 4.11.	98
4.13	Example 3: History of T and m vs. t for Fig. 4.11. Note $m_{n+} = 0.353 \text{ MU}$	98
4.14	Example 4: J vs. $\Delta t_{\text{total},s}$: Energy optimal solutions with varying total flight times for 1 to 10 revs. The square is the energy optimal, plotted example in Figs. 4.15 to 4.17.	100
4.15	Example 4: Fuel (top) and energy (bottom) optimal spacecraft trajectories plotted with the $\Delta \mathbf{v}$ (red lines) for a circular to hyperbolic orbit transfer. The squares are at the initial time of a hyperbolic segment and the right side is a magnification. $(a, e): (1.0 \text{ DU}, 0.0)$ to $(-2.4142 \text{ DU}, 1.4142)$; (241 nodes, 5 revs, 200 TU TOF) [both].	101
4.16	Example 4: t (left) and node i (right) history for \mathcal{E} and e for energy and fuel optimal spacecraft trajectories in Fig. 4.15.	101

4.17	Example 4: t (left) and node i (right) history for the energy and fuel optimal spacecraft trajectories in Fig. 4.15. The left and right side show the evolution of T and m , as well as Δv and m , respectively. Note $m_{n+} = 0.542$ and 0.297 MU for $k = 1.0$ and 0.0 , respectively.	102
5.1	Hohmann-like transfer for a minimum thrust-acceleration-squared problem using a direct, coast-impulse model or indirect, continuous-thrust model.	106
5.2	Mapping for the state and co-state at node i from the direct (left) to the indirect (right) solution.	122
5.3	Indirect solution (left) and relative error of the direct solution for 6, 24, and 96 segments per rev (mid-left to right, respectively) of an optimal transfer defined by five revs and $(\mathbf{a}, \mathbf{e}) = (1.0 \text{ DU}, 0.0)$ to $(5.0 \text{ DU}, 0.5)$	127
5.4	Position, velocity, co-position, and co-velocity spaces showing the continuous indirect solution (left) and the integrated approximate augmented state $\mathbf{y}_{i,\text{I}}$ (mid-left, mid-right, right). Per Eqs. (5.37) to (5.40), each $\mathbf{y}_{i,\text{I}}$ from nodes 2 to $n - 1$ is propagated backwards to t_2 and forwards to t_{n-1}	128
5.5	Histogram of the r_x vs. r_y solutions at the middle node shown in Fig. 5.4 for the 6, 24, and 96 segs/rev cases.	129
5.6	Histogram of the v_x vs. v_y solutions at the middle node shown in Fig. 5.4 for the 6, 24, and 96 segs/rev cases.	129
5.7	Histogram of the λ_{r_x} vs. λ_{r_y} solutions at the middle node shown in Fig. 5.4 for the 6, 24, and 96 segs/rev cases.	130

5.8	Histogram of the λ_{v_x} vs. λ_{v_y} solutions at the middle node shown in Fig. 5.4 for the 6, 24, and 96 segs/rev cases.	130
5.9	RMS of ϵ_r , ϵ_v , ϵ_{λ_r} , and ϵ_{λ_v} , which are computed with direct solution approximations at nodes 2 to $n - 1$ and integrated from t_1 to t_n . . .	131
5.10	Initial guess options for the indirect problem, an optimal thrust-acceleration-squared orbital transfer using five revs and defined by $(\mathbf{a}, \mathbf{e}) = (1.0 \text{ DU}, 0.0)$ to $(5.0 \text{ DU}, 0.5)$. Three discretization scenarios: 6, 24, and 96 segs/rev are used.	133
5.11	Optimization runtimes for the indirect problem for three orbital transfers that use an initial guess derived from a 24 segs/rev direct solution.	135
5.12	Two optimal, time-free, 500-rev transfers defined by $(\mathbf{a}, \mathbf{e}) = (1.0 \text{ DU}, 0.0)$ to $(2.0 \text{ DU}, 0.0)$ [left] and $(1.5 \text{ DU}, 0.0)$ to $(1.5 \text{ DU}, 0.2)$ [right], respectively.	135
5.13	Optimal, time-free, 250-rev GTO-to-GSO transfer defined by $(\mathbf{a}, \mathbf{e}, i) = (3.8305 \text{ DU}, 0.7258, 28.5^\circ)$ to $(6.6107 \text{ DU}, 0.0, 0.0^\circ)$. Note m is for mass.	136

Chapter 1

Introduction

Earth is a small blue marble in an immense black sea of space. Starting with Sputnik in 1957 [1] as the first artificial satellite and Yuri Gagarin in 1961 [2] as the first person in space, humanity has sought to explore, use, and inhabit space. The design of a space mission must contend with a plethora of objectives and constraints. Depending on the stakeholders, the mission objective can be communication (Sputnik 1 [3]; Telstar [4]; Syncom-2 [5, 6]), reconnaissance (CORONA [7]; Zenit [8, 9]), GPS [10], exploratory science [11], Earth observation (GOES-R Series [12]), sustained habitation [13], and more. The space mission analysis and design process must successfully minimize financial cost while maximizing return by meeting objectives ([14], ch. 1). The process is iterative and contains many different facets. Broadly speaking, after the objective and constraints are acknowledged, a mission architecture is designed and requirements are defined. The requirements are then allocated to system elements of the overall mission. After this design phase, the system is built, flown, and supported until the end of its life cycle.

One critical system element of the mission architecture relevant to this thesis work is the spacecraft trajectory, profoundly affecting every part of mission development and operational support. The design of the trajectory must meet mission

objectives while minimizing cost, often time or fuel, both of which have financial impacts. In particular, the research for this thesis improves preliminary and low-fidelity spacecraft trajectory optimization, used early in the mission life cycle, to inform mission architecture. The preliminary embedded boundary value problem (EBVP) technique is described in detail throughout. Briefly, a trajectory is modeled as set of position-continuous segments, where each segment is a solution to Lambert’s problem [15, 16, 17, 18]. Lambert’s problem is a boundary value problem using two-body dynamics. Position continuity is enforced by reusing terminal positions of each segment and the velocity discontinuities between segments are the control to be minimized. The technique is fast, simple to implement, and robust to poor initial guesses. The technique is for locally optimal solutions, but some scenarios explore families of solutions.

Historically, several missions were enabled by trajectory design. Voyager 1 and 2’s trajectory ([19, 20, 21]; [11], pp. 137–145) was called the Grand Tour. Voyager 1 was able to visit Jupiter (March 5th, 1979), Saturn (November 12th, 1980), and Saturn’s moon Titan with gravity assists. Voyager 2 visited and used Jupiter (July 9th, 1979), Saturn (August 25th, 1981), Uranus (January 24th, 1986), and Neptune (August 25th, 1989) for their gravity-assists too. LCROSS (Lunar Crater Observation and Sensing Satellite), launched together with the LRO (Lunar Reconnaissance Orbiter) on June 18th, 2009, used a highly inclined 4-3 resonance to investigate ice water at lunar poles [22, 23, 24]. The 4-3 resonance trajectory means the Moon and LCROSS orbited Earth four and three times, respectively, before a kinetic impact of the upper stage into a lunar southern crater. A kinetic impactor had never been done before. In 2018, the Parker Solar Probe [25], designed and flown by Johns Hopkins Applied Physics Laboratory (JHUAPL) to study the Sun, was

able to dramatically improve the scientific value of the mission through trajectory design. The use of Venus instead of Jupiter for gravity-assists enabled 24 solar flybys instead of just one. Another JHUAPL mission, MESSENGER [26, 27, 28], a science mission to Mercury, was made feasible by trajectory design, at least in part. To successfully orbit Mercury, a significant amount of velocity was lost before the insertion burn, which followed, in order, one Earth flyby, two Venus flybys, and three Mercury flybys.

While not exclusively enabled by, complicated satellite constellations in low-Earth orbit is facilitated by trajectory design and optimization techniques for complete coverage of the Earth and fuel minimization [29]. Some constellations are for internet access (Iridium, OneWeb, Starlink, and Amazon Kuiper), Internet of Things (Hiber, Astrocast, Athena, and Myriota), Earth observation (Iceye, Rapid-Eye, Planet, and Capella Space), autonomous transportation (Pulsar, GeeSpace), and more.

Optimal trajectory design enables effective use of low-thrust electrical engines too. Previous approaches for high-thrust trajectory design do not apply because low-thrust trajectories gradually, instead of quickly, change velocity over relatively longer flight times. Designing trajectories that assume this gradual change in velocity is advantageous because it saves fuel. High-specific-impulse low-thrust electrical engines use less fuel than low-specific-impulse high-thrust chemical engines. More spacecraft mass can then be used for instrumentation. For example, launched on October 24th, 1998, Deep Space 1 ([30, 31]; [11], pp. 207–208) was the first spacecraft to use low-thrust ion propulsion instead of high-thrust chemical propulsion. Some other low-thrust missions include SMART-1, which tested Hall ion thrusters to transfer from Geosynchronous Transfer Orbit (GTO) to lunar orbit in 2003 [32].

The DAWN mission [33, 34], launched on September 27th, 2007, was the first NASA exploratory mission to use ion propulsion, enabling the exploration of two large asteroids: Vesta (July 16th, 2011 to September 5th, 2012) and Ceres (March 6th, 2015 to November 1st, 2018). Hayabusa 1 [35] and 2 [36] are both JAXA asteroid sample return missions using ion thrusters. Hayabusa 1 launched on May 9th, 2003 and the sample returned in June 13th, 2010. Hayabusa 2 launched on December 3th, 2014 and returned its sample on December 5th, 2020. OSIRIS-REx [37, 38, 39, 40] is a NASA asteroid sample-return mission to Bennu, launched on September 8th, 2016 and is expected to return its sample on September 24th, 2023. OSIRIS-REx's engines stand in contrast to the previous sample-return missions, using 28 Hydrazine monopropellant thrusters, where only two are low-thrust. The navigation of Bennu consisted of multiple phases: (1) an orbit to characterize the asteroid and select a landing site, as well as (2) touch-and-go maneuvers to the surface and back to collect regolith.

Finally, theoretical trajectory design problems push the state of the art of mission feasibility. New technical insight can find application in practical problems. The Global Trajectory Optimization Competition (GTOC) [41] is an event that takes place every one to two years, where teams across the globe have one month to solve a very difficult trajectory optimization problem. As of 2022, eleven competitions have taken place. Some of the most recent problems are (i) building a Dyson ring, inspired by the well-known Dyson sphere; (ii) settling the Milky Way in a quest 90 million years long; (iii) removing 123 pieces of space debris about Earth; and (iv) building a high-resolution, space-based, very-long-baseline interferometry system for 420 radio sources. One write up of a fourth-place finish in 2022 by the team from UT Austin, which includes the thesis author, is found in Ref. 42.

1.1 Structure of Trajectory Optimization Problems

To determine an optimal spacecraft trajectory to enable a mission, the problem structure typically has the following form, divided into three main parts: the trajectory model, the formal problem statement, and the solution approach. In general, after the model and problem statement is defined, the solution approach minimizes an objective while meeting constraints. The constraints take the form of satisfying ordinary differential equations, initial and final boundary conditions, continuity constraints, path constraints, and more. Solution forms include deterministic or stochastic, as well as analytical and numerical. In the following structure description, special emphasis is given to the preliminary EBVP technique.

1.1.1 Historical Context

Breakwell’s 1959 paper [43] on “The Optimization of Trajectories” presented the first general theorem to control thrust magnitude, angle of attack, and thrust direction of a rocket, while minimizing time as an objective. Lawden’s 1963 book “Optimal Trajectories for Space Navigation” [44] was the first dedicated book on the trajectory optimization of rockets, notably deriving the control law: the primer vector. Edelbaum’s extensive collection of work [45, 46, 47, 48, 49, 50] in the 1960’s and 70’s extended the spacecraft trajectory optimization field too, often focusing on optimal transfers assuming many revolutions (revs) and small orbital element changes, but not exclusively.

Many surveys have extensively covered the breadth and structure of spacecraft trajectory optimization and optimal control in general. The 1972 survey paper by Polak [51] covers optimal control algorithms of the last few preceding decades for continuous ordinary differential equations: gradient methods, Newton-Raphson

methods, conjugate gradient methods, feedback solutions (dynamic programming), iterative methods (indirect methods), penalty functions, and discretization. Bryson [52] in 1996 thoroughly reviews the historical roots of optimal control, starting with the development of the calculus of variations in the 17th century by Fermat (1601–1665), Galileo Galilei and the Brachistochrone problem of 1638 ([53], pp. 117-120), as well as some contributions from Newton, Bernoulli [54], Leibniz, and l’Hospital. Calculus of variations was further developed in the 18th century by Euler and Lagrange, in the 19th century by Legendre, Jacobi, Hamilton, and Weierstrass, and in the 20th century by Bolza, Bliss, Pontryagin, and Lawden. Bellman introduced us to the Hamilton-Jacobi-Bellman theory, or dynamic programming, in 1957. The authors von Stryk and Bulirsch [55] in 1992 divide optimal control problems into direct and indirect methods, suggesting a hybrid approach between the two to improve the convergence area of indirect methods with low-accuracy direct methods. Though this low accuracy of the direct method in this work is more a function of the resolution of the state and control. The higher the resolution of the discrete problem, the better it represents the true continuous solution. The Brachistochrone problem and the Apollo reentry problem are presented. A little later in 1998, Betts [56] also divided the spacecraft trajectory optimization field into direct and indirect methods, acknowledging overlap between the two. He expanded on the formal problem statement, the nonlinear program (NLP) for the solution, and the discretization and modeling approaches for both the direct and indirect methods. A decade later, Rao [57] in 2009 produced a complementary survey for computational optimal control. Rao covered similar topics as Betts, also stating a formal optimal control problem, numerical methods to model the dynamics, the Karush-Kuhn-Tucker [58, 59] (KKT) conditions for the NLP, optimization software, and a

particular focus on collocation and pseudospectral methods. A few years later in 2012, Conway’s [60] survey focuses on a qualitatively different approach: stochastic evolutionary or heuristic algorithms, instead of the deterministic approach that discretizes the continuous optimal control problem and converts it to an NLP problem. The survey by Shirazi et al. [61] in 2018 organizes well and elaborates on the spacecraft trajectory optimization field by way of models, objectives, approaches, and solutions. Shirazi’s survey includes updates on new stochastic methods and a section on state representation (Cartesian, polar, as well as classical and modified equinoctial orbital elements). In 2021, Malyuta et al. [62] produced a survey on optimization-based space vehicle control, focusing on convex optimization techniques. Lastly, the excellent survey by Morante et al. [63] on optimal low-thrust trajectories, includes the typical discussion on the objective function, continuous and discrete control and dynamics, and the methods and software tools used to solve the optimization problems. Notably, a few new topics were included, namely the hybrid nature of low-thrust techniques, differential dynamic programming [64, 65, 66, 67], and multi-objective optimization.

1.1.2 Trajectory Model

The trajectory model can be generally defined in terms of dynamics, state representation, discretization, and control modeling. Low-fidelity approximations are assumed first and improved later. Fast and robust low-fidelity modeling minimizes computational and human effort as well as encourages thorough exploration of a design space.

Popular simplified models for dynamics of the n -body problem are the two-body problem and three-body problem, where n could be every gravitational body

in the solar system. The two-body problem (one body plus a spacecraft) first assumes a force model that only includes a gravitating point-mass and mass-less, or mass-negligible, spacecraft. The work in this thesis uses this simplified two-body model. Additional two-body perturbations to improve fidelity include spherical harmonic perturbations such as J2, J3, etc.; third-body effects from the Moon or the Sun; atmospheric drag; and more. The simplified relative dynamics of the two-body problem use the Clohessy-Wiltshire (CW) [68] equations of motion. The CW equations assume close, linearized motion about a spacecraft in a circular orbit. The three-body problem (primary and secondary body plus spacecraft) includes two gravitational point-masses and neglects the spacecraft's mass. The simpler circular-restricted three-body problem (CR3BP) assumes the gravitating bodies are in circular orbits and the rotating frame is centered either at one of the bodies or at the collective center of mass. Increasing fidelity, the bodies could also be in elliptical orbits [69]. Hill's approximation [70] is classically a simplification of the CR3BP, where motion near either the primary or secondary body is perturbed by the other body. Hill's approximation models lunar and solar tides.

After the dynamical model is defined, the state of the spacecraft is represented with either Cartesian, polar, or spherical coordinates, as well as classical or modified equinoctial orbital elements. The rectangular Cartesian coordinates are the simplest to understand and do not suffer from singularities. Cartesian coordinates are used in this current thesis work. Polar and spherical coordinates are slightly more complicated relative to Cartesian coordinates, but more natural for circular motion about a body. For the classical orbital elements, five are straightforward descriptions of the size, shape, and orientation of a spacecraft's orbit. The sixth element places the spacecraft on the orbit. One drawback with classical orbital

elements is they suffer from singularities at eccentricity or inclination equal to zero [see Lagrange’s planetary equations ([71], sec. 12.3.3) or Gauss’ variational equations ([71], sec. 12.3.5)]. The modified equinoctial orbital elements ([72]; [73], sec. 6.3.1) solve this orbital element singularity, allowing for an orbit with eccentricity or inclination equal to zero, but trade some physical intuition with respect to the classical orbital element set.

The control for spacecraft trajectories is thrust from an engine, either for translational or rotational motion. The thrusting maneuvers are for short or long time spans and modeled as continuous or discrete. Classically, short duration maneuvers relative to the total flight time of a transfer are modeled approximately as instantaneous. In the two-body problem, long Keplerian coasting legs are bounded by these instantaneous impulsive maneuvers, tying a solution together to be position and velocity continuous. Some well-known optimal orbit transfers include the Hohmann transfer ([74], pp. 529–530; [75], ch. IV, pp. 76–89) and the bi-elliptic Hohmann transfer ([76], sec. 6.4, pp. 264–268). Impulsive maneuvers can also be generated by gravity-assists in a particular two-body system. For example, in the Solar System or the moon system of Jupiter or Saturn, the orbiting bodies supply a gravity assist by losing (or gaining) a negligible amount of momentum to speed up (or slow down) a spacecraft. The Sims-Flanagan model [77] was first to use many sequential impulsive maneuvers and short coasting segments to model approximate low-thrust. In this thesis work, the EBVP technique is inspired by this Sims-Flanagan model. Instead of using many solutions to Kepler’s problem to represent the segments in between impulsive maneuvers, many solutions to Lambert’s problem are used. This discrete, simple model for low-thrust approximates the continuous-thrust equivalent, similar to how a Riemann sum is approximately

equivalent to integration. If the control is not discrete, a continuous control law is used instead to smoothly change the state of a spacecraft on an orbit. The form of the control law can be optimal or sub-optimal, depending on the dynamics, and a function of thrust constraints.

1.1.3 Formal Problem Statement

After the trajectory model is defined, the formal problem statement, or optimal control problem, is defined. The goal is to determine the state $\mathbf{x}(t)$ and control $\mathbf{u}(t)$ history to minimize (or maximize) a cost, subject to the constraints (dynamical, path, and boundary). The cost function is otherwise known as an objective function, a merit function, or a performance index.

In general, the cost J is

$$J = \Phi_f(t_f, \mathbf{x}_f) + \int_{t_0}^{t_f} \mathcal{L}(t, \mathbf{x}(t), \mathbf{u}(t)) dt$$

where the final cost Φ_f is a function of the final time t_f and the final state $\mathbf{x}_f = \mathbf{x}_f(t_f)$. The variable \mathcal{L} is the integrand of the cost or Lagrangian. The cost J is subject to the ordinary differential equations, or equations of motion, \mathbf{f} as

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

where $\dot{\mathbf{x}}(t)$ is the time-derivative of $\mathbf{x}(t)$; the initial and final boundary conditions, respectively,

$$\Psi_0 = \mathbf{0}$$

$$\Psi_f = \mathbf{0}$$

and the path constraints

$$\mathbf{c}_{\min} \leq \mathbf{c}(t, \mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{c}_{\max}$$

where $\mathbf{c}(t, \mathbf{x}(t), \mathbf{u}(t))$ is some function bounded by \mathbf{c}_{\min} and \mathbf{c}_{\max} . For equality constraints, $\mathbf{c}_{\min} = \mathbf{c}_{\max}$, and for inequality constraints, $\mathbf{c}_{\min} < \mathbf{c}_{\max}$.

A vector cost function \mathbf{J} , also known as a multi-cost function, also exists to explore a space with two or more competing values, known as Pareto optimal solutions [78]. For example, if both the minimization of flight time and fuel is desirable, then many point solutions can be found with various flight times and fuel expenditures. The Pareto front of this solution space is the set of solutions where neither cost can improve without penalizing the other. The thesis work only looks at single-objective minimization of fuel and energy (see Chapters 3 and 4), and thrust-acceleration (see Chapter 5). Generally, the problems are unconstrained, where state continuity is embedded by design through use of many Lambert problems that reuse terminal position vectors. Though, for a few problems, a total flight-time constraint is included.

1.1.4 Solution

The solution phase is complex with many different approaches available. Influenced by the trajectory model and problem statement, the solution process can be categorized as either deterministic or stochastic. Deterministic approaches use predictable, gradient-based solutions to find an optimal or nearly-optimal solution. Stochastic approaches use random processes, typically without gradient information, to select, modify, and improve candidate solutions. A third approach is a hybrid between the two, where a stochastic algorithm might explore a global space, while the determinis-

tic algorithm refines a solution found in that space. The deterministic categorization can be further divided into analytical or numerical, and a more granular stochastic categorization is dependent on the particular technique. Both approaches are discussed next in further detail.

Deterministic

Deterministic analytical solutions typically trade fidelity for computational speed. After the initial derivation, which can be complicated, analytical solutions are known in closed-form without the need for numerical integration. However, analytical solutions are typically based on a set of restrictive assumptions, making them less general and only applicable to certain scenarios. Nevertheless, analytical solutions provide fundamental insight to a problem. For example, the early work of Edelbaum [45] in 1961 explored the space of impulsive-maneuver cost for different orbital element changes, assuming circular and quasi-circular orbits in the two-body problem. The classic Hohmann transfer ([74], pp. 529–530; [75], ch. IV, pp. 76–89) of the two-body problem is an analytical solution, solving for the optimal transfer between circular and co-planar orbits. Analytical solutions can also reformulate a problem such that the independent variable requires no iterative routine, improving speed. For example, if the independent variable is true anomaly ([79], eq. 2.5-1 and 2.5-4) or eccentric/hyperbolic anomaly ([71], the two-body problem has a closed-form solution. If the independent variable is time, an iterative root-solve must be performed, such as Kepler’s problem ([79], ch. 4) or Lambert’s (or Gauss) problem ([79], ch. 5). The optimal control problems in this thesis work use many solutions to Lambert’s problem, where recently a nearly closed-form Lambert solution exists [16, 17, 18], circumventing the need for a root-solve.

Numerical solutions are more general, robust to complexity, and often necessary if no closed-form solution exists. The optimal control problem must minimize a numerical approximation to the cost via discretized state and control that is subject to dynamics and constraints. This discrete representation is the NLP problem and the first-order optimality conditions of the NLP problem are known as the KKT [58, 59] conditions.

The optimal control problem can either be solved directly (direct method) by minimizing a cost and enforcing constraints with the NLP problem, or indirectly (indirect method) by forming an equivalent system that augments the cost with Lagrange multipliers to enforce constraints. This equivalent system of the indirect method is a two-point boundary value problem that enforces the necessary conditions from the first differential of the augmented cost. The necessary conditions are satisfied if the optimal control law minimizes the Hamiltonian, the Euler-Lagrange equations govern the time dependence of the Lagrange multipliers (or co-states), and the transversality conditions (boundary conditions) are met for the initial and final values of the Lagrange multipliers.

Direct solutions are often seen as more robust but with many more decision variables. Indirect solutions usually have fewer decision variables, but trade for increased sensitivity and co-states that are non-intuitive or non-physical. These direct and indirect approaches are usually coupled to one of two propagation methods: a shooting method [80] or collocation [81]. The shooting method integrates the state and control forward in time and continuity is enforced with an optimizer. A collocation method represents the state and control with sets of basis functions, where continuity of the state and control is typically embedded but dynamical feasibility must be enforced by an optimizer.

A third propagation approach for a direct method that connects the state and control to a cost function is the EBVP technique, the crux of this thesis work. The EBVP technique discretizes a trajectory with many solutions to EBVPs—namely Lambert’s problem for two-body dynamics, is always state and control continuous, and is always dynamically feasible. The corresponding NLP can be conveniently solved with an unconstrained optimizer. A fourth qualitatively different approach is differential dynamics programming [64, 65, 66, 67], a method that relies on the Hamilton-Jacobi-Bellman equation ([82], sec. 7.2) to successively minimize quadratic approximations of the problem, sweeping forward to compute cost and partial derivatives and backward to generate a new control sequence.

Stochastic

Stochastic or heuristic approaches generate solutions by simulating improvement found in nature coupled with a random process. All stochastic approaches must define the state of the system with a discrete set of parameters. This discrete set can represent smooth and continuous behavior. Some examples of stochastic approaches follow: (1) A genetic algorithm mimics evolutionary changes for a sample population. A fitness function is defined and solutions with good genes breed and pass on their traits with random mutations mixed in. (2) Simulated annealing [83] comes from annealing in metallurgy. At each step, the current state and a random neighboring state is compared, randomly choosing whether or not to move. The move does not necessarily improve the solution to enable bin hoping and thus exploration of the global space. (3) Particle swarm optimization [84, 85, 86] mimics the flight of birds. A set of particles or states are randomly dispersed in the state space. On each iteration, each particle advances as a function of a weighted average between three

values: (i) the current velocity of the particle, (ii) its own personal best solution, and (iii) the best solution of the population. There are many more stochastic approaches [87, 88, 89] available for the interested reader.

1.2 Outline

The dissertation is organized as follows. The introduction broadly covers the structure of spacecraft trajectory optimization problems, with an emphasis on the new EBVP technique. In Section 1.1, the spacecraft trajectory optimization discussion divides the solution process into three main steps: defining a trajectory model (Section 1.1.2), formalizing the problem statement in terms of a cost function and constraints (Section 1.1.3), and finally solving the problem with the many possible approaches, methods, and techniques available (Section 1.1.4). The introduction also covers this outline and an explicit summary of contributions in Section 1.2 and Section 1.3, respectively. The descriptions of the chapters are brief, emphasizing organizational structure and leaving the details of the contributions to Section 1.3 and their respective chapters.

After the introduction, the discretization for the spacecraft trajectory models, used throughout this work, is covered in Chapter 2. While short, the chapter is important for clarity. Chapter 3 presents the new direct EBVP technique, rooted in legacy literature, for preliminary low-fidelity spacecraft trajectory optimization in the two-body problem. Chapter 4 presents a piecewise function to automatically vary flight time for each of the segments of the trajectory model. The previous Chapter 3 uses the simple solution of equal flight time per segment, limiting application to transfers that do not significantly change orbital elements. Chapter 5 solves a multi-revolution, low-thrust optimal control problem that minimizes thrust-

acceleration-squared. The solution approach maps from a direct, discrete, coast-impulse solution to its indirect, continuous, finite-thrust equivalent. The EBVP technique is one source of the optimal direct solution, but the mapping of Chapter 5 does not depend on the EBVP technique. The conclusions are in Chapter 6, summarizing the historical significance, derivation, and application of the EBVP technique.

1.3 Summary of Contributions

The major contributions to the spacecraft trajectory optimization field are summarized in this section. The contributions span the performance and feasibility of the EBVP technique, improvements to trajectory discretization, and a mapping procedure from a direct to indirect method.

- Chapter 3: The main contribution of this body of work is the improvement and demonstration of the EBVP technique for low-fidelity spacecraft trajectory optimization problems. The EBVP technique uses many solutions to Lambert’s problem per trajectory, similar to the Sims-Flanagan model which uses many solutions to Kepler’s problem. The EBVP technique is shown to be fast and robust. Many trajectory optimization problems use solutions to boundary value problems, but this dissertation research extends the EBVP technique to use up to thousands of solutions per trajectory for an optimal control problem.
- Chapter 3: A comparison of “the many-segment method” vs. “the fewer-segment method” for high-thrust impulsive-maneuver trajectory optimization. As the names imply, the fewer-segment method uses only as many segments as

needed and the many-segment method uses too many intentionally-short segments. The many-segment method demonstrates utility in finding a solution without knowing optimal structure.

- Chapter 4: The piecewise Sundman transformation (PST) is defined and applied to spacecraft trajectory propagation and optimization in the two-body problem. Importantly, this application of the PST is shown to bestow regularization benefits without modification to the dynamics.
- Chapter 4: The PST enables a comparison to the state-of-the-art Sims-Flanagan model. The EBVP technique, using PST flight time functions, is faster after 72 short segments per a numerical experiment. High segment counts can be used for trajectories with lower-resolution thrust profiles but with more revolutions, or higher-resolution thrust profiles with less revolutions. Some two-body examples of lower-rev solutions include gravity-assist problems in the Solar, Saturn, or Jupiter systems; as well as rendezvous and sample-return asteroid missions. These lower-rev examples can use high- or low-thrust. Examples of higher-rev solutions include orbit transfers around a singular gravitating body, such as a transfer from low-Earth orbit to geosynchronous orbit.
- Chapter 4: The PST also enables transfers that significantly change semimajor axis and eccentricity. These transfers use an exponent of $3/2$ for the PST to ensure spatially even segments, i.e. approximate equal changes in eccentric anomaly per segment.
- Chapter 5: Many-rev, low-thrust optimal control problems are solved via a simple mapping procedure that uses primer vector theory and Lambert's problem. The mapping connects the direct and indirect solution methods for a

class of trajectory optimization problems that minimize thrust-acceleration-squared. The direct method is discrete, using the coast-impulse model, and the indirect method uses continuous finite thrust. Importantly, the mapping is indifferent to the source of the direct model, though the EBVP technique is used. The mapping is enabled by the thrusting vectors and the co-states sharing the same linear space around the optimal solution.

- Chapter 5: A cost function is derived and demonstrated that approximates the finite sum of thrust-acceleration-squared. This finite sum is not the same as the well-known finite sum for energy, otherwise known as the sum of the square of impulsive maneuvers.
- Chapter 5: The last contribution is the optimization of low-thrust transfers with up to 500 revolutions in Cartesian coordinates; similar problems typically use orbital elements. The multi-shooting solution method, coupled to the direct-to-indirect mapping procedure, is applied to three transfers that significantly change semimajor axis, eccentricity, and inclination.

Chapter 2

Spacecraft Trajectory Model

Discretization

In this thesis work, the spacecraft trajectories are simply modeled using Keplerian two-body dynamics. Chapters 3 to 5 all use a “direct trajectory model”, discretizing the trajectory into many short segments bounded by nodes. Chapter 5 groups these segments into sets of legs, forming an “indirect trajectory model”. These names are often shortened to “direct model” and “indirect model”, respectively. The direct trajectory model is more generally known as the coast-impulse model, where the segments are coasting arcs and the impulsive maneuvers are at the nodes.

The direct trajectory model takes the transcription form of an initial value problem (IVP) or an EBVP technique. The IVP technique uses a forward-shooting method coupled to a Kepler solver for many sequential segments and impulsive maneuvers. If the segments use a solution to a Lambert problem instead, the transcription is known as the EBVP technique. Recently, Lambert’s problem is *completely* solved with a practically closed-form solution [16, 17, 18]. This Lambert solution is leveraged to benefit performance of the optimal control problems in this body of work.

Fig. 2.1 represents a sub-section of the overall direct trajectory model with and without impulsive maneuvers (bottom and top, respectively), otherwise known as controlled and uncontrolled. The following state values at each node i are known: the values of time, position, velocity, acceleration, and mass. The impulsive maneuvers, $\Delta \mathbf{v}_i$, serve as the low-fidelity thrusting control. This particular portion of a trajectory is discretized into three nodes and two segments. The entire trajectory is numbered sequentially with $n - 1$ segments or n nodes. Segments $(i - 1)i$ or $i(i + 1)$ have a start and end denoted by $i - 1$ to i or i to $i + 1$, respectively. The segment subscripts should be read left to right for the forward direction in time. Node i has a before and after denoted as $i-$ and $i+$ as a subscript, respectively. Put together, segment $(i - 1)i$ is from node $(i - 1)+$ to $i-$ and segment $i(i + 1)$ is from node $i+$ to $(i + 1)-$. Moreover, since position is assumed continuous at node i ,

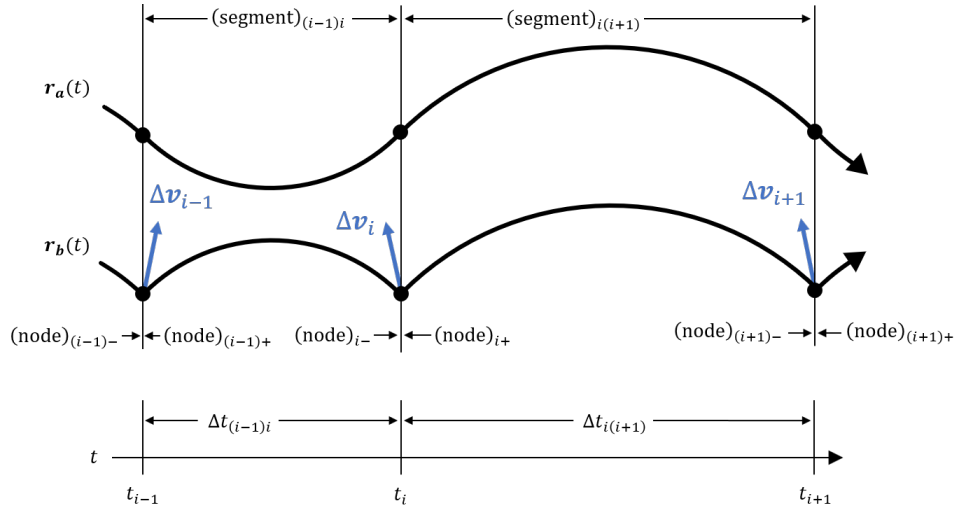


Figure 2.1: Two representative portions of a spacecraft trajectory are both discretized into two segments and three nodes. $r_a(t)$ (top) and $r_b(t)$ (middle) visualize an uncontrolled and controlled trajectory, respectively. The impulsive maneuvers for the control of $r_b(t)$ are the blue, upwards-pointing arrows. The axis (bottom), from left to right, points forward in time and illuminates the variable flight times, $\Delta t_{(i-1)i}$ and $\Delta t_{i(i+1)}$.

the position is denoted by $\mathbf{r}_i = \mathbf{r}_{i-} = \mathbf{r}_{i+}$. The value \mathbf{r}_i is simultaneously the final position of the previous segment $(i-1)i$ and the initial position of the next segment $i(i+1)$. Likewise, two-body acceleration is continuous because it is only a function of position and system constants: $\mathbf{a}_i = \mathbf{a}_{i-} = \mathbf{a}_{i+}$. In contrast, the velocity can discontinuously change at node i due to an impulsive maneuver, written as $\mathbf{v}_{i-} + \Delta\mathbf{v}_i = \mathbf{v}_{i+}$. Finally, node n differs for the EBVP and IVP technique in this thesis work. The EBVP technique does not need to enforce position or velocity continuity anywhere by design, but the IVP technique does. Node n of the IVP technique is not position or velocity continuous at the final state unless enforced through an optimizer. More details on the IVP technique are reserved for example 2 of Section 4.4.2 where it is used.

The illustration in Fig. 2.2 is a position update of a section of the direct trajectory model. Consider the independent position \mathbf{r}_i , the primary decision variable of this model, shown in Fig. 2.2, and the dependent velocities: $\mathbf{v}_{(i-1)+}$, \mathbf{v}_{i-} , \mathbf{v}_{i+} , and $\mathbf{v}_{(i+1)-}$. Fig. 2.2 shows the effect of the position update on the velocity vectors of node i . The plain and new superscript \square and \square^{new} is the previous and updated position or velocity, respectively. The solid lines —, the dashed lines - -, and the dotted lines \cdots , are the previous trajectory, the updated trajectory, and the differentials between the previous and updated trajectories, respectively. The velocity vectors are visually consistent in both the left and right sub-figures of Fig. 2.2.

The next discretization explanation is for Chapter 5, where a mapping is described and analyzed from a direct model, such as the EBVP technique in Chapters 3 and 4, to an indirect model. Importantly, though, the mapping in Chapter 5 is indifferent to the technique of the direct model. An IVP technique could be used. Thus, the direct and indirect trajectory models of Chapter 5 are distinguished by

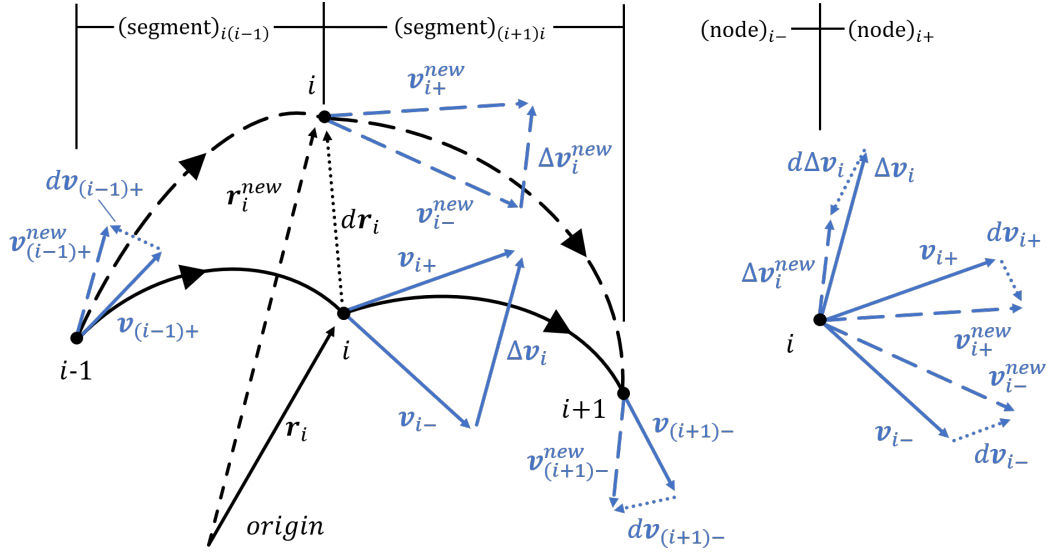


Figure 2.2: Position \mathbf{r}_i differential for node i and its effect on the neighboring differentials of velocity (left side) as well as impulsive maneuvers (right side).

the subscripts \mathbb{D} and \mathbb{I} , respectively. The direct model is still assumed to be discretized into $n - 1$ segments and n nodes, and the indirect model is discretized into $m - 1$ legs and $2m - 1$ nodes. The direct model nodes can be “continuation” or “decision state” nodes, where continuity is enforced at continuation nodes and an optimizer updates the state and control at decision-state nodes. Likewise, indirect nodes are categorized into m continuation nodes and $m - 1$ decision-state nodes, so $2m - 1$ in total. Each direct node is associated with a time, precisely notated by $t_{i,\mathbb{D}}$, where $i \in [1, n]$. For the indirect model, each continuation or decision-state node is associated with a time $t_{j,\mathbb{I}}$ or time $t_{k,\mathbb{I}}$, respectively, where $j \in [1, m]$ and $k \in [1, m - 1]$. For convenience, each indirect leg represents a group of direct segments with no overlap for a single- or multi-shooting method. Similar to segments of the direct model, legs are defined by their nodes. For example, leg $j(j + 1)$ is from continuation node j to $j + 1$ with a decision node k somewhere in between, such that $t_{j,\mathbb{I}} \leq t_{k,\mathbb{I}} \leq t_{j+1,\mathbb{I}}$. Accordingly, if relevant, a variable at $t_{i,\mathbb{D}}$, $t_{j,\mathbb{I}}$, or

$t_{k,\mathbb{I}}$ is augmented by the subscripts i, j , or k . Likewise, if a variable is from $t_{i,\mathbb{D}}$ to $t_{i+1,\mathbb{D}}$ or $t_{j,\mathbb{I}}$ to $t_{j+1,\mathbb{I}}$, then the subscripts $i(i+1)$ or $j(j+1)$, respectively, are used. If a variable does not include an index, i, j , or k , continuous time should be assumed instead of a discrete time associated with a node. For convenience, Fig. 2.3 shows both the direct model (left) [also shown in more detail in Figs. 2.1 and 2.2] and indirect model (right) for position vs. time. Notice the index labeling of the nodes, segments, and legs. The right side of Fig. 2.3 shows both types of nodes, emphasizing the discontinuity at continuity node j that must be eliminated by an optimizer. This right subfigure shows two legs for a multi-shooting method, but a single-shooting method would only need one leg.

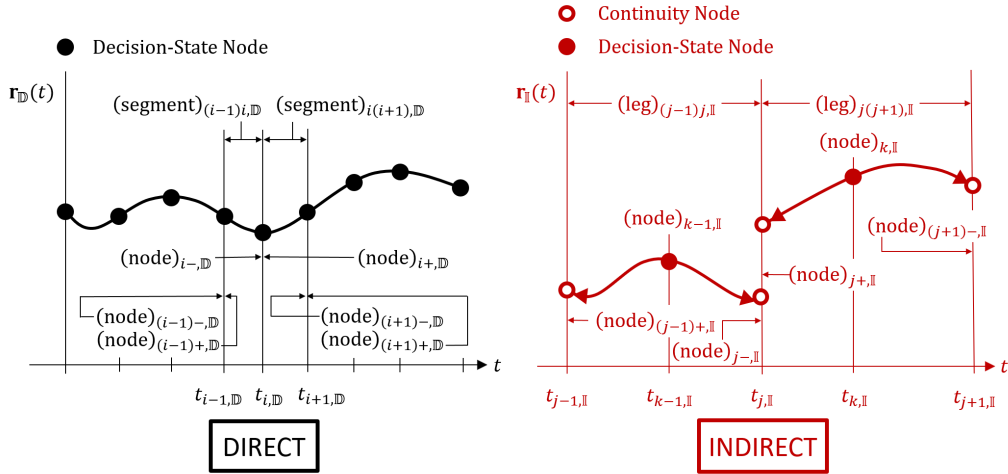


Figure 2.3: Visualization of the position discretization of a spacecraft trajectory for both the direct model (left) and indirect model (right).

Chapter 3

Unconstrained Direct Optimization of Spacecraft Trajectories Using Many Embedded Lambert Problems

3.1 Introduction

Spacecraft trajectory optimization is formulated in terms of models, objectives, approaches, and solution methods conveniently defined by Shirazi et al.[61]. The numerical approach considered here is a technique that uses a sequence of embedded boundary value problems (EBVPs), primarily Lambert’s problem, to discretize a trajectory into *many* segments. Position continuity is implicitly enforced by reusing terminal positions between neighboring EBVPs, while velocity continuity is provided through the addition of an impulsive $\Delta\mathbf{v}$ maneuver. The advancement of this EBVP technique to include up to thousands of segments for lower-fidelity, preliminary spacecraft trajectory optimization is the main contribution of this current work. The implementation of the EBVP technique is simple, using a two-body model, unconstrained optimization, and a Cartesian coordinate frame. Advancing

the performance and utility of simpler techniques common in practice facilitates design. Moreover, the implementation of this EBVP technique uses many short-arc segments, a defining and differentiating characteristic. For a specified direction of motion, short segments ensure a unique and smoothly varying solution to the EBVPs for all transfer angles between 0 and 180°. For transfer angles at the bounds of 0 and 180°, solutions encounter singularities; and for angles greater than 360°, the solutions may not exist or have up to two solutions. Furthermore, for high-thrust modeling, using many short segments also helps model minimum fuel solutions without having to guess the number of impulsive $\Delta\mathbf{v}$ maneuvers or switching structure if constraints on the thrust are considered. For low-thrust modeling, using many short segments with many impulsive $\Delta\mathbf{v}$ maneuvers helps model minimum energy solutions, analogous to the Sims-Flanagan model[90] proposed in 1999.

Lower-fidelity, impulsive $\Delta\mathbf{v}$ maneuver models are fast, robust, and useful, particularly when implemented with an EBVP technique. The motivation of this current work is to supplement, not replace, existing techniques for higher-fidelity spacecraft trajectory optimization, such as multi-shooting [91, 92], collocation [57], or differential dynamic programming [64, 65, 66, 93]. Additionally, other parameterizations exist that can enhance the EBVP technique further such as cylindrical or spherical coordinates. Cartesian coordinates are used in this current work for simplicity and to help mitigate singularities. Orbital elements are useful too, often necessary, for many-revolution (many-rev) solutions, particularly when coupled with orbital averaging techniques [94, 95, 96, 97, 98]. The EBVP technique described here is an alternative, direct, unaveraged approach that can be scaled up to a few hundred revolutions. This scalability represents a significant improvement over existing direct unaveraged methods, where the curse of dimensionality typically

limits such methods to a maximum of a few dozen revs at best. Furthermore, these simple models for two-body dynamics, patched by impulsive $\Delta\mathbf{v}$ maneuvers, can be tied together in phases to serve as an initial guess for higher-fidelity problems [99, 100, 101, 102, 103, 104].

The enabling technology for successfully scaling up the number of segments for this technique is a predictably fast, provably reliable solution to the EBVPs. For two-body dynamics, if the flight time is short and the angular displacement of the transfer is less than a half-revolution, then the classic Lambert problem solution is guaranteed to have a unique solution. While many fast Lambert solvers exist, the current approach uses a recently developed interpolation method with a fixed single iteration for the full domain of interest, is faster than all published approaches, and is provided with cheap to compute, analytic expressions for the first-order partial derivatives (partials), necessary for gradient-based optimization [16, 105]. Using the EBVP technique for optimization problems with a large number of segments has typically been avoided in the past because of a steep computational cost and robustness issues. An EBVP for general dynamics requires a numerical iterative routine to enforce either: dynamical constraints for collocation (see Conway's [60] or Jones and Anderson's [106] surveys), or position continuity for shooting methods [99, 107]. Successful optimization of the overall trajectory of many segments is reliant on successful solutions to the EBVPs for every segment at every optimization iteration. The numerical solutions to many EBVPs at every optimization iteration are computationally intractable if the EBVP solutions are not guaranteed to exist, convergence is unstable (see Ch. 3-5 of [108]), or if there is an unpredictable number of iterations.

Historically, the EBVP numerical technique was first used within an indirect

method, also known as primer vector theory, by Lion and Handelsman [109] or Jezewski [110, 111, 112] in 1968-1980, or used in a direct [56] method by D’Amario [113, 114] in 1979 and 1981. D’Amario’s work [113, 114] did not explicitly mention primer vector theory. Both methods optimize with positions and times only in two-body problems that include realistic considerations. For example, Jezewski’s work in [110] included an Apollo application for a rendezvous problem with a large plane change and D’Amario’s work in [114] included optimized Galileo satellite tours containing up to eleven flybys. Today, primer vector theory coupled to EBVPs is routinely performed. Often the problems include on the order of dozens of EBVP segments. For a two-body problem that uses the Lambert problem, Landau [115] in 2018 developed an efficient maneuver algorithm to minimize the maneuver design space for many-rev problems using primer vector theory. The presented examples include: Jovian and Saturn moon tours that minimize $\Delta\mathbf{v}$ and time, as well as a broken plane transfer from Earth to Mars. In particular for these problems, the highest segment count was for the two Saturn tours, containing 54 and 36 flybys for the minimum $\Delta\mathbf{v}$ and time solutions, respectively. For both tours, impulsive $\Delta\mathbf{v}$ maneuvers are efficiently included between flybys, per primer vector theory, so the number of conic arcs is slightly more than 54 and 36, respectively. For three-body problems, some examples include: the work by Davis et al. [116] in 2011 or Bokelmann and Russell [117] in 2019 using at most four and five segments, respectively, in the circular restricted three-body problem; and the work by Hiday-Johnston and Howell [118] in 1994 using at most three segments in the elliptical restricted three-body problem [69].

Other examples that do not use primer vector theory but use EBVPs include: (i) The optimization work of Marchand et al. [119] in 2007 using about a

dozen segments for an n -body to ephemeris model to find ballistic or near-ballistic solutions for approximate periodic orbits. (ii) In 2008, Di Lizia et al. [120] used a differential algebraic technique to expand a two-point BVP solution in different dynamical systems using higher-order Taylor polynomials. (iii) And, in the case of multiple gravity-assist trajectories, a preliminary design can tie together flyby encounters with solutions to Lambert’s problem. Some well-known historical missions are well-modeled this way: Voyager 1 and 2 [19], launched in 1977 to study the outer solar system; Galileo [121], launched in 1989 as the first spacecraft to orbit Jupiter; Cassini-Huygens [122, 123, 124] in 2004–2017, studying Saturn and its moons; and more recently, MESSENGER [27] in 2004–2015, studying Mercury. Among these missions, MESSENGER’s flight plan contained the most interplanetary gravity assists. MESSENGER’s flight plan from (E)arth to (M)ercury includes (V)enus and is compactly written from start to finish as EEVVMMMM with five major deep space maneuvers. Galileo’s moon tour nominally contained ten revs, using moon flybys to minimize fuel. Ultimately, 34 revs were achieved. Cassini-Huygens’ moon tour had 293 revs total, but in three sequential stages. The primary mission included 75 revs, and the two mission extensions included, in order, 64 and 154 revs, both designed in sequential phases [125, 126] that are driven by science objectives. Each phase contained on the order of dozens of flybys. Finally, set for a launch in 2024, the Europa Clipper mission [127] will contain 45 Europa, 5 Ganymede, and 9 Callisto flybys over the course of 3.5 years. The mission will contain five distinct phases for full coverage of Europa.

Some theoretical, many-rev examples using many Lambert solutions are as follows: (i) In 2006, the ephemeris model, Mars cycler problem is solved in Ref. [128], where up to 49 Lambert segments are simultaneously optimized. (ii) Izzo et al. [129]

in 2007 described a heuristic global optimization algorithm to search among tens of thousands of solutions for interplanetary problems where the final test cases are on the order of several segments. (iii) Abdelkhalik and Mortari’s [130] work in 2007 used a Sims-Flanagan model and one Lambert solution for the last segment. (iv) Vasile et al. [131] in 2008 worked on globally optimized, gravity-assisted, two-body problems, defining D’Amario’s technique as a ‘position formulation’. The presented problems had a few to several segments. (v) In 2014, the Jovian moon tour design by Lynam [132] contains two Lambert calls, but searches a large parameter space [133] to align Callisto, Ganymede, and Io for a Jovian orbit capture. Thus, the state of the art and practice for many-rev, interplanetary and moon tour design is to use on the order of dozens of segments separated by impulsive $\Delta\mathbf{v}$ maneuvers and gravity-assists. For larger many-rev problems, trajectories are sequentially stitched together in phases, and the resultant ten to hundreds of segments are not simultaneously optimized [126]. This current work demonstrates the simultaneous optimization of a many-rev solution with thousands of segments.

The EBVP technique also shares important characteristics with other methods within spacecraft trajectory optimization field. In this current work, position continuity is implicitly enforced when using many Lambert solutions. Interpolation can also embed position continuity, among other constraints, as shown by Mortari [134] in 2017. This ‘Theory of Connections’ interpolation approach was applied to an energy optimal spacecraft guidance problem by Furfaro and Mortari [135] in 2020. More generally, collocation embeds constraints at the expense of relaxing others. For collocation, dynamics are treated as constraints and state continuity can be embedded, such as position continuity. Each iteration of a collocation method is then position continuous but not necessarily dynamically feasible until convergence.

Shooting methods contrast with collocation in that shooting methods are always dynamically feasible, but not necessarily position continuous until convergence.

Shape-based methods are similar to the EBVP technique as well. Shape-based methods produce an independent path and a dependent control to satisfy this path. For this current work, the trajectory—as a collection of segments, shares this similar characteristic. The impulsive $\Delta\mathbf{v}$ maneuvers are dependent on the independent nodes which define the path. Many functions exist to shape a spacecraft trajectory including: a logarithmic spiral [136] (the first shape-based method), a Fourier series [137, 138, 139], a six-coefficient inverted polynomial [140], an exponential sinusoid [141, 142], a three-dimensional spherical shaping method [143] using an inverted polynomial and sinusoids, a family of generalized logarithmic spirals [144], linear plus sinusoid and exponential shaping for pseudo-equinoctial elements [145], and more [146]. The coefficients of these shape-based functions are optimized to satisfy boundary conditions, and minimize a cost function for the trajectory. Whereas in this current work, the positions of the EBVP technique are directly optimized. This inverted dependency is also known as ‘inverse dynamics’ in robotics [147], spacecraft attitude dynamics [148], and flight dynamics [149, 150], to name a few examples.

As mentioned, the enabling EBVP solver for this current work is the Lambert EBVP solver [16] (ivLam [see Table 3.1]) that includes fast and accurate partials by the second author. Some other Lambert problem solvers include Battin’s [151], Gooding’s [152], Izzo’s [153], or Arora and Russell’s [154]. Russell’s Lambert solver is ideal for the current application because the solution has a fixed single iteration, leading to predictable runtimes and simplified partials. The other EBVP solvers considered in the present work are simple Newton-Raphson shooting methods (rk7sh

and rk78sh [see Table 3.1]), notably requiring iterations to keep position continuous during optimization. The rk78sh EBVP solver uses a variable-step, 7(8)th order, Runge-Kutta integrator while the rk7sh EBVP solver uses a fixed-step, 7th order, Runge-Kutta integrator. The variable-step rk78sh enables modeling of perturbed two-body dynamics, but comes at the expense of much slower runtimes. This compute cost simultaneously serves as a reference on performance degradation that has historically prevented the widespread adoption of the EBVP technique for problems with more than a few dozen segments. Each integrated EBVP solver (rk7sh or rk78sh) requires the computation of the variational state transition matrix (STM) to root-solve on the initial velocity in a shooting method. Thus, for the ordinary differential equation (ODE) system of rk7sh or rk78sh, the propagated state is length 6 and the variational STM is an additional 36 components. On the other hand, Lambert solvers root-solve a single scalar equation, and notably for ivLam, only a fixed single iteration is needed, practically leading to a closed-form solution.

Table 3.1: Overview of embedded boundary value problem solvers.

Name	Description	Dynamics	# of Iter. Variables	# of Iterations	Dim. of ODE Sys.
rk78sh	Runge-Kutta 7(8), Variable-Step, First-Order Shooting Method	General	3	Varies	6 + 36
rk7sh	Runge-Kutta 7, Fixed-Step, First-Order Shooting Method	General	3	Varies	6 + 36
ivLam	Interpolated, Practically Closed-Form Lambert Solver [16]	Two-Body	1 ^a	1 ^a	n/a

^aTypical Lambert BVP solvers iterate on one variable, and notably, Gooding's[152] method uses three fixed iterations.

Overall, the current work presents the successful advancement of the EBVP technique to include up to several thousand segments, scoped for two-body spacecraft trajectory problems. Furthermore, the transfers in the current study are re-

stricted to low eccentricity. Previous work by the authors used many EBVPs with a constrained optimizer for a proof-of-concept [155], then another work demonstrated regularizing time-of-flight functions, termed the piecewise-constant Sundman transformation [156] that is especially useful for more eccentric trajectory design. In other words, low eccentricity transfers are only presented within this current work for simplicity, not because of a limitation inherent to the EBVP technique.

The chapter is outlined as follows. In Section 4.1, the EBVP technique as it relates to legacy literature was just described. In Section 3.2, an optimal control problem is formulated for fuel and energy optimal spacecraft trajectories modeled with impulsive $\Delta\mathbf{v}$ maneuvers. In Section 3.3, the trajectory discretization of using many short segments is described and termed the many-segment method. Solutions using the many-segment method are described, relative and equivalent to, solutions with fewer segments. The many-segment method is one utility justification for scaling up the EBVP technique. In Section 3.4, the numerical performance of an unconstrained optimization algorithm using EBVPs is demonstrated and analyzed. The examples minimize fuel or energy for local solutions of fixed-state to fixed-state, or orbit-to-orbit scenarios. In Section 4.5, the many-segment EBVP technique is summarized. Finally in Section 3.6, the appendix includes an algorithm for the optimizer, an algorithm for the search direction, and a discussion on partials for the EBVP technique.

3.2 Problem Formulation

The overview of the problem is

$$\begin{array}{llll}
\text{minimize} & \text{cost function} & J = k \sum_{i=1}^n \Delta v_i + (1 - k) \sum_{i=1}^n \frac{1}{2} \Delta v_i^2 & a \\
\Delta v_i(\mathbf{x}_d) & & & \\
\text{subject to} & \text{dynamics} & \mathbf{a} = -\frac{\mu}{r^3} \mathbf{r} & b \\
& \text{boundary conditions} & \mathbf{x}_{1-} = \mathbf{x}_{1-,s}, \mathbf{x}_{n+} = \mathbf{x}_{n+,s} & c \\
& & \mathbf{x}_{1-} = \mathbf{x}_{1-,s}(\tilde{t}_{1-}), \mathbf{x}_{n+} = \mathbf{x}_{n+,s}(\tilde{t}_{n+}) & d \\
& \text{path constraints} & \Delta \mathbf{r}_i = \mathbf{r}_{i+} - \mathbf{r}_{i-} = \mathbf{0} & e
\end{array}$$

^afuel plus energy | ^btwo-body | ^cfixed-state to fixed-state | ^dorbit to orbit | ^eposition continuity

In this section, the formulation of an optimal control problem for unconstrained spacecraft trajectory optimization using many EBVPs is described in detail. The cost function of the optimal control problem minimizes fuel or energy shown in Eq. (3.1), predominately using time and position decision variables as shown in Eq. (4.13). The dynamics are two-body for simplicity, though perturbations are feasible to employ. The state \mathbf{x} is length six containing Cartesian position and velocity shown in Eq. (3.2). The boundary conditions for the trajectory using n nodes or $n - 1$ segments goes from a fixed-state to fixed-state or orbit-to-orbit. The fixed-state can be implemented as a fixed-point on an orbit. The path constraint, position continuity, is not explicitly enforced by the optimizer, but by design through the use of EBVPs. Finally, mass continuity and approximate thrust are computed after optimality because the cost is a function of neither.

The cost function J is

$$J = k \sum_{i=1}^n \Delta v_i + (1 - k) \sum_{i=1}^n \frac{1}{2} \Delta v_i^2 \quad (3.1)$$

where Δv_i is the magnitude of an impulsive $\Delta \mathbf{v}$ maneuver at node i , and k is a homotopy variable between 0 and 1 to emulate minimum fuel ($k = 1$), minimum energy ($k = 0$), or a combination of the two ($0 < k < 1$). The impulsive $\Delta \mathbf{v}$ maneuvers and their magnitudes at node i are notated as $\Delta \mathbf{v}_i$ and Δv_i , respectively. For

$k = 1$, minimizing impulsive $\Delta \mathbf{v}$ maneuver magnitudes connects to minimizing fuel through the rocket equation shown in Eq. (3.4). For $k = 0$, minimum energy implies minimizing incremental kinetic energy. Strictly speaking, the solution process does not need to vary k to find a solution. However, a homotopy can be performed, for example, on a minimum energy solution to transform it to a minimum fuel solution in the same neighborhood (see example 1.g.). Of note, $k = 0.5$ does not necessarily equally weight the fuel and energy components of Eq. (3.1). A good choice of k is problem dependent, where the goal is to weight both the fuel and energy summation terms of Eq. (3.1), to be the same approximate order of magnitude. Furthermore, for emphasis and elaborated on next, the optimal solutions generated by minimizing the impulsive $\Delta \mathbf{v}$ maneuvers in Eq. (3.1) are independent of the particular rocket engine characteristics because they are only functions of position and time. The thrust and mass histories defined later by Eqs. (3.4) and (3.5) that are dependent on a rocket engine are computed after optimality.

The magnitudes of the impulsive $\Delta \mathbf{v}_i$ maneuvers are a function of their vector form written within the cost function as $J = J(\Delta v_1(\Delta \mathbf{v}_1), \dots, \Delta v_n(\Delta \mathbf{v}_n))$. The impulsive $\Delta \mathbf{v}_i$ maneuver is a function of the individual velocities before \mathbf{v}_{i-} and after \mathbf{v}_{i+} node i written as $\Delta \mathbf{v}_i = \mathbf{v}_{i+} - \mathbf{v}_{i-}$. These velocities are a function of position and time, the independent decision variables of Eq. (4.13), shown here as $\mathbf{v}_{i-} = \mathbf{v}_{i-}(\mathbf{r}_{i-1}, \mathbf{r}_i, t_{(i-1)+}, t_{i-})$ and $\mathbf{v}_{i+} = \mathbf{v}_{i+}(\mathbf{r}_i, \mathbf{r}_{i+1}, t_{i+}, t_{(i+1)-})$. In general, each node i is characterized by the state \mathbf{x}_i and time-derivative of the state $\dot{\mathbf{x}}_i$ shown later in Eq. (3.2). Node i has a before and after (denoted as a subscript as $i-$ and $i+$, respectively) for time, position, velocity, acceleration, and mass. The illustration in Fig. 2.2 shows a representative section of a trajectory with two segments and three nodes on the left sub-figure and just one node on the right

sub-figure. For position, it is assumed continuous before and after node i , so the position is notated $\mathbf{r}_i = \mathbf{r}_{i-} = \mathbf{r}_{i+}$. The velocity can discontinuously change at node i due to an impulsive $\Delta \mathbf{v}$ maneuver written as $\mathbf{v}_{i-} + \Delta \mathbf{v}_i = \mathbf{v}_{i+}$. For two-body dynamics, the accelerations before and after node i are dependent only on the position: $\mathbf{a}_{i-} = \mathbf{a}_{i-}(\mathbf{r}_i)$ and $\mathbf{a}_{i+} = \mathbf{a}_{i+}(\mathbf{r}_i)$, respectively, and since position continuity is assumed, the acceleration difference is always zero: $\Delta \mathbf{a}_i = \mathbf{a}_{i+} - \mathbf{a}_{i-} = \mathbf{0}$.

The propagated Cartesian state \mathbf{x} of this optimal control problem is

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}^T, & \mathbf{v}^T \end{bmatrix}^T \quad (3.2)$$

where \mathbf{r} is position and \mathbf{v} is velocity. The propagated state of Eq. (3.2) is not necessarily known for all time, but only at the nodes. Moreover, for two-body dynamics, time t and mass m are not needed in the propagated state of Eq. (3.2). The nonlinear spacecraft dynamics $\mathbf{f}(\mathbf{x})$ are equal to the time-derivative of the propagated state $\dot{\mathbf{x}}$, i.e. $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{\mathbf{r}}^T, & \dot{\mathbf{v}}^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{v}^T, & \mathbf{a}^T \end{bmatrix}^T$ where the time-derivative of position is equal to velocity, $\dot{\mathbf{r}} = \mathbf{v}$, and the time-derivative of velocity is equal to acceleration or the natural dynamics of the system, $\dot{\mathbf{v}} = \mathbf{a} = -(\mu/r^3)\mathbf{r}$. Note the control does not show up in the dynamics over each coasting segment. Instead, the dependent control is included as impulsive $\Delta \mathbf{v}$ maneuvers. Although not demonstrated here, perturbations can also be approximated as impulsive $\Delta \mathbf{v}$ maneuvers[157].

The decision variables are combined into a decision state \mathbf{x}_d as

$$\mathbf{x}_d = \begin{bmatrix} \tilde{t}_{1-}, & \tilde{t}_{n+}, & \Delta t, & \mathbf{r}_2^T, & \dots, & \mathbf{r}_{n-1}^T \end{bmatrix}^T \quad (3.3)$$

The time of flight Δt is included to represent the single, equal propagation time for every segment to produce time-free problems. The initial \tilde{t}_{1-} and final \tilde{t}_{n+} time-

like decision variables parameterize the initial and final orbits, respectively, for an orbit-to-orbit transfer. The total time of flight is the multiplication of Δt times the number of segments. Additionally, since unconstrained optimization is assumed, a specific total time of flight can be achieved by removing Δt as a decision variable and setting it to a fixed value. The intermediate positions, \mathbf{r}_2 to \mathbf{r}_{n-1} , in between the initial and final orbit or terminal fixed-states, are position continuous because \mathbf{r}_i is used for both the previous and next segments as shown in Fig. 2.2. For the first and last positions, \mathbf{r}_1 and \mathbf{r}_n are either fixed or vary on the initial and final orbits. In total, there are $3n - 3$ decision variables: two time-like variables, \tilde{t}_{1-} and \tilde{t}_{n+} ; one time of flight Δt variable; and $3(n - 2)$ position \mathbf{r}_i variables where n is the number of nodes. Even though the decision state can be shortened by two for the fixed-state to fixed-state case and by another one for fixed time of flight, implementation is easier to simply set the associated partials to zero instead.

Partials of the cost with respect to the decision variables $\partial J / \partial \mathbf{x}_d$ (termed cost partials) are derived in the Appendix B using the relationship between the differential of the decision variables and cost. A gradient-based, nonlinear optimizer uses these cost partials to update the independent decision variables towards optimality.

Finally, mass m_{i+} and thrust $T_{i(i+1)}$ histories are defined by Eq. (3.4) and Eq. (3.5), respectively, and are computed after optimization, i.e. there are no mass continuity or a thrust inequality constraints enforced by the optimizer. Mass over a coasting segment is constant, i.e. $m_{(i-1)+} = m_{i-}$ or $m_{i+} = m_{(i+1)-}$. Each m_i is discontinuous because the rocket equation models the change of mass for a Δv_i in Eq. (3.4) where c is the exhaust velocity. For reference, Curtis [76] presents the rocket equation in Section 11.5. The $T_{i(i+1)}$ approximation assumes a short coast for each segment. The $T_{i(i+1)}$ magnitude is forward-approximated for segment $i(i + 1)$

from node $i+$ to $(i+1)-$ in Eq. (3.5).

$$m_{i+} = m_{i-} e^{-\Delta v_i/c} \quad (3.4) \quad T_{i(i+1)} = m_{i+} \frac{\Delta v_i}{\Delta t_{i(i+1)}} \quad (3.5)$$

3.3 The Many-Segment Method

The spacecraft trajectories optimized within this current work are discretized into many segments, termed the many-segment method, affecting the resolution of the state and control. Higher vs. lower resolution is an important trade-off to consider when solving any direct optimization problem. Typically, the downside of a higher resolution is more decision variables, leading to a higher iteration count and longer runtimes. The upside is higher model fidelity and reduced sensitivity to poor initial guesses for improved robustness. In particular, when using many short segments for minimum fuel trajectories, a priori knowledge about where or when to apply impulsive $\Delta \mathbf{v}$ maneuvers is not required. In other words, a fuel optimal trajectory can be discretized into many short segments to represent an equivalent trajectory using fewer longer segments to approximately model the same physical path with the same total time of flight.

The following is a representative demonstration of finding a minimum fuel solution using either two or three impulsive $\Delta \mathbf{v}$ maneuvers (the count is not known ahead of time) with both methods (many short segments vs. fewer longer segments). The EBVP technique is used for this optimal control problem, though the many-segment could use other techniques. The trajectory is time-free and goes from a circular-orbit to a circular-orbit. Marec in 1979 presented this problem, leading to the left sub-figure of Fig. 3.1 reproduced from page 272 of Ref. [158]. For this

transfer, varying relative inclinations and sizes for the initial and final circular orbits determines if two or three impulsive $\Delta\mathbf{v}$ maneuvers are needed at optimality. Among the options, either a two-impulse generalized Hohmann transfer is needed (generalized means not necessarily co-planar with the initial or final orbit), a three-impulse bi-elliptic transfer, or a two-impulse bi-parabolic transfer (a limiting case). A many-segment approximation can model both two- or three-impulse scenarios by driving unneeded impulsive $\Delta\mathbf{v}$ maneuvers to zero, except for a small mass-leak (see appendix) at optimality. An alternative, approximately equivalent approach, is to add one segment (or impulse) at a time using Jezewski and Rozendaal's[110] n -impulse method, leveraging primer vector theory.

Two example cases from the left sub-figure of Fig. 3.1 are selected (starred): a 30° and 50° inclination change for a circular radius ratio of $r_0/r_f = 0.5$. The middle sub-figure of Fig. 3.1 is for the first case and uses 2 nodes (1 segment) and a 20-node (19 segments) equivalent solution. The right sub-figure of Fig. 3.1 is for the second case and uses 3 nodes (2 segments) of unequal flight time and the *same* 20-node method for an equivalent solution. Conveniently, the 20-node solution can be used for both the two- or three-impulse optimal solution depending on the example without a priori information. The histories of the Cartesian positions and impulsive $\Delta\mathbf{v}$ maneuvers are shown in Fig. 3.2 for both examples. Notice the 2- or 3-node Cartesian histories are approximately equivalent to the 20-node solutions, and the smallest $\Delta\mathbf{v}$ is equal to approximately the mass-leak of 10^{-4} for the 20-node solutions.

The optimal control problems in this current work use a large number of segments associated with equal flight times. While the number of decision variables is increased, the following benefits are introduced. First, as previously mentioned,

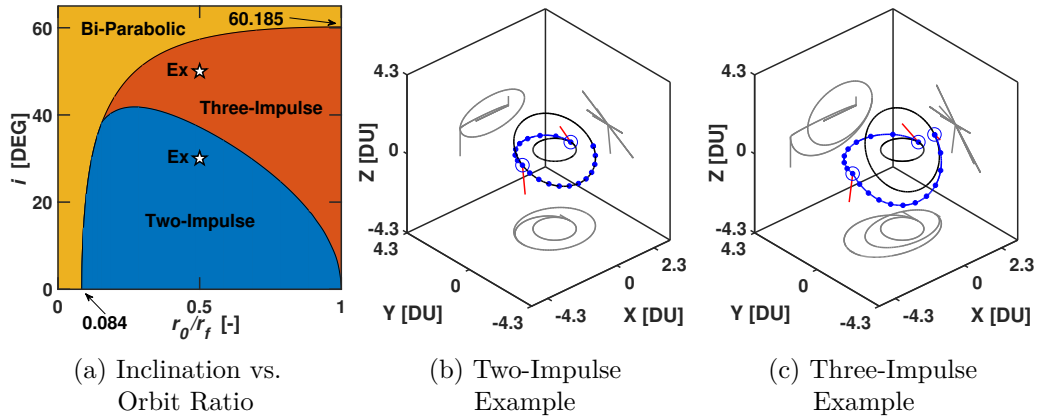


Figure 3.1: Circular-orbit to circular-orbit optimal transfers. Solid and hollow blue circles are the nodes for the 20-node, and 2- or 3-node examples, respectively; blue lines are the integrated segments; the red lines are impulsive $\Delta \mathbf{v}$ maneuvers, and the gray lines are the 2D projections of the trajectory. Note for the middle sub-figure, the 11 nodes after the second impulsive $\Delta \mathbf{v}$ maneuver are on the final orbit, i.e. a coasting phase

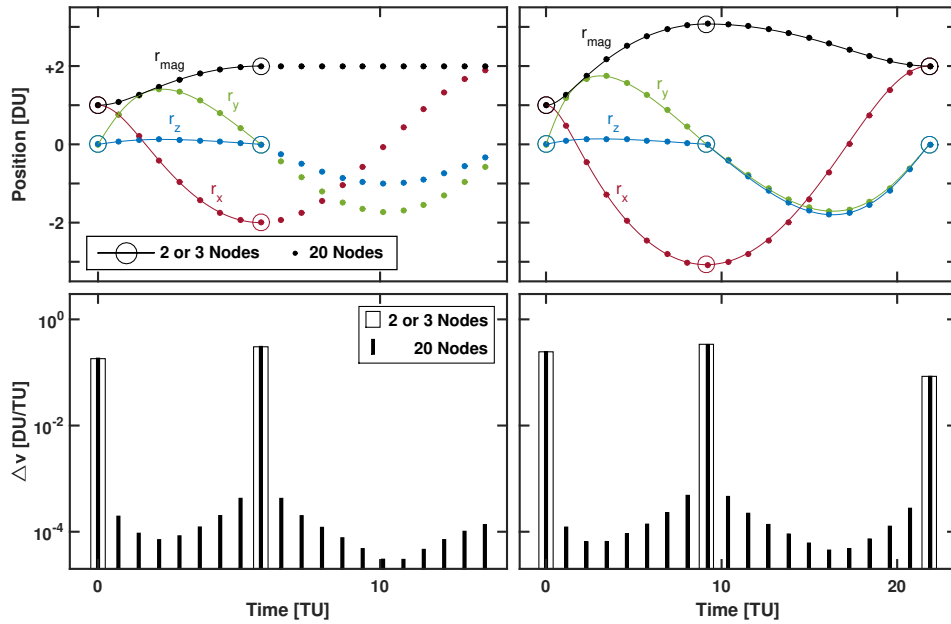


Figure 3.2: Histories of Cartesian positions and impulsive $\Delta \mathbf{v}$ maneuvers for the two- and three-impulse examples (left and right, respectively)

the spacecraft trajectories do not need a priori knowledge for the location and number of impulsive $\Delta\mathbf{v}$ maneuvers. Using more segments than needed approximately models the equivalent solution of fewer segments with varying flight times. Direct or indirect methods can solve for solutions with fewer segments, while the direct many-segment method drives unneeded nodes or impulsive $\Delta\mathbf{v}$ maneuvers to zero. Second, also as previously mentioned but now in more detail, a large number of nodes (i) allows for short segments with unique solutions guaranteed for all practical flight times, (ii) enables approximate low-thrust modeling when minimizing the sum of the square of the impulsive $\Delta\mathbf{v}$ maneuvers, and (iii) avoids the 0° and 180° transfer-plane ambiguities associated with the Lambert problem, assuming lower eccentricity transfers. The EBVP technique struggles when using free time of flight for each of the many segments because (i) a ballistic spacecraft trajectory composed of many segments is non-unique, and (ii) the optimizer will cluster the nodes to exploit small advantages in the cost function. Automatic equal spacing of the flight time mitigates both issues and reduces the burden on the optimizer that would otherwise have to enforce a spacing with explicit constraints. Third, the many-segment method is robust and flexible, succeeding for a wide range of resolutions (defined as nodes or segments per rev), ranging between at least 3 segments to at most, approximately 100 segments per rev. Though, 6 to 12 segments is usually a safe minimum and 24 to 48 segments per rev is sufficient for finer thrust fidelity. The range provides flexibility for many different initial guesses (see examples 1.e and 1.f in Section 3.4.5) where some revolutions can have more segments than others. While these aforementioned benefits for the many-segment method trade for higher optimization runtime, this increased runtime is mitigated by straightforward parallelization of the optimization algorithm via both the computation of the EBVPs

and the linear algebra of the search direction. The trade-off of increased size for unique and robust solutions is thus more palatable.

3.4 Performance Results

The performance of the unconstrained optimizer, termed $\text{FMINUNC}_{\text{UT}}$, is presented, importantly demonstrating the scaling up of the EBVP technique. As the name implies, the $\text{FMINUNC}_{\text{UT}}$ optimizer is designed for unconstrained, gradient-based, locally optimal solutions. Components of $\text{FMINUNC}_{\text{UT}}$ are explained within this section and in the appendix. In particular for $\text{FMINUNC}_{\text{UT}}$, performance curves are presented for runtime and runtime per iteration vs. number of nodes for the cost, cost plus partials, and search direction function calls. Additionally, while a single processor is used for these performance curves and all examples, parallelization improvements are demonstrated for specific linear algebra functions as well as the computation of many parallel calls to the interpolated Lambert solver ivLam . The performance of the examples are consistent with the performance curves in runtime per iteration. Table 3.2 is an example overview of different optimization characteristics, using representative fuel and energy optimal examples. The design space spanned by the examples includes two search directions, two EBVP solvers, three initial guess schemes, a homotopy between fuel and energy optimal solutions, and a continuation method to produce long flight time spacecraft trajectories with many segments. Exact performance numbers are reserved for Tables 3.3 to 3.5 in the appendix, as well as algorithms for $\text{FMINUNC}_{\text{UT}}$ for implementation convenience. The computations are performed on a 64-bit Windows desktop using a single processor, the code is compiled with Intel Visual Fortran version 17.0.0.109, and runs on an Intel Xeon CPU 3.07 GHz processor with 12 GB of memory. Note that a is

semimajor axis, e is eccentricity, i is inclination, Ω is right ascension of the ascending node, and ω is the argument of periapse.

Table 3.2: Example overview for comparisons of optimization characteristics.

	Search Direction	EBVP Solver	Initial Guess	Optimal Solution	Revs	# of Nodes	Boundary Conditions
1.a.i	BFGS	ivLam	Lambert	Fuel	0	14	Fixed-State to Fixed-State ^a
1.a.ii	HZ						
1.b.i	BFGS	ivLam	Shape	Energy	7	120	
1.b.ii	HZ						
1.c.i	BFGS	ivLam	Lambert	Fuel	0	14	
1.c.ii		rk78sh					
1.d.i	BFGS	ivLam	Shape	Energy	7	120	
1.d.ii		rk78sh					
1.e.i	BFGS	ivLam	Shape	Fuel	0	14	
1.e.ii			Manual				
1.e.iii			Lambert				
1.f.i	BFGS	ivLam	Shape	Energy	7	120	
1.f.ii			Manual				
1.f.iii			Lambert				
1.g.i ^c	BFGS	ivLam	Optimal Solution	Fuel	7	120	
1.g.ii ^c			of 1.f	Hybrid			
1.g.iii ^c				Energy			
2.a to	BFGS	ivLam	Shape	Energy	2^{α}	12 × 2^{α}	Orbit
2.h ^c					$\alpha = 1 \rightarrow 8$	to	
2.i					256	6144	Orbit ^b

^a(r_x DU, r_y DU, r_z DU, v_x DU/TU, v_y DU/TU, v_z DU/TU):

(2.495e-1, 9.684e-1, 0.000, -9.684e-1, 2.495e-1, 0.000) to
(2.901, -4.265, -4.064e-1, 3.319e-1, 2.761e-1, -1.307e-2)

^b(a , e , i , Ω , ω): (1.0 DU, 0.0, 30.0°, 0.0°, 0.0°) to (1.5 DU, 0.0, 0.0°, 0.0°, 0.0°)

^cThe exhaust velocity used for thrust and mass histories after optimization is $c = 0.6325$ DU/TU.

Notes: For simplicity of the comparisons, example 1.a.i is 1.c.i is 1.e.iii and 1.b.i is 1.d.i is 1.f.i.

Bolded cells highlight specific comparisons.

3.4.1 Runtime vs. Number of Nodes

The custom-built optimizer FMINUNC_{UT} is validated against the sparse, nonlinear, gradient-based optimizer SNOPT [159] version 7.2-12.2 by producing similar runtimes for all examples within this work. The FMINUNC_{UT} optimizer contains three main functions that occur at every major iteration: (1) the cost without partials [see

Eq. (3.1)] used inside the line search; (2) the cost plus partials used at the beginning of a new line search [see Eq. (3.1) and Algorithm 2]; and (3) the computation of the search direction (see Algorithm 3) for the line search. The runtime of these three main functions are shown in Fig. 3.3 with particular examples. For $\text{FMINUNC}_{\text{UT}}$, the functions for the cost and cost plus partials are called approximately seven and one times, respectively, for a line search to approximate the minimum with a golden ratio method, then three of the last four points are chosen to fit a quadratic polynomial. The process is repeated where a new search direction is computed for a new line search. $\text{FMINUNC}_{\text{UT}}$ runs until the magnitude of the cost partial is below a small tolerance. The two other tolerances are a minimum cost difference $|J_i - J_{i-1}|$ and an iteration maximum that stops optimization if it has failed or is close to an optimal solution. An algorithm for $\text{FMINUNC}_{\text{UT}}$ is in the appendix as Algorithm 2.

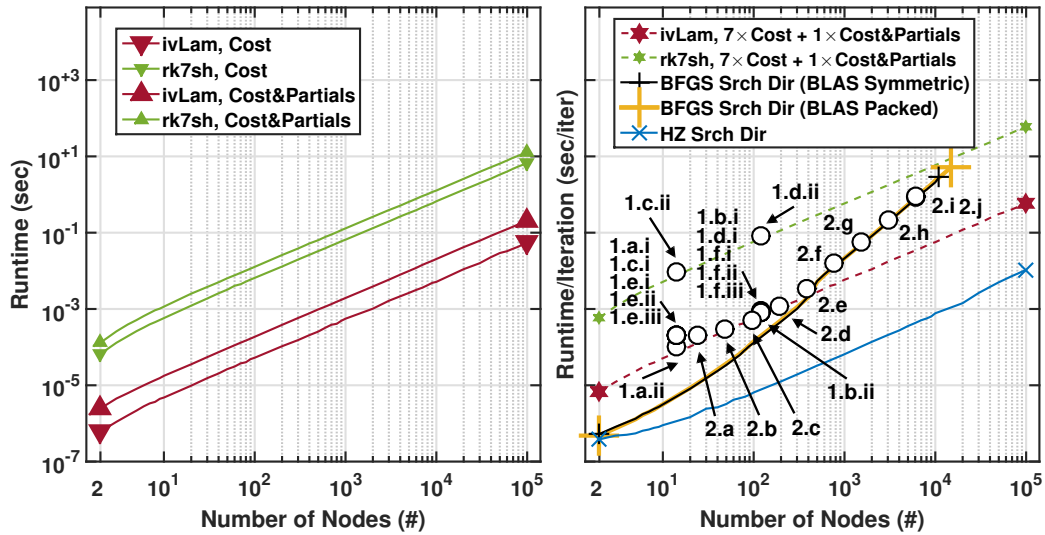


Figure 3.3: $\text{FMINUNC}_{\text{UT}}$ runtime of the cost and cost plus partials function calls (left), and runtime per iteration of the search direction as well as the combined cost and partials (right). Examples are plotted.

On the left side of Fig. 3.3, the cost and cost plus partials functions grow lin-

early with respect to node count for both EBVP solvers: ivlam and rk7sh. The rk7sh BVP solver is approximately $100\times$ or two orders of magnitude slower than ivLam for all node counts, independent of implementation within an optimizer. Accordingly, shooting methods to solve EBVPs should only be used when analytic solutions to the BVPs are not available, such as the case when perturbations are present. Note also that the rk7sh is reporting an extremely conservative performance. Only three fixed-steps are used in each segment, and only one iteration is required to solve a two-body only EBVP. A more general shooting method may require more steps per segment and will certainly require more iterations. In contrast, one cost call for ivLam using only two nodes produces a runtime of 6.1×10^{-7} seconds, which includes overhead. Note that overhead means other operations within the cost call besides the ivLam EBVP solver, such as the summation of impulsive $\Delta \mathbf{v}$ maneuvers, if-else statements, unpacking of the decision state in Eq. (4.13) into a convenient structure of arrays, and more. The ivLam function by itself in this work has a runtime of 4.3×10^{-7} seconds, consistent with Ref. [16] when using the highest accuracy, fixed one-step version of the BVP solver.

The right side of Fig. 3.3 illustrates linear and approximate quadratic growth for the HZ and BFGS search direction computation, respectively. BFGS stands for the Broyden[160]-Fletcher[161]-Goldfarb[162]-Shanno[163] method and HZ stands for the Hager-Zhang [164] conjugate gradient method. Conjugate gradient methods, such as the HZ search direction, are desirable because they are memory efficient due to the lack of an approximate Hessian, important for problems with many decision variables. However, the BFGS search direction converges to optimality in less runtime versus these memory efficient search directions for the examples within this current work. Many[165] other search directions for $\text{FMINUNC}_{\text{UT}}$ were consid-

ered, including the steepest descent method (simple but produces intractably large optimization runtimes), limited-BFGS [166], and the conjugate gradient methods: Fletcher-Reeves [167] and Polak-Ribiere [168]. BFGS and HZ are the top two performing search directions, and are therefore demonstrated with specific examples (see examples 1.a and 1.b in Section 3.4.3).

The EBVP solvers ivLam or rk7sh dominate runtime computation for all node counts if the HZ search direction is utilized. The HZ computation in runtime per iteration is less than ivLam and rk7sh, and linear in growth on the log-log scale of the right sub-figure of Fig. 3.3. The BFGS search direction dominates the runtime after approximately 400 nodes for ivLam and memory runs out before dominating rk7sh. The runtime per iteration at the crossover point for the ivLam EBVP and BFGS search direction is approximately 2.1×10^{-3} seconds. As shown for the representative examples 1.a and 1.b in Section 3.4.3, even if the BFGS search direction quadratically increases in runtime versus node count, the better convergence rate produces more desirable, shorter overall runtimes.

The BFGS search direction methods is implemented with either symmetric or packed BLAS [169] subroutines to produce similar runtime per iteration curves on a single processor. The packed BLAS subroutines enable larger node counts than the symmetric BLAS subroutines, however packed storage does not benefit from parallelization as shown in Section 3.4.2. For this work, the allowable max node count difference between the packed and symmetric BLAS routines is approximately 4000 nodes. The quadratic growth of the BFGS search direction is dependent on the BLAS matrix and vector operations, symmetry, and the packed storage for the approximate inverse Hessian. The $\mathcal{O}(n^2)$ storage of the approximate inverse Hessian is the only large memory allocation in FMINUNC_{UT}, enabling large problems on

the order of thousands of decision variables. Example 2 shows up to 18,429 decision variables ($3 \times 6144 \text{ nodes} - 3$).

All examples compute the cost, cost plus partials, and search direction about seven, one, and one times, respectively, for every major iteration that updates the decision state shown in Eq. (4.13). The actual runtime per iteration of examples that use either ivLam or rk78sh for the EBVP solver, or BFGS or HZ for the search direction, are shown as specific points on the right of Fig. 3.3. As expected, all example points lie close, but above the dominating curves in this log-log plot, noting the expected runtime per iteration of an example includes the sum of both curves (“*Search Direction*” and “ $7 \times \text{Cost} + 1 \times \text{Cost \& Partials}$ ”) plus time for auxiliary calculations. More details on the examples using different search directions and EBVP solvers follow in Section 3.4.3 and Section 3.4.4, respectively.

The computational runtime of the dynamics has been minimized for smaller problems below 400 nodes and rendered relatively insignificant above 400 nodes if a BFGS search direction and ivLam EBVP solver are used. The new bottleneck is the linear algebra within the BFGS search direction, necessary for minimizing overall runtime in this current work. Additionally, all ivLam solutions take the same reliable amount of time because they use a single fixed iteration. In contrast, numerically integrated shooting methods will vary in runtime per segment and may not converge at all.

3.4.2 Parallelization Improvements: Speed Up vs. Threads

Spacecraft trajectory optimization using EBVPs is parallelizable for improved performance. FMINUNC_{UT} uses the parallelizable symmetric BLAS functions (`dsymv`, `dsyr`, and `syr2`) and OpenMP [170] for the ivLam EBVPs. These parallelization

improvements are especially important for larger problems with thousands of nodes or segments. As such, Fig. 3.4 shows the speedup as a function of thread count for symmetric BLAS routines and ivLam EBVPs. The packed BLAS subroutines do not show a performance speed up vs. thread count so they are not presented here. Modest performance gains are present above approximately 100 nodes for the linear algebra of the BLAS functions, while their peak performance remains approximately less than half the theoretical linear speedup. On the other hand, ivLam finds some gains even for 10 nodes, and achieves near the theoretical maximum speedups as the node counts approach and exceed 1000. For simplicity and ease of comparison, none of the examples from Table 3.2 use parallelization.

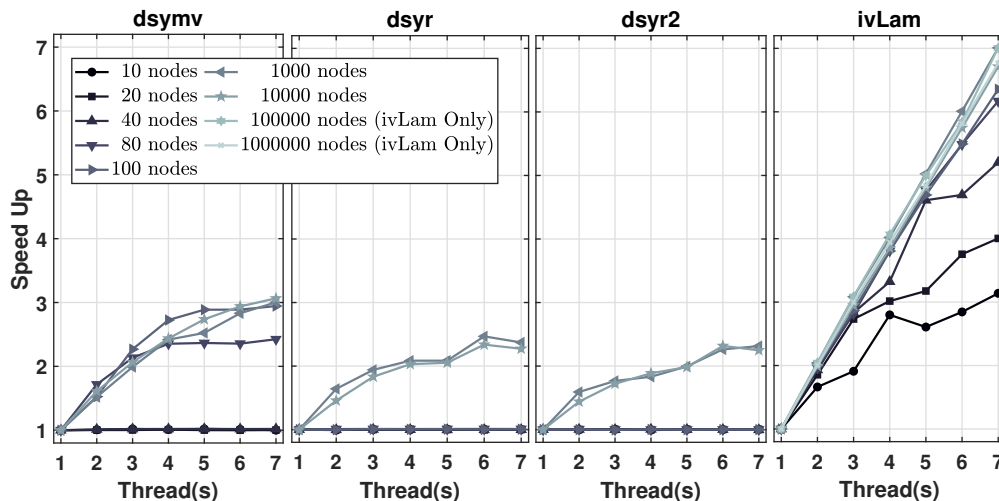


Figure 3.4: Speed up of three symmetric BLAS subroutines and the ivLam EBVP.

3.4.3 Search Direction Demonstration: Examples 1.a and 1.b

For both the fuel (zero-rev) and energy (seven-rev) optimal examples, 1.a and 1.b, respectively, BFGS outperforms HZ by minimizing total optimization runtime, even though HZ can be faster per iteration and increases only linearly with respect to

node count. In Fig. 3.5, examples 1.a and 1.b are time-free transfers from a fixed-state to fixed-state. The initial guesses for 1.a and 1.b are Lambert 1.e.iii and shape-based 1.f.i, respectively, shown in Fig. 3.6. Likewise respectively, the optimal solutions are shown in Fig. 3.7. Example 1.a.i and 1.b.i (BFGS search direction) converged in 0.0335 and 7.19 seconds, respectively, a significant speed up of $197\times$ and $157\times$ over 1.a.ii and 1.b.ii (HZ search direction). While only two examples are presented here, the performance comparisons represent a trend of many other internally tested examples. The exact performance numbers for examples 1.a and 1.b are in Table 3.3.

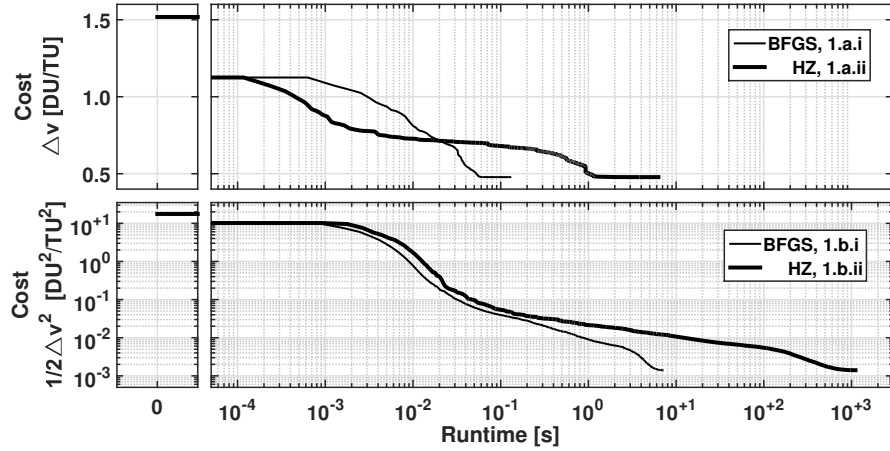


Figure 3.5: Cost vs. runtime for two examples using either the BFGS or HZ search direction for $FMINUNC_{UT}$. Total major iteration count is 209, 56336, 8821, and 1314720 for 1.a.i, 1.a.ii, 1.b.i, and 1.b.ii, respectively.

3.4.4 EBVP Solver Demonstration: Examples 1.c and 1.d

Two EBVP solvers are considered: ivLam, a two-body Lambert problem solution method; and rk78sh, a general dynamics shooting method using only two-body dynamics for simplicity. Like examples 1.a and 1.b, examples 1.c (minimum fuel, zero-rev) and 1.d (energy optimal, seven-rev) are time-free transfers from a fixed-

state to fixed-state. The optimal solutions are also shown in Fig. 3.7 and the initial guesses for 1.c and 1.d are the Lambert 1.e.iii and shape-based 1.f.i, respectively, shown in Fig. 3.6. The exact performance numbers are in the appendix in Table 3.3. For the EBVP solver comparison in terms of total runtime: examples 1.c and 1.d show a speed up factor for ivLam with respect to rk78sh of $60\times$ and $105\times$, respectively. As mentioned in Section 4.1, the non-trivial slowdown of rk78sh is due in part to the dimension of the ODE system, at a minimum 6 for the state and 36 for the variational STM. Additionally, the runtimes for rk78sh are conservative. While this EBVP executes with a variable number of steps per segment, it always converges after one iteration because of an exactly correct guess provided by ivLam. In other words, while this current work only models two-body dynamics for simplicity, the runtimes of rk7sh or rk78sh demonstrate a baseline of additional cost associated with solving the EBVPs with numerical integration and a shooting method. More complicated dynamics, more steps per segment, and more iterations for the EBVP solution would only increase the runtime. The exact performance numbers for examples 1.c and 1.d are in Table 3.3.

3.4.5 Initial Guess Demonstration: Examples 1.e and 1.f

FMINUNC_{UT} is robust and performance agnostic to many different initial guesses shown in Fig. 3.6. Within Fig. 3.6, example 1.e is a zero-rev, minimum fuel trajectory, and example 1.f is a seven-rev, minimum energy trajectory. Both examples 1.e and 1.f are time-free transfers from a fixed-state to fixed-state. In order from left to right, the shape-based initial guesses are generated using codes described in Refs. [171] and [141], the manual initial guesses are simple user-defined waypoints, and the Lambert initial guess utilizes a zero or seven-rev ballistic solution from the

initial to final position on an orbit discretized into segments with equal times of flight. All initial guesses converge to the same optimal solutions shown in Fig. 3.7, i.e. examples 1.e.i, 1.e.ii, and 1.e.iii converge to the left sub-figure of Fig. 3.7 and examples 1.f.i, 1.f.ii, and 1.f.iii converge to the right sub-figure. Note that in Figs. 3.6 and 3.7, the z -axis is not to scale. Example 1.e.iii is chosen to be the initial guess for examples 1.a and 1.c, and example 1.f.i is the initial guess for 1.b and 1.d. Both the minimum fuel (1.e) and minimum energy (1.f) examples perform similar relative to their different initial guesses. In particular, the runtime per iteration for examples 1.e and 1.f is 0.0002 or 0.008 seconds, respectively, and the total runtimes are approximately of the same order at tens of milliseconds and seconds for examples 1.e and 1.f, respectively. The exact performance numbers for examples 1.e and 1.f are in Table 3.4. Exceptional initial guesses are possible; the intent here is to demonstrate that converged solutions do not typically require good initial guesses.

3.4.6 Cost Homotopy Demonstration: Example 1.g

For example 1.g shown in Fig. 3.8, a homotopy solution is produced in sequential stages by adjusting the k variable in Eq. (3.1). First a minimum energy solution is obtained, next the k variable is adjusted, and then a new optimal solution is found in the same local neighborhood. Thus, the obtained minimum fuel and hybrid solutions are similar in revolutions to the minimum energy solution. In particular, the hybrid solution uses $k = 0.025$ to equally weight both the minimum fuel and energy contributions of the cost function in Eq. (3.1) by ensuring both terms are of the same order of magnitude. Additionally, the k parameter can be tuned to limit max thrust, producing a smooth thrust profile shown in Fig. 3.9 for example 1.g.ii. Figure 3.9 also shows the mass history. These smooth thrust profiles with

lower maximum thrust are ideal initial guess candidates for low-thrust trajectory optimization with a max thrust constraint. The orbital element histories of example 1.g are shown in Figs. 3.10 and 3.11, all showing piecewise-constant behavior that emphasizes each segment is ballistic. Notice the parallels between the profiles for each k parameter for both thrust and mass. There are about nine burns for the minimum fuel, energy optimal, and hybrid cases. The cases do not end at the same time because the problem is time-free. The minimum fuel solution uses less fuel compared to the energy optimal solution shown in Fig. 3.9. The minimum fuel

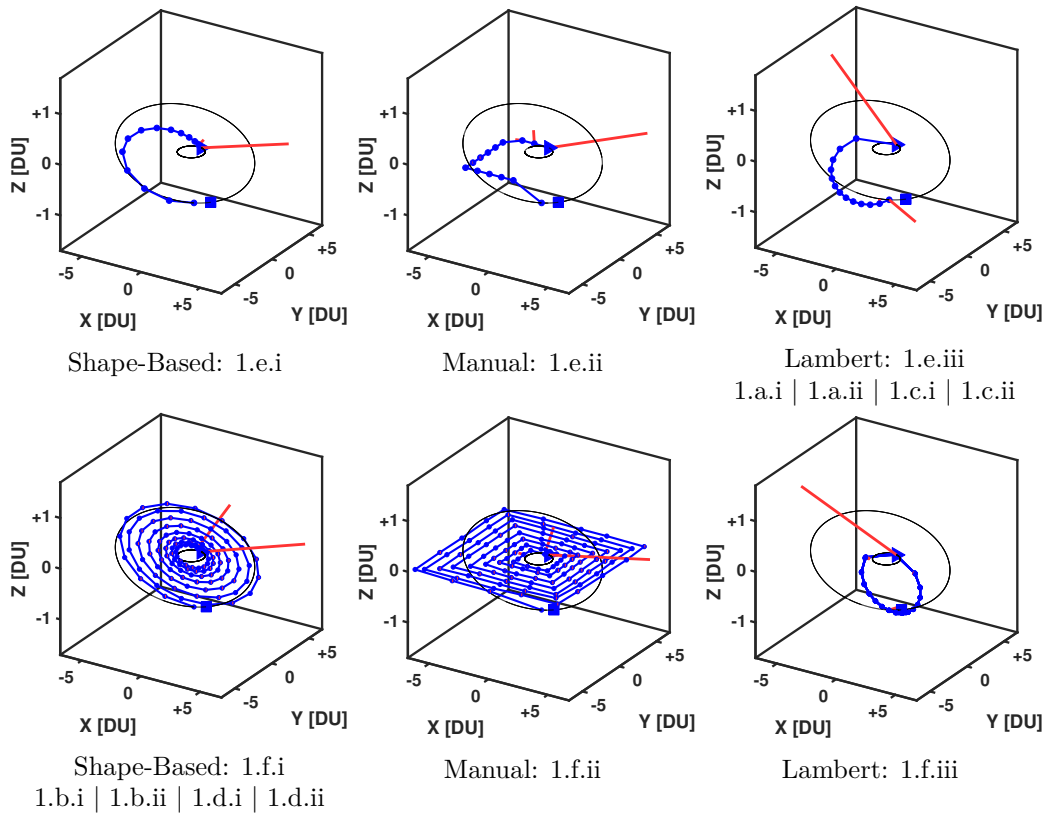


Figure 3.6: Initial guesses: min fuel, zero-rev: 1.a, 1.c, 1.e (top) and min energy, seven-rev: 1.b, 1.d, 1.f (bot). z -axis not to scale. Solid blue circles are the nodes. Blue lines connect the nodes, but are not the integrated segments. Red lines are impulsive Δv maneuvers.

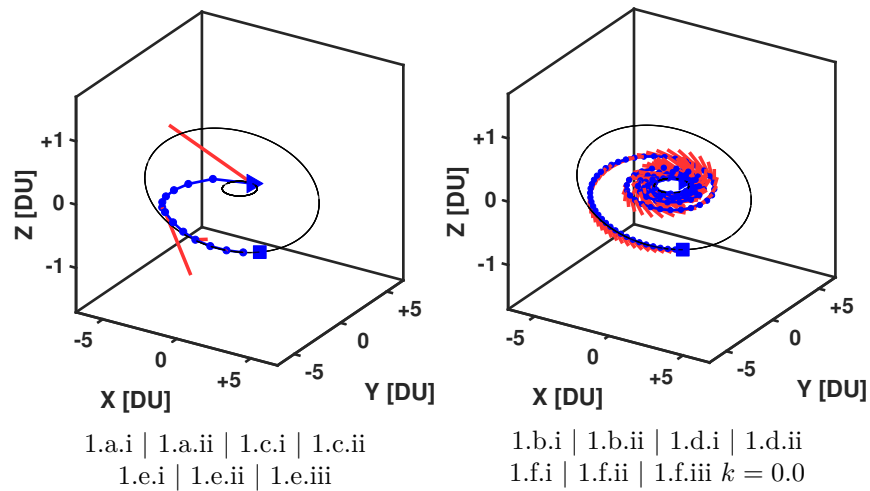


Figure 3.7: Optimal solutions from the initial guesses of Fig. 3.6: min fuel, zero-rev: 1.a, 1.c, 1.e (left) and min energy, seven-rev: 1.b, 1.d, 1.f (right). z -axis not to scale. Solid blue circles are the nodes. Blue lines connect the nodes, but are not the integrated segments. Red lines are impulsive $\Delta\mathbf{v}$ maneuvers.

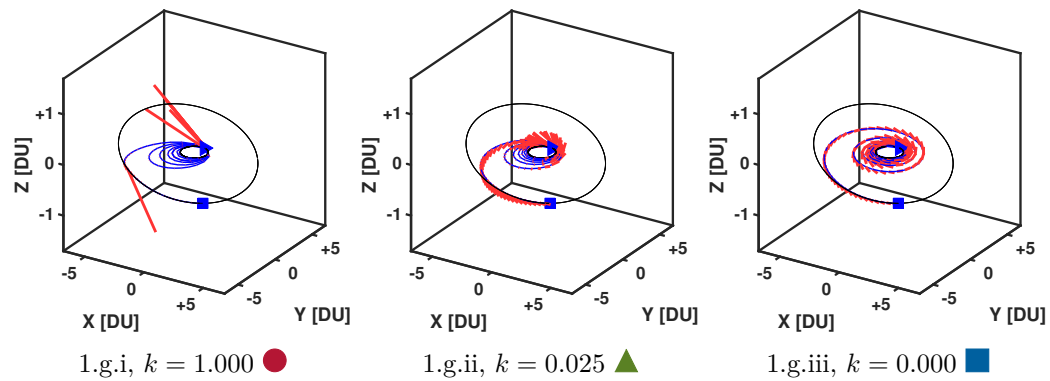


Figure 3.8: Optim. solutions: exs. 1.g.i and 1.g.ii from 1.g.iii. z -axis not to scale. Solid blue circles are the nodes; blue lines connected the nodes, but are not the integrated segments; the red lines are impulsive $\Delta\mathbf{v}$ maneuvers.

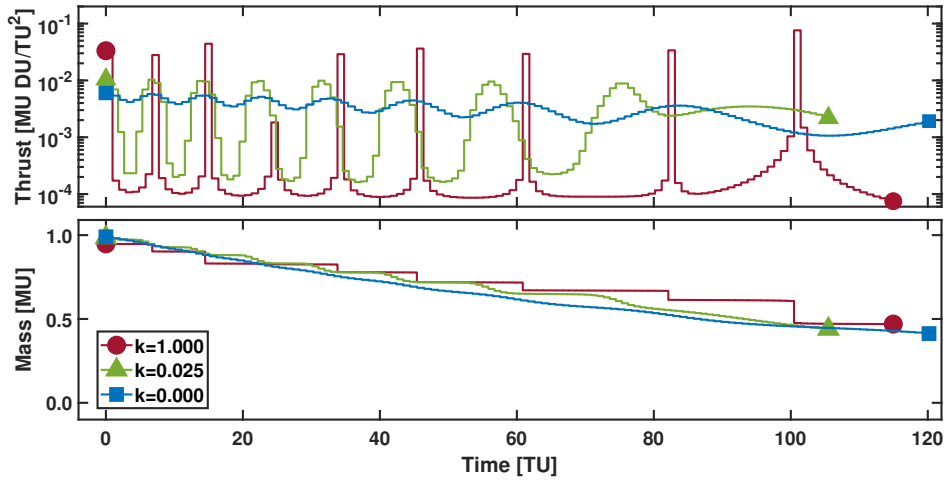


Figure 3.9: Mass and thrust history: 1.g.i, $k = 1.0$, $m_{120+} = 0.4695$ MU | 1.g.ii, $k = 0.025$, $m_{120+} = 0.4427$ MU | 1.b, 1.d, 1.f, 1.g.iii, $k = 0$, $m_{120+} = 0.4129$ MU. $c = 0.6325$ DU/TU for 1.g.

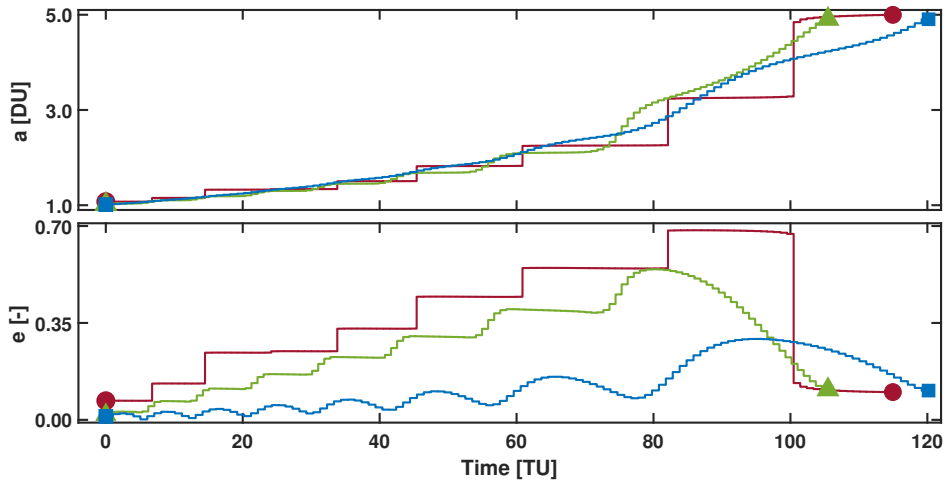


Figure 3.10: a and e history: 1.g.i, $k = 1.0$ | 1.g.ii, $k = 0.025$ | 1.b, 1.d, 1.f, 1.g.iii, $k = 0$.

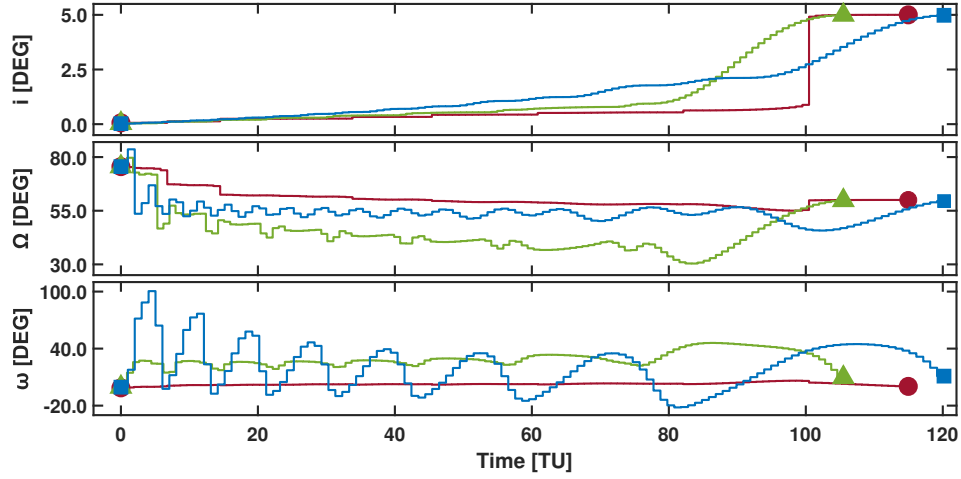


Figure 3.11: i , Ω , and ω history: 1.g.i, $k = 1.0$ | 1.g.ii, $k = 0.025$ | 1.b, 1.d, 1.f, 1.g.iii, $k = 0$.

solution also converges to the multi-rev Lambert solution, distributing the energy raising maneuver near the initial orbit over seven revolutions. The orbital element histories shown in Figs. 3.10 and 3.11 are all consistent with the thrusting maneuvers of Fig. 3.9. Fuel or energy optimal solutions vary orbital elements more dramatically or gradually, respectively. Notably for the minimum fuel solution, most of the inclination change is accomplished with the last burn away from the gravitational body. Lastly, the exact performance timings are omitted because the runtime per iteration of 1.g.i and 1.g.ii is similar to the already presented example 1.f.

3.4.7 Large Number of Nodes Demonstration: Example 2

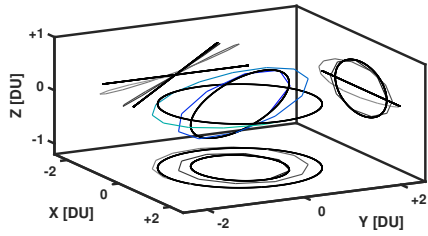
This final example demonstrates a spacecraft trajectory with up to 256 revolutions and discretized into 6,144 nodes (6,143 segments) or 18,429 decision variables (3×6144 nodes $- 3$). The solution is found with $\text{FMINUNC}_{\text{UT}}$, leveraging fast ivLam EBVPs, an unconstrained formulation, and a continuation method. The trajectory is a transfer from an initial to final orbit, not fixed-state to fixed-state like in the cases

of example 1. Within the BFGS search direction computation of $\text{FMINUNC}_{\text{UT}}$, the $\mathcal{O}(n^2)$ operations grow quickly with respect to number of nodes relative to the linear $\mathcal{O}(n)$ growth of the cost and cost plus partials computations as shown in Fig. 3.3. To mitigate this quadratic growth on runtime, the continuation method doubles the number of revs and nodes of the previous lower-node and lower-rev optimal solution to eventually obtain a higher-resolution thrust profile with many revs and nodes. The doubling stages are defined as α stages, so the node count is $n = 12 \times 2^\alpha$ and the rev count is 2^α where $\alpha = 1 \rightarrow 8$. The 9th α stage is the final stage at 6,144 nodes and 256 revs. The corresponding decision variable count is $3n - 3$ or $36 \times 2^\alpha - 3$ where $\alpha = 1 \rightarrow 9$.

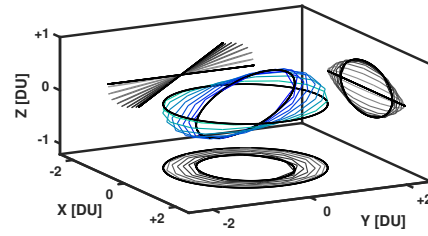
The continuation method description is as follows. First, the initial guess is shape-based [171, 141] containing 24 nodes and 2 revs for approximately 12 nodes per rev. The optimal solution of this 24-node case is example 2.a and is shown on the top left of Fig. 3.12 with its 2D projections. Second, the number of revs and nodes are doubled eight times with initial guesses from the prior optimal solutions. Example 2.c is shown in the top right of Fig. 3.12. Third and finally, the node count is doubled one last time, but the rev count is kept fixed, to produce the 6,144 node, 256 rev, optimal solution as shown on the bottom of Fig. 3.12. The continuation method produces a total runtime of 5.5 hours vs. 23.5+ hours without the continuation method (example 2.j)—a significant speed up. In fact, inspection of Table 3.5 for the performance numbers of example 2 reveals runtime is dominated by the doubling of the final two stages (2.h and 2.i). Runtime for those are 50.1 minutes and 4.5 hours, respectively, or a total of 97.1% of the total runtime for all the stages. The runtimes for the other examples 2.a to 2.e take under a minute, and examples 2.f and 2.g take 1.6 and 7.6 minutes, respectively. This increasing

runtime per stage trend verifies that the computation of the BFGS search direction dominates runtime after 400 nodes, also shown in the right figure of Fig. 3.3. In other words, the smaller problems using a couple dozen to a thousand segments, examples 2.a through 2.g, are particularly fast and thus useful for preliminary solutions in the two-body problem. For larger node counts up to 6,144, a lower-segment solution can be augmented with more segments for more revs or for higher-resolution thrust and mass histories if optimization runtime of hours is tolerable. Additionally, per Section 3.4.2, solutions with more than 1,000 segments begin to notably benefit from parallelization improvements. The high-resolution thrust and mass histories of example 2.i are in Fig. 3.13.

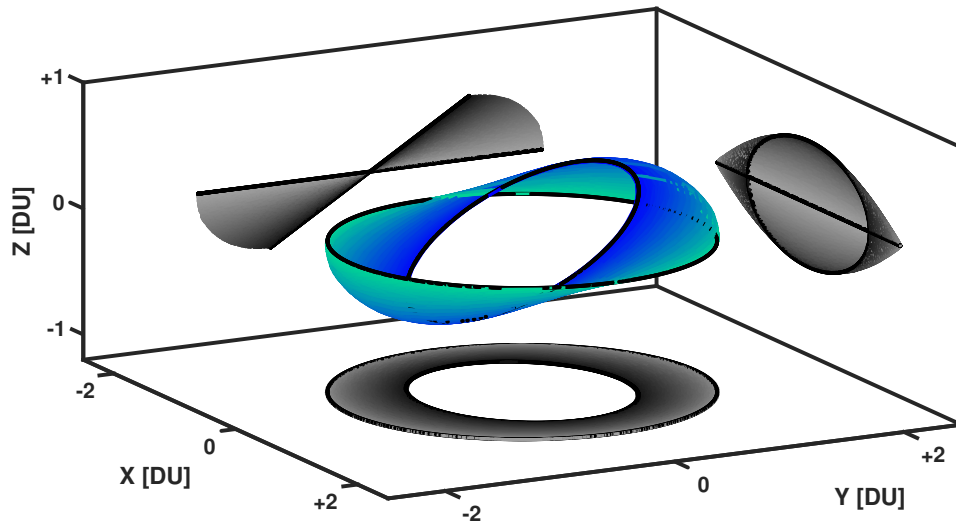
Besides increasing runtime, increasing node count for a given solution can also decrease the minimum energy cost in Eq. (3.1). In Table 3.5, notice the minimum energy cost approximately halves for every doubling of nodes and revs. See the ‘Minimized Cost’ row. This shrinking cost can cause the nonlinear optimization program, FMINUNC_{UT}, to prematurely exit because the impulsive $\Delta\mathbf{v}$ maneuvers are already small, but not necessarily optimal. To avoid this numerical issue, the minimum energy summation in Eq. (3.1) can benefit from scaling by the number of nodes n or segments $n - 1$. This scaling was not included in Example 2, but more complicated problems may benefit from similar tuning efforts.



Optimal solution: 2.a (24 nodes, 2 revs)



Optimal solution: 2.c (96 nodes, 8 revs)



Optimal solution: 2.i (6144 nodes, 256 revs)

Figure 3.12: Optimal solutions: trajectories of select examples 2.a, 2.c, and 2.i of the large number of nodes problem for some of the stages of the continuation method. The blue-green lines are the trajectory. The grey lines are the 2D projections of the trajectory. The color gradient is consistent with the mass and thrust history of Fig. 3.13.

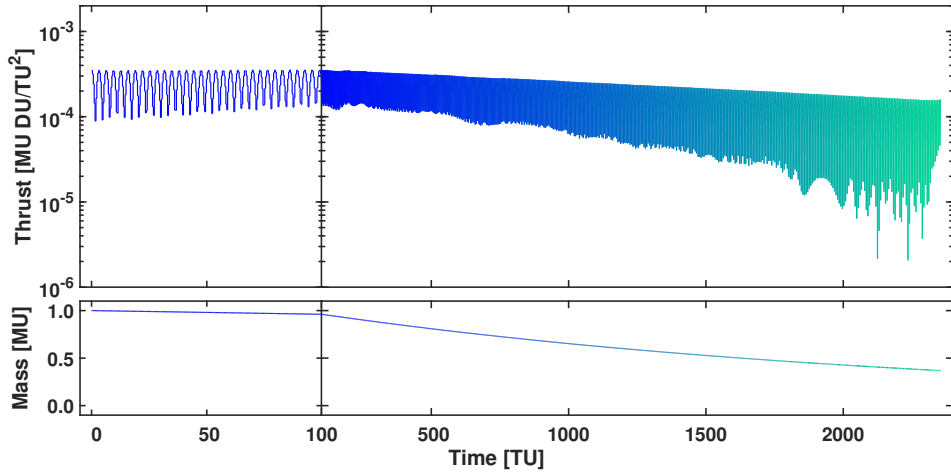


Figure 3.13: Thrust and mass history: example 2.i contains 6144 nodes and 256 revs, and uses $c = 0.6325$ DU/TU for a final mass $m_{6144+} = 0.3843$ MU. The smaller time domain of the left side reveals the smooth behavior of the thrust profile. The color gradient is consistent with the optimal solution in Fig. 3.12.

3.5 Conclusions

Direct, spacecraft trajectory optimization using many embedded boundary value problems, specifically the Lambert problem for two-body dynamics, is advanced relative to the state of the art. For preliminary design of spacecraft trajectories, this embedded Lambert problem technique discretizes the path into many short-arc segments, demonstrated here for trajectories with up to 6,144 nodes (6,143 segments or 18,429 decision variables) and 256 revolutions. Previous work in primer vector theory, flyby tours, and direct impulsive $\Delta\mathbf{v}$ maneuver optimization uses embedded boundary value problems for on the order of a few dozen segments at most. Problems with this many revolutions are typically intractable to optimize without using orbital averaging or indirect methods. The optimization performance is enabled by the unconstrained formulation and a newly developed Lambert solver, providing a fixed-computational cost to model the dynamics, a robust global domain of

convergence, and inexpensive and exact partials. The optimization runtime on a single processor of the 6,144-node problem is 5.5 hours, serving as a practical upper limit on node size for this technique. Smaller problems with dozens to up to approximately a thousand nodes optimize in substantially shorter runtimes at seconds to minutes, respectively. In other words, the technique is particularly effective at generating fast, preliminary solutions for these smaller problems. For solutions requiring more short-arc segments for more revs up to and beyond the practical limit of 6,144 nodes, other techniques can be preferable or even necessary. The computational runtime of the dynamics and partials for smaller problems is reduced through the use of embedded Lambert problem solvers and for larger problems this computation is rendered relatively insignificant. In particular, these larger problems at or greater than approximately 400 nodes are dominated by any linear algebra required by the well-known Broyden-Fletcher-Goldfarb-Shanno line search direction computation. This search direction outperformed a variety of conjugate gradient methods in total optimization runtime. Besides runtime performance, other strengths of the technique include insensitivity to poor initial guesses and its unconstrained formulation using Cartesian coordinates, making it simple to implement. The technique is also parallelizable because of the independence of each segment and the computation of the linear algebra for runtime speedups. The numerical examples presented are time-free, fixed-state to fixed-state and orbit-to-orbit that contain no thrust or mass constraints for both minimum fuel and energy optimal solutions. Future work will demonstrate perturbation approximations and incorporate the Sundman transformation into this embedded Lambert problem optimization technique.

3.6 Appendix

3.6.1 Performance Tables for Examples 1 and 2

The performance tables are for examples 1 and 2 (see Fig. 3.3). Tables 3.3 to 3.5 show total runtime per iteration, total runtimes, number of iterations that update the decision state, function calls, cost, and optimality tolerances. Function call counts are for the computation of the cost, the cost plus partials, and the search direction, respectively.

Table 3.3: Performance of examples 1.a, 1.b, 1.c, and 1.d, comparing search direction and EBVP solver.

Search Direction EBVP Solver Initial Guess	BFGS: 1.a.i	HZ: 1.a.ii ivLam	ivLam: 1.c.i Lambert	BFGS rk78sh: 1.c.ii
^a Minimum Fuel Ex. 1.a, 1.c				
Total Runtime/Iter. sec/iter	0.0002	0.0001	0.0002	0.0093
Total Runtime sec	0.0335	6.6140	0.0335	1.9996
Major Iterations	209	56336	209	216
^b Function Calls	1503, 210, 209	757731, 56337, 56336	1503, 210, 209	1604, 217, 216
Search Direction EBVP Solver Initial Guess	BFGS: 1.b.i	HZ: 1.b.ii ivLam	ivLam: 1.d.i Shape-Based	BFGS rk78sh: 1.d.ii
^a Minimum Energy Ex. 1.b, 1.d				
Total Runtime/Iter. sec/iter	0.0008	0000.0009	0.0008	000.0818
Total Runtime sec	7.1900	1132.1470 (18.9 min)	7.1900	745.1770 (12.4 min)
Major Iterations	8821	1314720	8821	9113
^b Function Calls	61848, 8822, 8821	15410419, 1314721, 1314720	61848, 8822, 8821	63635, 9114, 9113

^aMin. cost for 1.a, 1.c, and 1.e is 0.4777 DU/TU, and 1.b, 1.d, and 1.f is 0.0014 DU²/TU².

Optimality is less than 10⁻⁶ for both 1 and 2. Lastly for simplicity, example 1.a.i is 1.c.i is 1.e.iii and 1.b.i is 1.d.i is 1.f.i.

^bFunction calls are for cost only, cost plus partials, and search direction, respectively.

Table 3.4: Performance of examples 1.e and 1.f for the initial guess demonstrations.

Search Direction EBVP Solver Initial Guess		Shape-Based: 1.e.i	BFGS ivLam Manual: 1.e.ii	Lambert: 1.e.iii
Minimum Fuel Example 1.e				
Total Runtime/Iter.	sec/iter	0.0002	0.0002	0.0002
Total Runtime	sec	0.0632	0.0454	0.0335
Major Iterations		278	186	209
Function Calls		1917, 279, 278	1353, 187, 186	1503, 210, 209
Search Direction EBVP Solver Initial Guess		Shape-Based: 1.f.i	BFGS ivLam Manual: 1.f.ii	Lambert: 1.f.iii
Minimum Energy Example 1.f				
Total Runtime/Iter.	sec/iter	0.0008	0.0008	0.0008
Total Runtime	sec	7.1900	7.4670	7.8030
Major Iterations		8821	9085	9593
Function Calls		61848, 8822, 8821	63248, 9086, 9085	66725, 9594, 9593

Table 3.5: Performance of ex. 2 that demonstrates a continuation method for a large number of nodes.

Search Direction EBVP Solver Initial Guess		BFGS ivLam				
		Shape-Based: 2.a 24 nodes, 2 revs	Prev. Sol'n: 2.b 48 nodes, 4 revs	Prev. Sol'n: 2.c 96 nodes, 8 revs	Prev. Sol'n: 2.d 192 nodes, 16 revs	Prev. Sol'n: 2.e 384 nodes, 32 revs
Minimum Energy Example 2						
Total Runtime/Iter.	sec/iter	0.0002	0.0003	0.0005	0.0012	00.0035
Total Runtime	sec	0.0484	0.1155	0.5573	2.9240	12.6760
Major Iterations		247	441	1119	2458	3628
Function Calls		1728, 248, 247	2834, 442, 441	6938, 1120, 1119	15239, 2459, 2458	22269, 3629, 3628
Minimized Cost	DU^2/TU^2 $\times 10^{-5}$	898.456	466.768	237.003	119.343	059.890
Initial Guess		Prev. Sol'n: 2.f 768 nodes, 64 revs	Prev. Sol'n: 2.g 1536 nodes, 128 revs	Prev. Sol'n: 2.h 3072 nodes, 256 revs	Prev. Sol'n: 2.i 6144 nodes, 256 revs	Lambert: 2.j 6144 nodes, 256 revs
Minimum Energy Example 2						
Total Runtime/Iter.	sec/iter	00.0156	000.0558	0000.2184	00000.8516	00000.7839
Total Runtime	sec	97.3610 (1.6 min)	456.4620 (7.6 min)	3003.5550 (50.1 min)	16218.7830 (270.3 min, 4.5 hr)	84660.0492+ (1411.0+ min 23.5+ hr)
Major Iterations		6235	8174	13751	19046	108001+
Function Calls		38712, 6236, 6235	51372, 8175, 8174	90611, 13752, 13751	163889, 19047, 19046	652671+, 108002+, 108001+
Minimized Cost	DU^2/TU^2 $\times 10^{-5}$	030.013	015.033	007.527	003.769	009.058-

Chapter 4

Piecewise Sundman Transformation for Spacecraft Trajectory Optimization using Many Embedded Lambert Problems

4.1 Introduction

A spacecraft trajectory is typically discretized into smaller paths, defined as segments here, for numerical propagation and for approximate numerical solutions of continuous optimal control problems. A problem formulation balances the number of variables and associated problem size against increased computational runtime to efficiently and accurately capture the system dynamics and control. Conway and Paris [172] in chapter 3 and Betts [73] in sections 3.5 and 4.5 provide good overviews of the discretization process for optimal control problems.

Previous spacecraft trajectory optimization work by the authors [173] used a simple and obvious discretization where each trajectory segment is a short arc that uses equal times of flight (TOF). Such a scheme works well for low eccentricity, but

loses efficiency at high eccentricity. Furthermore, this scheme encounters spatial inefficiency problems for the segments if continuous thrust significantly changes semi-major axis and eccentricity (a and e , respectively) of the trajectory. In Ref. [173], the equal flight time discretization was coupled to an unconstrained optimization framework that uses *many* embedded boundary value problems (EBVPs), namely a Lambert problem solver, to implicitly enforce position continuity—an important equality constraint in spacecraft trajectory design. Each segment uses two-body spacecraft dynamics and neighboring impulsive maneuvers to approximate finite thrust. The optimizer iterates on time and position only as decision variables, removing the need to guess velocity or control. Inverse dynamics formulations in optimal control similarly iterates on paths rather than controls, e.g. shape-based spacecraft trajectory design [136, 141, 142, 144]. This EBVP technique motivates the development of more spatially efficient discretization schemes, expanding the design space.

A discretization of equal changes in eccentric anomaly (ΔE) is an efficient spatial discretization for spacecraft trajectory design using two-body dynamics. Equal steps in eccentric anomaly E provides a near-equal spacing and a consistent number of steps per revolution along a Keplerian orbit. Figure 4.1 presents two representative discretizations for near-circular and eccentric Keplerian orbits. The black dots are nodes that bound the segments of the discretization. The segments are propagated with equal times of flight, or regularized times of flight to produce equal ΔE 's. The two left sub-figures show a discretization that has equal times of flight and equal ΔE , respectively. Notice these two sub-figures are almost visually indistinguishable for a near-circular orbit. The two right sub-figures show the same equal times of flight and ΔE discretization, respectively, emphasizing the

problem of a spatially inefficient discretization for eccentric orbits that is absent in near-circular orbits. For eccentric orbits, Kepler's second law dictates that equal flight time segments produce spatially shorter distances near the apoapsis or longer distances near the periapsis.

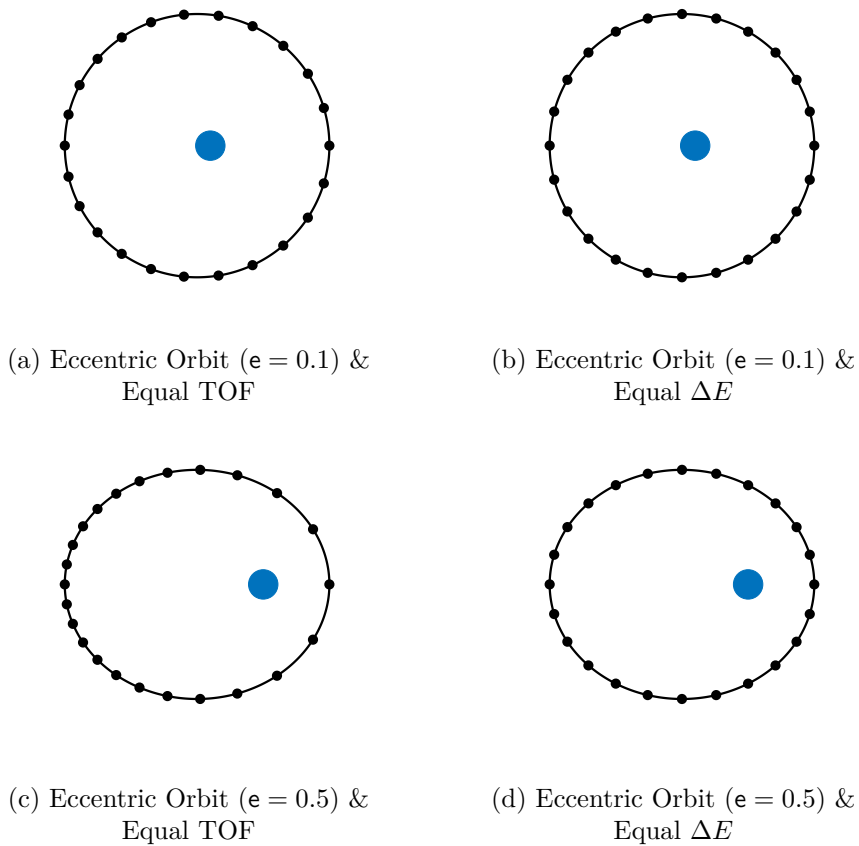


Figure 4.1: Discretization of a trajectory into segments bounded by nodes defined by either equal flight times (a,c) or equal ΔE (b,d) for a near-circular and eccentric Keplerian orbit.

To achieve equal steps in ΔE for a rev, the regularization of spacecraft trajectory dynamics informed the solution of this current work. In 1908, Cowell's method [174] was the first numerical integration routine to more accurately model the Sun-

perturbed orbit of one of Jupiter’s moons. However, it was noticed a singularity was approached as the relative distance between two bodies decreased. The now well-known Sundman transformation ([175], 1913) on the dynamics helped mitigate this singularity. Furthermore, over the last century, other stabilizing improvements to avoid this singularity include transforming the equation of motion into a linear oscillator (See Ref. [176], Sect. 3.2.3, 2017 for a straight-forward derivation.) through a coordinate transformation in the complex plane by Levi-Civita ([177], 1906), and embedding integrals of motion such as energy [178, 179]. The Sundman transformation has other consequences as it also produces a spatially even discretization for Keplerian orbits and generalizes Kepler’s equation for all conics (circular, elliptical, parabolic, and hyperbolic). For example, see the flight time function by Bate et al. that uses Stumpff functions [79] (p. 196, eq. 4.4-12, 1971) or a similar solution by Battin that uses universal functions [74] (p. 178, eq. 4.81, 1999). This Sundman transformation on the continuous dynamics is now referred to as the differential Sundman transformation (DST). For further historical account, the excellent monograph of Roa [176] (Chapter 2) on regularization is recommended.

The discretization solution for this chapter is a time of flight function [See Eq. (4.1).] named the piecewise Sundman transformation or PST. This new flight time function divides the total flight time of a trajectory by generating the individual flight times of each segment. A spatially even discretization is recovered and is approximately equivalent to the discretization produced by regularizing the dynamics. All the segments of a trajectory can be equal or near-equal step in ΔE , but no explicit transformation on the dynamics is applied. Further discussion on what form of the PST achieves equal or near-equal step in ΔE is throughout, specifically Sections 4.2.3, 4.3, and 4.4.

For the outline of this chapter, the PST is discussed and applied in six sections plus an appendix. This current Section 4.1 describes the motivation, i.e. a spatially efficient discretization of a spacecraft trajectory is sought, defined as equal fixed-steps in ΔE , but without a time transformation on the dynamics. For Section 4.2, the PST is explicitly defined and related to the DST, density functions, and the eccentric anomaly differential. In Section 4.3, a comparative numerical study is performed where an eccentric Keplerian orbit is integrated forward in time using a set of different differentials and their approximately equivalent flight time functions, notably comparing the DST and PST. In Section 4.4, spacecraft trajectory optimization problems are formulated using EBVPs and the PST, and then performance is analyzed. The examples include the simple modeling of a Keplerian orbit (example 1), a performance comparison between the EBVP technique and the IVP technique (example 2), energy optimal transfers that significantly change semimajor axis and eccentricity (example 3), and both fuel and energy optimal trajectories from a circular to hyperbolic orbit (example 4). In particular, the IVP technique uses a Kepler solver and is a state of the art for low-fidelity, multi-impulsive maneuver trajectory design, directly competing with the alternative EBVP technique. In Section 4.5, the PST for spacecraft trajectory optimization is summarized. Finally, the appendix includes a discussion on the partial derivatives (partials) for the optimization problems in Section 4.4.

4.2 Piecewise Sundman Transformation

The new flight time function, the PST, for a spacecraft trajectory segment is

$$\Delta t = r_*^\alpha \Delta \tau \tag{4.1}$$

where Δt is the time of flight, $\Delta\tau$ is the pseudo time of flight, and the piecewise-constant coefficient is composed of an exponent α and a piecewise-constant position magnitude r_* . The “piecewise-constant position magnitude” is referred to as “position coefficient” for simplicity. This flight time function in Eq. (4.1) is linear in $\Delta\tau$ and, if it is used repeatedly for every segment of a trajectory, the $\Delta\tau$ has the same value for all segments (or phase in a more complicated multi-phase trajectory). $\Delta\tau$ automatically varies time of flight Δt in a regular, accordion-like fashion via the scaling position factor r_* of a segment, where the exponent α controls the relative spacing.

The position coefficient is a piecewise-constant function of the terminal positions of a segment, $r_* = r_*(\mathbf{r}_0, \mathbf{r}_f)$. Only three position coefficients are considered for simplicity: (i) an initial position coefficient $r_* = r_0$, (ii) a final position coefficient $r_* = r_f$, and (iii) an average position coefficient $r_* = (r_0 + r_f)/2$. Different α 's enable a fixed-step in $\Delta\tau$ propagation to be proportional to fixed-steps in different Keplerian orbital anomalies. For $\alpha = 0$, a fixed-step in $\Delta\tau$ is proportional to a fixed-step in mean anomaly (or time), as no Sundman transformation is applied. For $\alpha = 1, 2$, and $3/2$, a fixed-step in $\Delta\tau$ is proportional to E , true anomaly, and the intermediate anomaly [180], respectively. However, for trajectories utilizing large impulsive maneuvers, neighboring segments can have significantly different orbital elements. Similarly, segments at the start and end of a trajectory can also have significantly different orbital elements if many small impulsive maneuvers are applied over a long total flight time. Therefore, the relationship of r_* and α to the discretization for trajectories that include impulsive maneuvers in the two-body problem is more complicated and is explored with respect to the differential of E in Section 4.2.3 and in examples 1 and 3 in Section 4.4.2.

Time of flight functions are a beneficial alternative to a time-transformation on the dynamics because time, not a time-like variable, remains the independent propagation variable. This function form decreases implementation difficulty and maintains the dimension of the state. Alternatively, the approach using transformed dynamics requires an additional state and more complicated dynamics. During development of these flight time functions, other approaches were considered to enforce efficient discretizations proportional to E : (i) Use a constrained optimizer such as SNOPT [159] or IPOPT [181] to enforce specific, desirable flight times per segment as constraints. While feasible, these nonlinear constraints add an unnecessary numerical burden on the optimizer if a flight time function, such as the PST, is available to automate flight time selection. See Ref. [172], Ch. 4, Sections 4.5 and 4.6 for an example of explicit flight time constraints on the segments of two Lunar trajectory missions. (ii) A second option is to modify a solution to the Lambert problem to receive directly the transformed time-like independent variable, such as the square of the eccentric anomaly difference $z = \Delta E^2$ (see Eq. 4.5-7 in Bate et al. [79] for a Sundman-related transformation [175] and its relation to semimajor axis). However, a single z value cannot be used for all segments of a trajectory that transitions from elliptic to hyperbolic orbits because z is negative for hyperbolic orbits and positive for elliptical orbits. Additionally, the negative values are not physically intuitive to guess. (iii) A third solution includes the square root of semimajor axis \sqrt{a} as the coefficient of the Sundman transformation instead of the unity value used in this current work. While shown to be useful by Baù et al. to propagate elliptical [182, 183] or hyperbolic [184] orbits, or by Roa and Peláez [185] to accurately model hyperbolic trajectories, or by the authors (unpublished numerical experiments) for eccentric orbit propagation in perturbed two-body dynamics, this solution cannot

smoothly transition between elliptic and hyperbolic conics. The transition from elliptical orbits toward the parabolic boundary approaches infinite semimajor axis and flight time and thus loses accuracy. For hyperbolic orbits, the negative semi-major axis under the square root must be handled too, usually with a discontinuous sign flip or via a complex modeling space. As mentioned, a goal of this current work is a regularization method that not only works for all conics (circular, elliptic, parabolic, and hyperbolic), but also transitions through the parabolic boundary. Of note, a recent work in 2020 by Baú and Roa [186] replaces a simpler Cartesian coordinate representation of position and velocity with eight orbital elements, accurately propagating a comet near, and transitioning through, the parabolic boundary. The coefficient of the Sundman transformation used in Ref. [186] is unity. (iv) The fourth and final option considered uses the time regularization $dt = \frac{1}{h}r^2d\tau$, where h is angular momentum and $\mu = 1.0 \text{ DU}^3/\text{TU}^2$ is assumed. This regularizing solution is promising for initial value problems because fixed-steps in $\Delta\tau$, while proportional to true anomaly, are proportional to eccentric anomaly *on average*¹ over one period of a Keplerian orbit. This on-average behavior is also independent of a or $e < 1$. However, this regularization solution cannot be applied to BVPs because h is not known before the BVP is solved.

In the next sub-sections before the numerical studies in Sections 4.3 and 4.4, the PST is explained relative to (i) the DST, (ii) density functions, and (iii) the differential of E . Density functions are a regularizing method by another name that are commonly found in optimal control and other engineering disciplines. The differential of E helps explain how the PST enables a larger design space for optimization

¹Similar to the derivation in Section 4.2.3, Kepler's equation can be integrated by separation of variables after substitution of $dt = \frac{1}{h}r^2d\tau$ to get $\tau_p = \int_0^{2\pi} \frac{\sqrt{1-e^2}}{1-e\cos E} dE = 2\pi$. Divide both sides by 2π to get the average change in τ with respect to E over one period is one, i.e. $[\frac{d\tau}{dE}]_{avg} = \frac{\tau_p}{2\pi} = 1$.

problems using EBVPs in Section 4.4.

4.2.1 Differential Sundman Transformation

A differential form of the Sundman transformation, or DST, is

$$dt = r^\alpha d\tau \tag{4.2}$$

where dt is the differential of time, $d\tau$ is the differential of pseudo time, and, similar to the definitions of the terms in Eq. (4.1), the coefficient is composed of a position magnitude r and an exponent α . The differential equation for time can be written as $dt/d\tau = r^\alpha$. Eq. (4.1) is the approximation of Eq. (4.2) where $\Delta\tau$ is a finite difference, instead of differential, multiplied by a piecewise-constant r_* to the power of an α to produce a finite Δt for a segment. The PST approaches the DST as the flight time of a segment approaches zero. The DST is used explicitly in the trajectory propagation problem of Section 4.3 and the optimization problems of Section 4.4 (examples 1 and 2).

4.2.2 Density Functions

Density functions are another way to understand the PST. The density function or related monitor function are a mesh generation tool for segments that enforce the equidistribution principle: the area under a particular function is the same for every interval. The PST also generates a convenient mesh, so to speak, for an orbital mechanics optimal control problem. The equidistribution concept was first introduced in 1974 by Burchard [187] for finding variable nodes for optimal spline approximations. Many engineering and scientific fields have used density functions for mesh generation techniques, such as Thompson [188, 189, 190], who reviewed

the solution of partial differential equations (PDEs), and others described density functions in the finite element method field [191, 192, 193, 194]. More recently, Baker [195] reviewed the evolution of meshes, demonstrating the critical drivers of reliable and efficient mesh generators are computational fluid dynamics and aerodynamics. Huang and Russell [196] presented mesh adaptivity with emphasis on time-dependent PDEs. Mitchell and McClain [197] compared several hp-adaptive versions of the finite element method for solving PDEs that modify the width and the polynomial degree of the mesh. Optimal control started using monitor functions and density functions when they were first presented by Betts et al. [198] in 2003 and Zhao and Tsiotras [199, 200] in 2011, respectively. Afterward, a density function was used in an adaptive mesh-refinement algorithm from Darby et al. [201, 202] that created an adaptive pseudospectral method to iteratively determine the number, width, and polynomial degree in each segment given an accuracy. Liu et al. [203] used a density function inspired by Zhao and Tsiotras to help design a multi-criterion, hp-adaptive strategy that introduced prior knowledge, intermediate error, and curvature as useful criteria for adaptive refinement. Peng et al. [204] used the same curvature-based density function by Zhao and Tsiotras for a symplectic algorithm in an optimal control problem.

The density function integral or equidistribution mesh for one segment of a spacecraft trajectory is

$$\int_{t_0}^{t_f} \rho(t) dt = \zeta \quad (4.3)$$

where $\rho(t)$ is a general density function integrated from an initial time t_0 to final time t_f of a segment and ζ is the constant of the equidistant principle. The $\rho(t)$ can be a function of other variables besides time t not explicitly shown here. If $\rho(t) = \bar{\rho}$ is piecewise-constant along a spacecraft trajectory, where the bar means constant,

then the integral of Eq. (4.3) is significantly simplified to

$$\bar{\rho}\Delta t = \zeta \quad (4.4)$$

where $\Delta t = t_f - t_0$. For a spacecraft trajectory optimization problem, the general density function $\bar{\rho}$ and constant ζ can be decision variables of the optimizer. If $\bar{\rho} = 1/r_*^\alpha$, the inverse of a position coefficient, and $\zeta = \Delta\tau$, the same value for every segment along a spacecraft trajectory, then Eq. (4.4) is $\frac{1}{r_*^\alpha}\Delta t = \Delta\tau$, which is Eq. (4.1) rearranged. In other words, the PST is an approximation of a density function method, where $1/r_*^\alpha$ and Δt vary along a trajectory to regularize the discretization, while $\Delta\tau$ is a positive value updated by an optimizer.

4.2.3 Differential of Eccentric Anomaly

The change in eccentric anomaly ΔE and flight time of a segment is related through Kepler's equation. For a short Keplerian segment, ΔE is approximately equivalent to the differential because $\Delta E = dE + \mathcal{O}(d^2 E) \approx dE$. See Ref. [205], Section 2.2 for the relationship between a Taylor series and the equivalent first order, second order, etc. differentials that combine to form a total differential. Kepler's equation for elliptical orbits is

$$nt = E - e \sin E \quad (4.5)$$

where the mean motion is $n = \sqrt{\mu/a^3}$ and the gravitational parameter is μ . Replacing n , the differential of Eq. (4.5) is

$$\sqrt{\frac{\mu}{a^3}} dt = (1 - e \cos E) dE \quad (4.6)$$

Substitute Eq. (4.2) into Eq. (4.6) and the position magnitude, represented as $r = a(1 - e \cos E)$ ([76], Eq. 3.22). Rearrange and the eccentric anomaly differential is

$$dE = \sqrt{\mu} a^{\alpha-3/2} (1 - e \cos E)^{\alpha-1} d\tau \quad (4.7)$$

Equation (4.7) was first reported in 1980 by Janin and Bond [206] (Eq. 2.5). Each segment of a trajectory is characterized by a value of ΔE , ideally the same for every segment. Equation (4.7) shows that ΔE is a function of a , e , E , and α . If the exponent $\alpha = 1$ or $3/2$, then ΔE is independent of e or a , respectively, for a ballistic segment. Moreover, Eq. (4.7) can be integrated in closed-form via separation of variables to find τ_p for different values of α . The results are reported in Table 4.1 and the derivation is explained in detail by Pellegrini et al. in [207]. Notably, for $\alpha = 0, 1, 3/2$, and 2 , τ_p is a function of a only, a only, e only, and both a and e , respectively.

Now, assuming Eq. (4.1) for the flight time of a segment, it is not possible to find an α that exactly discretizes a trajectory into equal ΔE if a and e are changing. Nevertheless, the most efficient discretization is given by $\alpha = 3/2$. This discretization is (i) periodic in E , (ii) approximately equal to a constant ΔE for lower e , (iii) independent of changes in a , and (iv) the decrease in ΔE is practically limited near the parabolic boundary of $e = 1.0$. To elaborate, consider τ_p for the different Sundman exponents over a practical domain of $10^{-1} \leq a \leq 10^3$ DU and $0 \leq e \leq 0.9999$, as shown in Fig. 4.2. For the left plot in Fig. 4.2, $\alpha = 3/2$ is the only Sundman exponent that maintains a constant period when varying a only. For the right plot in Fig. 4.2, only minor growth for τ_p is shown due to varying e only, considered acceptable for transfers that vary both a and e . In other words, for $\alpha = 3/2$, a fixed-step size can efficiently model across a large domain of a and e .

Note $e = 0.0$ for $\alpha = 3/2$ and 2 in the left plot, and $a = 1.0$ DU for $\alpha = 0, 1,$ and 2 for the right plot. Other fixed values of a or e for Fig. 4.2 produce qualitatively similar results. Example 3 in Section 4.4 shows this approximately constant segment count per rev for $\alpha = 3/2$.

Lastly, the modeling of elliptic and hyperbolic arcs can be related through a replacement of variables. Kepler's equation for ellipses shown in Eq. (4.5) is converted to the equivalent Kepler's equation for hyperbolas by replacing E with $\sqrt{-1}H$ where H is the hyperbolic anomaly. It is also assumed that $a > 0$ and $a < 0$ model ellipses and hyperbolas, respectively. After replacing E and using some trigonometry identities, multiply Kepler's equation by $\sqrt{-1}$ to extract the imaginary part. Kepler's equation for hyperbolas is thus

$$\sqrt{\frac{\mu}{-a^3}}t = -(H - e \sinh H) \quad (4.8)$$

and the differential of H from Eq. (4.8), after substitution of the DST and $r = a(1 - e \cosh H)$, is

$$dH = \sqrt{\mu}(-a)^{\alpha-3/2}(1 - e \cosh H)^{\alpha-1}d\tau$$

The hyperbolic anomaly is emphasized less in this current work because many segments over many revs are not needed to model the hyperbolic phase of a solution. Still, a hyperbolic phase is demonstrated in Example 4 of Section 4.4. Overall, Eqs. (4.5) and (4.8) and their differential counterparts approximately model the discretization of the trajectories in this work. Eq. (4.5) in particular is only used as a post-process step to track the evolution of ΔE in Section 4.3 and examples 1, 2, and 3 in Section 4.4.

Table 4.1: Period in τ_p for Sundman exponents $\alpha = 0, 1, 3/2$, and 2. The τ_p 's are plotted in Fig. 4.2.

$\alpha = 0$: $\tau_p = 2\pi\sqrt{\frac{a^3}{\mu}}$	$\alpha = 1$: $\tau_p = 2\pi\sqrt{\frac{a}{\mu}}$
$\alpha = 3/2$: $\tau_p = \frac{4}{\sqrt{\mu(1+e)}}\text{ELL}_K\left(\sqrt{\frac{2e}{1+e}}\right)$	$\alpha = 2$: $\tau_p = 2\pi\frac{1}{\sqrt{\mu a(1-e^2)}}$

$\text{ELL}_K(\cdot)$ is the complete elliptic integral of the first kind.

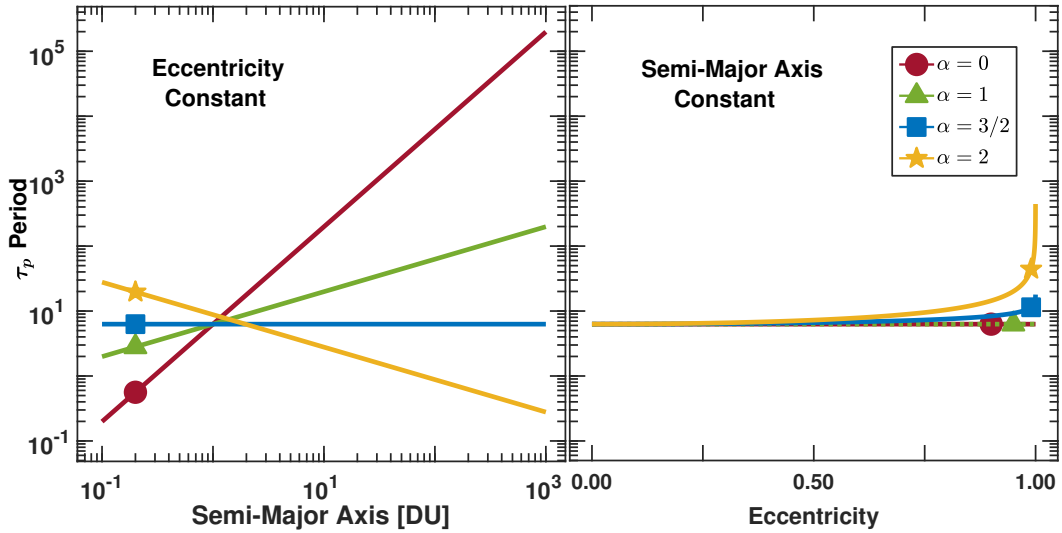


Figure 4.2: τ_p vs. a and e : τ_p for varying a and e using the expressions in Table 4.1. For varying a (left), $e = 0.0$ for $\alpha = 3/2$ and 2. For varying e (right), $a = 1.0$ DU for $\alpha = 0, 1$, and 2; and the curves for $\alpha = 0$ and 1 overlap.

4.3 Trajectory Propagation: Piecewise Functions vs. Differential Transformations

The Sundman transformation is one of many differentials that regularize dynamics and provide improved discretizations. In this numerical study, an eccentric Keplerian orbit is propagated with numerical integration, comparing these new flight time functions and the differential equivalents: the PST vs. the DST. The flight

time functions enable integration of the Keplerian orbit in time, while regularizing the dynamics with a differential transformation integrates in pseudo time. These time of flight functions can use initial values, such as initial position, velocity, or acceleration of each segment. In contrast, and later in Section 4.4 for trajectory optimization using EBVPs, initial and final positions are available per segment, as well as initial and final two-body accelerations, but not velocities. Second-order contributions to a total differential to motivate other forms of flight time functions are not considered in this current work for simplicity. This work considers $\Delta t \approx dt$ only, and not, for example, $\Delta t \approx dt + \frac{1}{2}d^2t$.

For the results in this section and next, Sections 4.3 and 4.4, the Fortran 2008 code is compiled with Microsoft Visual Studio, Community 2015, Version 14.0.25431.01 Update 3, and uses Intel Parallel Studio XE 2017 Composer Edition for Fortran Windows. The code runs on two quad-core Intel Xeon CPU X5647 2.93 GHz processors, and can access 24 GB of memory. Only the largest problem of this current work, the third sub-example of example 3 in Section 4.4, makes use of seven cores and up to approximately 11 GB of memory.

4.3.1 Regularization on Two-Body Dynamics

The regularization on the two-body dynamics is derived as follows. First consider a six-state \mathbf{x} that combines position \mathbf{r} and velocity \mathbf{v} in a Cartesian space:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \end{bmatrix} \quad (4.9)$$

and its time-derivative

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \end{bmatrix}$$

$\dot{\mathbf{x}}$ can also be written as

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$$

to explicitly show the derivative with respect to time t , i.e. $\dot{\square} = \frac{d}{dt}\square$. Moreover, $\dot{\mathbf{x}}$ is equal to the dynamics:

$$\dot{\mathbf{x}} = \mathbf{f}(t) = \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{a} \end{bmatrix} \quad (4.10)$$

where two-body acceleration \mathbf{a} is defined as $\mathbf{a} = -\mu/r^3\mathbf{r}$. Notice no control shows up in the dynamics. Instead, the control, if needed, is indirectly included as impulsive maneuvers between segments as seen in Fig. 2.1 and minimized in the optimization problems of Section 4.4.

To regularize the two-body dynamics of Eq. (4.10), a differential transformation is applied by multiplying Eq. (4.10) by the DST shown in Eq. (4.2):

$$\mathbf{x}' = \dot{\mathbf{x}}t' = \frac{d\mathbf{x}}{d\tau} = \frac{d\mathbf{x}}{dt} \frac{dt}{d\tau} = \mathbf{f}(t)r^\alpha \quad (4.11)$$

where the derivative with respect to a pseudo time τ is defined as $(\)' = \frac{d}{d\tau}(\)$. The independent variable is thus τ when numerically integrating Eq. (4.11) and time t is a dependent variable, increasing the six-state \mathbf{x} in Eq. (4.9) to seven states. In other words, t is tracked by integrating $t' = r^\alpha$. Other differential time transformations on the two-body dynamics shown in Eq. (4.10) are similar to the DST, for example arc-length defined as $dt = v^{-1}d\tau$.

4.3.2 Propagation Results

Five total differentials are compared to five total equivalent flight time functions. These regularization scenarios are summarized in Table 4.2 and, in column-order, are time, three different Sundman transformations, and a generic arc-length transfor-

mation. The arc-length transformation in this work demonstrates that flight time functions built from differential regularizations are not limited to the Sundman transformation. Different applications will benefit from different regularizations. Also note, for column two, constant steps in E are achieved for $\alpha = 1$ if \mathbf{a} and \mathbf{e} are constant, as is the case for Keplerian orbits.

Table 4.2: Regularization scenarios for the forward propagation of an orbit. $(\mathbf{a}, \mathbf{e}) = (5.0 \text{ DU}, 0.5)$.

Regularization Scenario	Time $\alpha = 0$	Sundman $\alpha = 1$	Sundman $\alpha = 3/2$	Sundman $\alpha = 2$	Arc-Length
Differential	$dt = d\tau$	${}^a dt = r d\tau$	${}^a dt = r^{3/2} d\tau$	${}^a dt = r^2 d\tau$	$dt = v^{-1} d\tau$
Piecewise	$\Delta t = \Delta\tau$	${}^b \Delta t = r_0 \Delta\tau$	${}^b \Delta t = r_0^{3/2} \Delta\tau$	${}^b \Delta t = r_0^2 \Delta\tau$	$\Delta t = v_0^{-1} \Delta\tau$

^aDST | ^bPST

For the representative forward-propagation example, a Keplerian orbit ($\mathbf{a} = 5.0 \text{ DU}$ and $\mathbf{e} = 0.5$) is numerically integrated with a fixed-step, eighth-order Runge-Kutta method [208]. This orbit is discretized into three different segment quantities: 12, 24, and 48 segments via both the piecewise and differential transformations. Thus, there are 30 scenarios to consider, i.e. ('5 differentials' + '5 piecewise functions') \times '3 discretization quantities'. For all 30 scenarios, the ΔE and energy are tracked for each segment of the Keplerian orbit. Energy is defined as $\mathcal{E} = v^2/2 - \mu/r$, a constant of integration for Keplerian motion, where r and v are the magnitude of position and velocity, respectively. The performance is measured for every segment as a deviation from a constant ΔE and a deviation from a constant \mathcal{E} . The 'energy error' of a segment is defined as the current \mathcal{E} minus the previous \mathcal{E} , both measured at the initial position and velocity of the respective segments. The energy error increases for longer segments because each segment is just a single fixed step of the Runge-Kutta integrator.

The flight time functions and the differential equivalents must propagate

the exact same total flight time for a fair comparison. The DST has an analytic Keplerian period τ_p as reported in Table 4.1, so stepping in 12, 24, and 48 equal steps is straight-forward: simply divide τ_p by these quantities. For the PST, stepping the same amount of $\Delta\tau$, derived from the DST, produces a slightly different total flight time. A constraint on total time of flight must be imposed as a pre-process step for all the DST and PST scenarios: $|\Delta t_{\text{total}} - \Delta t_{\text{total},s}| < 10^{-8}$, where Δt_{total} is the computed total flight time and the inclusion of subscript s is for specified. Then the unique $\Delta\tau$ for each scenario is used for the forward propagation. Otherwise, slight variations in the final, total time of flight occur between the numerically integrated solutions, including the DST. Total time of flight is also constrained in examples 1 and 2 for the optimization problems in Section 4.4. Both these examples use the PST and DST, where $\alpha = 1$.

For efficient forward propagation of Keplerian orbits in time, the piecewise functions only need to approximate the differential behavior shown in Figs. 4.3 and 4.4. Figure 4.3 plots the ΔE of each constant $\Delta\tau$ segment vs. the segment number as a percentage of the total number of Keplerian orbit segments. For the 24-segment orbit, the right subfigure of Fig. 4.1 is the same spacecraft trajectory using the DST and $\alpha = 1$, while visualizations of the other 24-segment trajectories of Table 4.2 are excluded for brevity. Similar Keplerian orbit discretization visuals using the Sundman transformation are found in Fig. 1 of Ref. [207]. As expected in Fig. 4.3, the DST with $\alpha = 1$ produces a discretization that is proportional to ΔE , while the other differentials produce a discretization that is periodic about a constant ΔE . The discretizations of the time of flight functions trend (top to bottom) toward a better approximation to the equivalent differentials as the number of segments increases. A major goal of a regularizing mechanism in orbital mechanics

is to achieve a constant level of error across a Keplerian orbit when using fixed steps. Figure 4.4 shows the poor performance of the time case (no regularization) with a seven to eight order-of-magnitude variation in the energy error, while the regularized cases show a two to five order-of-magnitude variation. For columns two through four in Fig. 4.4, both the PST and DST behave similarly, decreasing the energy error as segment count increases. Overall, this numerical experiment demonstrates that a forward propagation can retain time as the independent variable for the dynamics if flight time functions are used to approximate regularizing behavior.

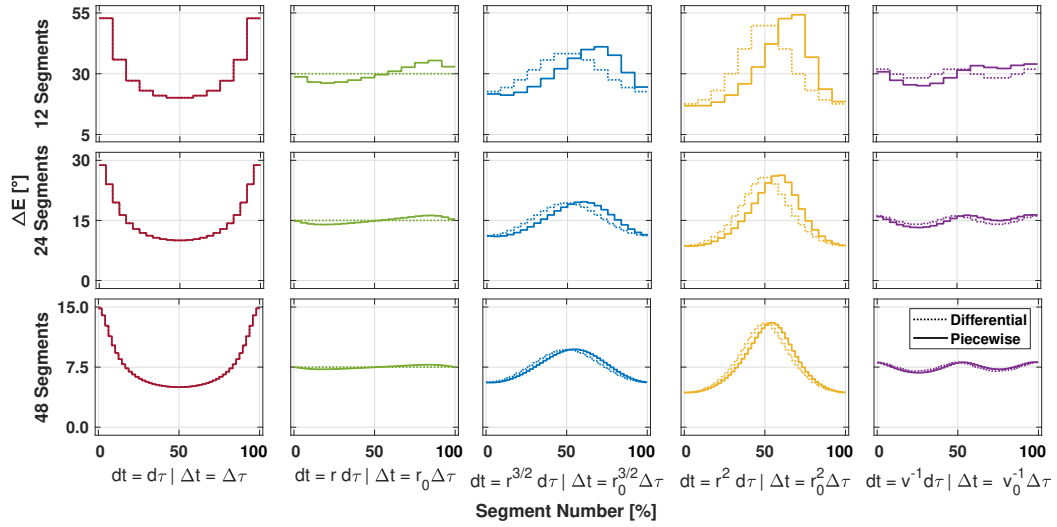


Figure 4.3: ΔE vs. Segment Number Percentage: various differentials and their equivalent piecewise flight time functions for a Keplerian orbit. $(a, e) = (5.0 \text{ DU}, 0.5)$; $(1 \text{ rev}, 10\pi\sqrt{5} \text{ TU TOF})$.

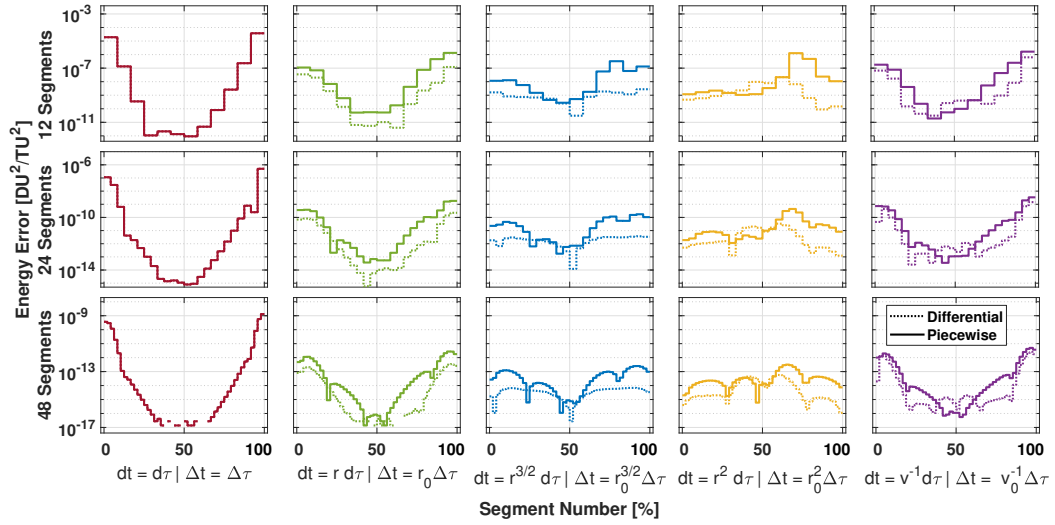


Figure 4.4: Energy Error vs. Segment Number Percentage: various differentials and their equivalent piecewise flight time functions for a Keplerian orbit. $(a, e) = (5.0 \text{ DU}, 0.5)$; (1 rev, $10\pi\sqrt{5} \text{ TU}$ TOF).

4.4 Trajectory Optimization: Piecewise Sundman Transformation

In this section, a trajectory optimization problem is formulated in Section 4.4.1 and followed by results in Section 4.4.2. The formulation of the optimal control problem stands to benefit from the PST and is used in four different examples. The formulation describes (i) the minimized cost function, (ii) the decision variables, (iii) the inclusion of constraints, (iv) two optimization techniques (EBVP and IVP)—connecting the decision variables and the cost, (iv) mass and thrust histories, and (v) a custom-built unconstrained optimizer. The cost and PST partials are included in the appendix.

The four examples and their solution to the optimal control problem illustrate PST behavior. Large-scale problems with many segments are directly solved

that may otherwise be intractable. All locally optimal solutions are fixed-state to fixed-state, but orbit-to-orbit scenarios are simple extensions. In example 1, the position coefficient of the PST varies to model a simple Keplerian orbit. The numerical analysis is an extension of the previous propagation problem in Section 4.3. The position coefficient $r_* = (r_0 + r_f)/2$ best approximates the DST by averaging between the discretization behavior of the initial and final forms: $r_* = r_0$ and $r_* = r_f$. In example 2, a comparison is presented between the EBVP and IVP techniques: two low-fidelity, impulsive maneuver solution methods for the optimal control problem. The following technique descriptions are repeated here for convenience. The IVP technique is a state-of-the-art shooting method, using a Keplerian solver for sequentially dependent segments and impulsive maneuvers for control. The decision variables are the impulsive maneuvers. The EBVP technique uses embedded Lambert problems, independent segments, and positions as decision variables. The impulsive maneuvers are dependent on the positions. The relatively faster and more robust performance of the EBVP technique is enabled by the PST. In example 3, the first two sub-examples vary only a and e , respectively, to demonstrate that the PST produces a discretization that can approximate equal steps in ΔE in both cases. Then in the third sub-example, a , e , and inclination i significantly change for a large problem that contains 12,287 segments (12,288 nodes) and 512 revs. In example 4, optimal transfers are found from a circular to a hyperbolic orbit using the same $\Delta\tau$ for every segment of a solution. For a particular solution, the total time of flight $\Delta t_{\text{total},s}$ is fixed and a homotopy is performed from an energy to a fuel optimal solution.

4.4.1 Problem Formulation

The overview of the problem is

$$\begin{array}{llll}
 \text{minimize} & \text{cost function} & J = k \sum_{i=1}^n \Delta v_i + (1 - k) \sum_{i=1}^n \frac{1}{2} \Delta v_i^2 & a \\
 \Delta v_i(\mathbf{x}_d) & & & \\
 \text{subject to} & \text{dynamics} & \mathbf{a} = -\frac{\mu}{r^3} \mathbf{r} & b \\
 & \text{boundary conditions} & \mathbf{x}_{1-} = \mathbf{x}_{1-,s}, \mathbf{x}_{n+} = \mathbf{x}_{n+,s} & c \\
 & \text{path constraints} & \Delta \mathbf{r}_i = \mathbf{r}_{i+} - \mathbf{r}_{i-} = \mathbf{0} & d
 \end{array}$$

^afuel plus energy | ^btwo-body | ^cfixed-state to fixed-state | ^dposition continuity

As shown in Eq. (3.1) of Chapter 3 and repeated here for convenience, the minimized cost J is a weighted sum of impulsive maneuver magnitudes, Δv_i , and their square, Δv_i^2 :

$$J = k \sum_{i=1}^n \Delta v_i + (1 - k) \sum_{i=1}^n \frac{1}{2} \Delta v_i^2$$

where the Δv_i are velocity discontinuities between each segment or at each node from node 1 to n along the spacecraft trajectory, as described in Chapter 2. The weighted sum using k enables a homotopy where $k = 1$ corresponds to fuel optimal trajectories and $k = 0$ corresponds to energy optimal trajectories. If $0 < k < 1$, the result is a hybrid between the two. When selecting k , it is a fixed quantity for a particular optimal solution, but can vary for different neighboring optimal solutions. Fuel optimal trajectories minimize the use of fuel mass through the rocket equation shown in Eq. (3.4). Energy optimal solutions minimize the additional kinetic energy of the Δv 's.

The cost J in Eq. (3.1) is augmented to include constraints. Each constraint has quadratic form and a penalty coefficient, turning the cost function into a penalty function. The specified constraints are total time of flight $\Delta t_{\text{total},s}$, final position

$\mathbf{r}_{n,s}$, and final velocity $\mathbf{v}_{n,s}$. The penalty function is

$$\begin{aligned} \tilde{J} = J &+ \frac{1}{2}\tilde{c}_{\Delta t}(\Delta t_{\text{total}} - \Delta t_{\text{total},s})^2 \\ &+ \frac{1}{2}\tilde{c}_r(\mathbf{r}_n - \mathbf{r}_{n,s})^T(\mathbf{r}_n - \mathbf{r}_{n,s}) + \frac{1}{2}\tilde{c}_v(\mathbf{v}_{n+} - \mathbf{v}_{n+,s})^T(\mathbf{v}_{n+} - \mathbf{v}_{n+,s}) \end{aligned} \quad (4.12)$$

where $\tilde{c}_{\Delta t}$, \tilde{c}_r , and \tilde{c}_v are the penalty coefficients for total flight time, final position, and final velocity, respectively. Note $\mathbf{r}_{n,s}$ and $\mathbf{v}_{n,s}$ are always satisfied at every iteration when using the EBVP technique. In contrast, the IVP technique must enforce these constraints: $\mathbf{r}_{n,s}$ and $\mathbf{v}_{n+,s}$.

The decision state \mathbf{x}_d for the EBVP technique is

$$\mathbf{x}_d = \left[\Delta\tau, \mathbf{r}_2^\top, \dots, \mathbf{r}_{n-1}^\top \right]^\top \quad (4.13)$$

where $\Delta\tau$ is equal for each segment and position \mathbf{r}_i is from node 2 to $n-1$, not 1 to n , so the length is $3n-5$. The initial position \mathbf{r}_1 and final position \mathbf{r}_n are fixed along with the respective initial velocity \mathbf{v}_{1-} and final velocity \mathbf{v}_{n+} for fixed-state to fixed-state optimal trajectories, i.e. $\mathbf{r}_1 = \mathbf{r}_{1,s}$, $\mathbf{r}_n = \mathbf{r}_{n,s}$, $\mathbf{v}_{1-} = \mathbf{v}_{1-,s}$, and $\mathbf{v}_{n+} = \mathbf{v}_{n+,s}$. In other words, the initial and final fixed-states are defined as $\mathbf{x}_{1-} = [\mathbf{r}_1^\top, \mathbf{v}_{1-}^\top]^\top$ and $\mathbf{x}_{n+} = [\mathbf{r}_n^\top, \mathbf{v}_{n+}^\top]^\top$, respectively. The decision state \mathbf{x}_d for the IVP technique is

$$\mathbf{x}_d = \left[\Delta\tau, \Delta\mathbf{v}_1^\top, \dots, \Delta\mathbf{v}_n^\top \right]^\top \quad (4.14)$$

where $\Delta\tau$ is still the independent propagation variable, but the impulsive maneuvers, $\Delta\mathbf{v}_i$, are now independent from node 1 to n . The length is $3n+1$.

For examples 1–4, the cost function Eq. (3.1) and penalty function in Eq. (4.12) depend on the decision variables in Eqs. (4.13) and (4.14) through the use of

the EBVP or IVP technique. The EBVP technique, using Eq. (4.13) as input, uses a fast, robust, and publicly available [17] Lambert solver [16]. This interpolated Lambert solver provides a fixed, one iteration solution for unperturbed two-body dynamics. The partials of this EBVP solver are directly found using the interpolated solution, but the variational state transition matrix and the Lagrange \mathbf{f} and \mathbf{g} coefficients can also be used to compute the partials, as illustrated in Ref. [105]. For example 2 only, the IVP technique uses Eq. (4.14) and an analytic, universal variable solution to Kepler’s problem [79] (See Sect. 4.3–4.5, pp. 191–212.). No iterative routine is needed because the independent variable is the universal variable, $\Delta\tau$ in this current work, not flight time. The single leg of the IVP technique is comprised of many sequential, position continuous segments with impulsive maneuvers at each node. Only the position and velocity continuity at the last node must be enforced by the optimizer.

Similar to Chapter 3, for the optimization, a custom-built, gradient-based, unconstrained optimizer (FMINUNC_{UT}) employs a Broyden [160]-Fletcher[161]-Goldfarb[162]-Shanno[163] (BFGS) search direction, and a line search using a golden ratio method with quadratic interpolation at every major iteration. FMINUNC_{UT} takes advantage of fast BLAS [169] matrix and vector operations and the parallelizable EBVPs for examples 3 and 4. Parallelization is not necessary for example 1 because of the short runtime and example 2 does not implement it because the single-shooting version of the IVP technique is not parallelizable. Multi-shooting IVP techniques are parallelizable but position and velocity continuity must be enforced. Examples 1, 3, and 4 satisfy the inequality $\|dJ/d\mathbf{x}_d\| \leq 10^{-6}$. Example 2 satisfies the following three inequalities (Gill et al. [209], pp. 306–307): $(J_{\text{prev}} - J)/(1 + |J|) < 10^{-10}$, $\|\mathbf{x}_{d\text{prev}} - \mathbf{x}_d\|/(1 + \|\mathbf{x}_d\|) < \sqrt{10^{-10}}$, and $\|dJ/d\mathbf{x}_d\|/(1 + |J|) \leq \sqrt[3]{10^{-10}}$, where

$(\)_{\text{prev}}$ means the previous iteration and no subscript means current. For more details on FMINUNC_{UT}, see Ref. [173].

Finally, as mentioned in Chapter 3, specifically Section 3.2, mass m_{i+} and thrust $T_{i(i+1)}$ histories are defined by Eq. (3.4) and Eq. (3.5), respectively, and are computed after optimization. The equations are repeated here for convenience.

$$m_{i+} = m_{i-} e^{-\Delta v_i/c} \qquad T_{i(i+1)} = m_{i+} \frac{\Delta v_i}{\Delta t_{i(i+1)}}$$

4.4.2 Optimization Results

Example 1: Effect of the Position Coefficient of the PST

Example 1 is an extension of the previous propagation of an eccentric Keplerian orbit in Section 4.3. The solution to this BVP is trivial. The fuel optimal solution is to apply no impulsive maneuvers to model a Keplerian orbit. Importantly though, analyzing the average position coefficient $r_* = (r_0 + r_f)/2$ of the PST for a simple Keplerian orbit is only possible with an EBVP technique. Furthermore, for comparison, the initial and final position coefficients, $r_* = r_0$ and r_f , are also included using the EBVP technique. The inclusion of the DST from Section 4.3 is used as a reference, where fixed-steps in $\Delta\tau$ are proportional to ΔE . See Fig. 4.3, second column from the left, and in the middle row for this particular DST case. As a reminder, no optimization is performed for this DST case (or differential in Fig. 4.5). All four scenarios (3 position coefficients for the PST + 1 DST) use a Sundman exponent of $\alpha = 1$ to efficiently model a Keplerian orbit. The initial and final fixed-states are the same at the true anomaly of 0° and 360° for a Keplerian orbit defined by $(\mathbf{a}, \mathbf{e}) = (5.0 \text{ DU}, 0.5)$ and discretized into 24 segments. See the right subfigure of Fig. 4.1 for a similar trajectory.

As in the propagation problem of Section 4.3, ΔE varies depending on the form of the discretization. Fig. 4.5 shows the evolution of ΔE for the three different choices of position coefficients. The observations are (i) the average position coefficient of the PST, i.e $r_* = (r_0 + r_f)/2$, gives a closer approximation to a constant ΔE than the initial and final position coefficients, (ii) all three position coefficients r_* are periodic over one orbit, and (iii) the initial and final position coefficients, $r_* = r_0$ and r_f , mirror each other's behavior. The ΔE of the outbound journey away from periapsis underestimates and overestimates a constant ΔE for the initial and final position coefficient, respectively. For the inbound journey towards periapsis, the reverse is true. The ΔE history of the initial position form of the discretization using an EBVP technique is similar to the propagation problem. Again, see Fig. 4.3, second column from the left, and middle row for the PST case where $r_* = r_0$ and $\alpha = 1$. Energy error is not included here because of the mass-leak set to 10^{-4} , commonly used to avoid singularities in the partials that occur when impulsive maneuvers approach zero (see appendix). The PST using $r_* = (r_0 + r_f)/2$ is used for examples 2, 3, and 4 hereafter.

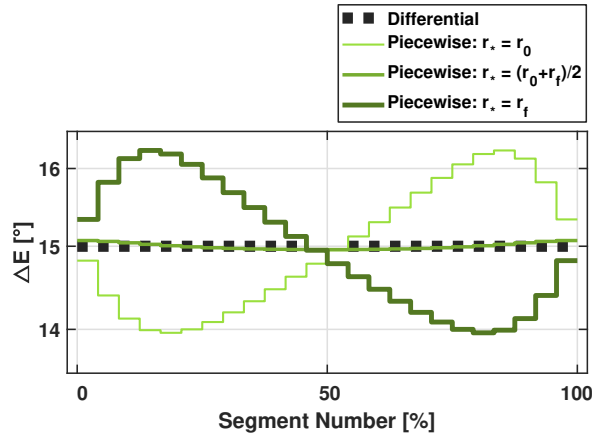


Figure 4.5: Example 1: ΔE vs. Segment Number Percentage: Discretization effect of three position coefficients r_* of the PST. $(a, e) = (5.0 \text{ DU}, 0.5)$; (24 segments, 1 rev, $10\pi\sqrt{5}$ TU TOF).

Example 2: Effect of the PST on Optimization Performance

Example 2 compares the optimization performance of two low-fidelity, impulsive maneuver techniques: the IVP vs. EBVP technique. The PST enables the EBVP technique and IVP techniques are found in state-of-the-art software such as MALTO[90], GALLOP[210], and PyKEP[211]. For a fair comparison, both techniques use the same unconstrained FMINUNC_{UT} optimizer, notably using a BFGS search direction and an unsophisticated penalty function to enforce constraints. The position coefficient for the PST is $r_* = r_0$ and both the PST and DST use exponent $\alpha = 1$. Defining a stage, the FMINUNC_{UT} optimizer solves one unconstrained problem using the penalty function Eq. (4.12). Many sequential stages are needed to enforce the constraints below a small tolerance. After one unconstrained problem is solved for a stage, the penalty coefficients are each updated by a factor of five if their respective constraint is not satisfied. In particular, the tolerances for total flight time, final position, and final velocity are $|\Delta t_{\text{total}} - \Delta t_{\text{total},s}| < 10^{-3}$, $\|\mathbf{r}_n - \mathbf{r}_{n,s}\| < 10^{-7}$, and $\|\mathbf{v}_n - \mathbf{v}_{n,s}\| < 10^{-4}$, respectively. These tolerances are slightly more conservative than 1 s, 1 m, and 1 m/s in metric units if the canonical units are defined as 1 DU = 6,378,000 m and 1 TU = 806.8 s. Additionally, and by design, the first stage is time-free, but afterwards, the second and later stages enforce a total flight time that is 10% greater than the time-free flight time. The position and velocity constraints are implicitly enforced with the EBVP technique but must be enforced with the IVP technique. Overall, there are seven constraints for the IVP technique [total flight time (1), final position (3), and final velocity (3)] and one constraint for the EBVP technique [total flight time (1)].

The energy optimal transfers [$k = 0$ in Eq. (3.1)] are from a fixed-state to fixed-state, defined by $(\mathbf{a}, \mathbf{e}) = (1.0 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.6)$ at true anomaly

0° for both orbits. The number of revs is varied from 1 to 30 where each rev contains initially 24 segments per rev via a Keplerian orbit as the initial guess: $(\mathbf{a}, \mathbf{e}) = (4.9 \text{ DU}, 0.7959)$. The IVP and EBVP techniques both use as an initial guess $\Delta\tau = 2\pi/24$ for the propagation variable, but the IVP technique uses zero-magnitude, impulsive maneuvers for decision variables, while the EBVP technique uses the position variables of the Keplerian orbit. Importantly, this initial guess was designed to accommodate the sensitivity of IVP technique. Sometimes when iterating on the $\Delta\tau$ variable with the IVP technique, the solution can jump to a neighboring rev solution without safeguards, namely a good initial guess. The EBVP technique is generally more robust against bad initial guesses (See Section 4.5 of Ref. [173]).

Fig. 4.6 presents the performance results of this study. The top-left subplot of Fig. 4.6 shows gradual speed up of runtime per iteration for segment counts up to approximately 400 segments or about 17 revs. After 400 segments or the kink in the curve, the cost quadratically increases because of the BFGS search direction, consistent with previous work [173] (See Fig. 4.). While the runtime per iteration of the IVP technique is always less than the EBVP technique, the total runtime, shown in the top-right, is more because more iterations are needed. See bottom-right. In fact, past about 360 segments or 15 revs, the IVP technique is runtime intractable, taking minutes to optimize when the EBVP technique takes seconds. Furthermore, parallelization improvements are available, but not implemented here, for the EBVP technique because each segment is independent and not sequentially dependent on previous segments like in the IVP technique. Lastly, the penalty coefficient updates of the bottom-left subplot of Fig. 4.6 track the number of times a penalty coefficient is updated by a factor of five until a tolerance in the particular constraint is satisfied.

The total amount of stages is equal to the max number of penalty coefficient updates needed overall, accounting for the time-free solution of stage one.

The illustrated example in Figs. 4.7 and 4.8 is one data point of Fig. 4.6. Fig. 4.7 shows the discretization history of both the IVP and EBVP technique are approximately equivalent, while Fig. 4.8 reinforces the substantial increase in iterations of the IVP technique relative to the EBVP technique. Notice both techniques of this 240-segment, or 10-rev, solution need eight stages to satisfy the constraints. For this set of examples, the final position or total flight time constraint drives the need for more stages. However, the EBVP technique satisfies the flight time constraint in substantially fewer iterations and does not need to enforce position or velocity continuity explicitly with the optimizer. Also note, at the top of Fig. 4.8, two magnifications of the last three stages of the cost history are shown for both the IVP and EBVP technique. Both magnifications show a slight jump in cost as the penalty coefficients of the constraints are increased.

In all, the EBVP technique, improved by the PST, performs substantially better than its counterpart, the IVP technique. The EBVP technique is faster for almost any segment count, save for the 24-72 segment range. This study also helps motivate example 3, an exploration of more efficient discretizations for transfers that substantially vary both semimajor axis and eccentricity by varying α . For this example 2, using exponent $\alpha = 1$, the PST produces a discretization of fixed-steps in $\Delta\tau$ that are not equivalent to equal steps in ΔE .

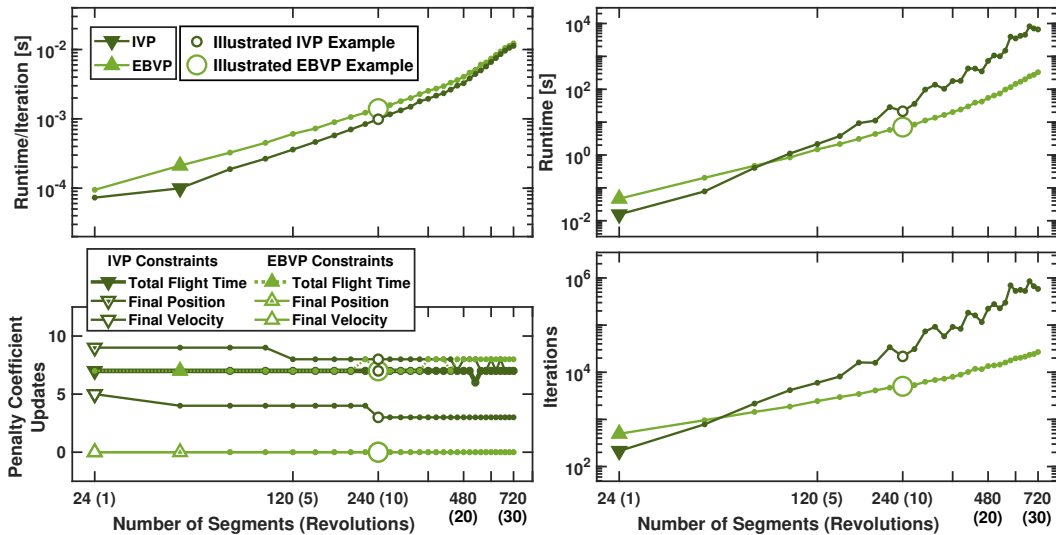


Figure 4.6: Example 2: Performance of the IVP and EBVP technique for an energy optimal transfer. Runtime, iteration count, and penalty coefficient updates are tracked vs. varying segment or rev counts. The particular example is illustrated in Figs. 4.7 and 4.8. The transfer is from (1.0 DU, 0.0) to (5.5 DU, 0.6), using (a, e).

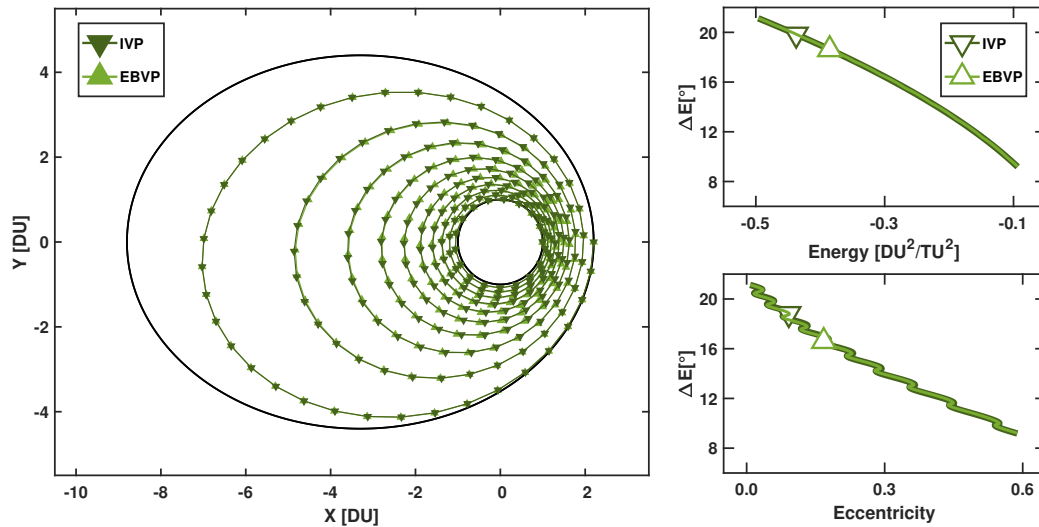


Figure 4.7: Example 2: Discretization comparison of the IVP and EBVP technique for a 240-segment or 10-rev energy optimal solution. The two discretizations of the trajectory (left) and the plots (right) for ΔE vs. energy and eccentricity are nearly identical. Energy here is $\mathcal{E} = -\mu/2a$. The transfer is defined by (a, e): (1.0 DU, 0.0) to (5.5 DU, 0.6) with a specified 211.0356 TU TOF.

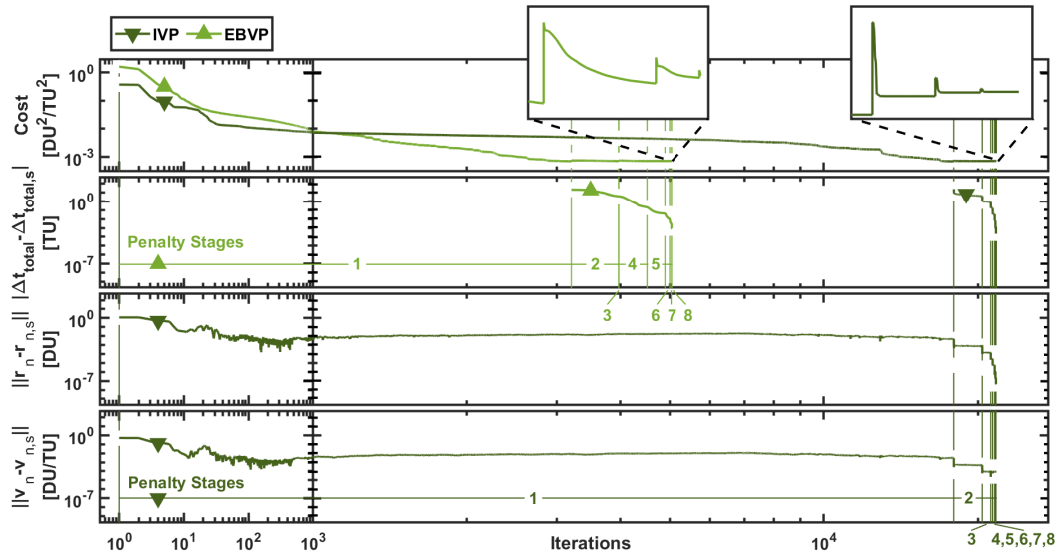


Figure 4.8: Example 2: Cost and constraints vs. iteration count for a 240-segment or 10-rev, energy optimal solution using the IVP and EBVP technique. Penalty stages are labeled and magnifications (top) of the cost for the last three stages are included. The transfer is defined by (a, e): (1.0 DU, 0.0) to (5.5 DU, 0.6) with a specified 211.0356 TU TOF.

Example 3: Effect of the Exponent of the PST

By varying the α , example 3 demonstrates how the PST enables a larger design space for direct spacecraft trajectory optimization problems that use many segments for the discretization. The first and second sub-examples of example 3 significantly change a and e in Figs. 4.9 and 4.10, and the third sub-example significantly changes a, e, and i in Figs. 4.11 to 4.13. All these sub-examples minimize energy [$k = 0$ in Eq. (3.1)] and use the EBVP technique.

The first two sub-examples are time-fixed at 220 TU and use 10 revs and 240 segments total. Notably, the rev count is naturally enforced when minimizing energy, a quadratic cost, if the initial guess is of the same rev count. These two sub-examples are orbital transfers from a fixed-state to a fixed-state at a true anomaly

of 0.0° . The transfers are defined by $(a, e) = (1.0 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.0)$ and $(5.5 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.6)$, respectively. Both sub-examples achieve approximately equal steps in ΔE (24 steps per rev) if the trajectory approximately changes a only or e only using $\alpha = 3/2$ or 1, respectively, for the PST. Moreover, for the second sub-example that changes e only, $\alpha = 3/2$ also performs well where the initial to final segment count per rev is 23 to 26, respectively, noting that the initial and final segment count per rev of all examples in this work create an approximate range of possible values. Thus, the use of $\alpha = 3/2$ benefits transfers changing both a and e with approximately constant changes in segment count per rev. Other segment counts of the first and second sub-examples are presented in Table 4.3,

For these optimal control problems, equal steps in ΔE via the PST produces two notable benefits. First, any trajectory design that significantly varies a and e with impulsive maneuvers produces a higher-fidelity approximate thrust or control model over the entire trajectory. Second, a unique problem for spacecraft trajectory optimization using EBVPs is transfer-plane ambiguity for a single segment. Two transfer-plane ambiguities that can be encountered are the 0° transfer with non-zero flight time, and the 180° transfer. These transfer-plane ambiguities are mitigated through the use of the PST. For example, and for performance emphasis, the third sub-example is not feasible without the PST using $\alpha = 3/2$ and $r_* = (r_0 + r_f)/2$ because the 0° and 180° transfer-plane ambiguities are encountered if equal steps in time are used. Equal steps in time encourage 180° and 0° transfer ambiguities near and far from the gravitational body, respectively. Similar, potentially problematic transfer-plane behavior is shown in the top left sub-figure of Fig. 4.9 if fewer segments were used for this trajectory.

For the final sub-example, the problem is time-free and large, using 12,287

segments (12,288 nodes) and 512 revs to transfer from a fixed-state to fixed-state at the periapsis of an initial and final orbit defined by $(a, e, i) = (1.0 \text{ DU}, 0.0, 30^\circ)$ to $(5.5 \text{ DU}, 0.6, 0^\circ)$. The ΔE is periodic about approximately 15° . The segment count on the first rev is between 23 and 24 (rounded down or up, respectively) and the segment count on the last rev is between 25 and 26, a $1.09\times$ increase approximately predicted by dividing the final and initial τ_p periods: $\tau_{p,f}/\tau_{p,0} \approx 6.82/2\pi \approx 1.09\times$. The optimization of the final sub-example is performed with a continuation method. The first optimal solution is from a coasting trajectory on the initial orbit using 12 nodes for 2 revs. The next eight solutions optimize a modified, previously optimal solution. The modification doubles the rev count from the previous solution, so the node count and total time of flight double, but not the number of nodes per rev. The last two solutions double the number of nodes per rev, but keep fixed the number of revs. Toward the end of the continuation method, Table 4.4 shows the runtime is dominated by the n^2 computations, necessary for a BFGS search direction calculation. The total runtime is 39.4 hours where 89.4% of it is from the last doubling. However, inspection of Table 4.4 reveals significantly shorter runtimes for smaller node (or segment) counts, e.g. 4.2 hours total runtime for a 6144-node (6143-segment) trajectory. As a final note, conjugate gradient search directions to avoid the n^2 computations of the BFGS search direction are investigated, but found to be insufficient for this challenging problem, taking more total runtime.

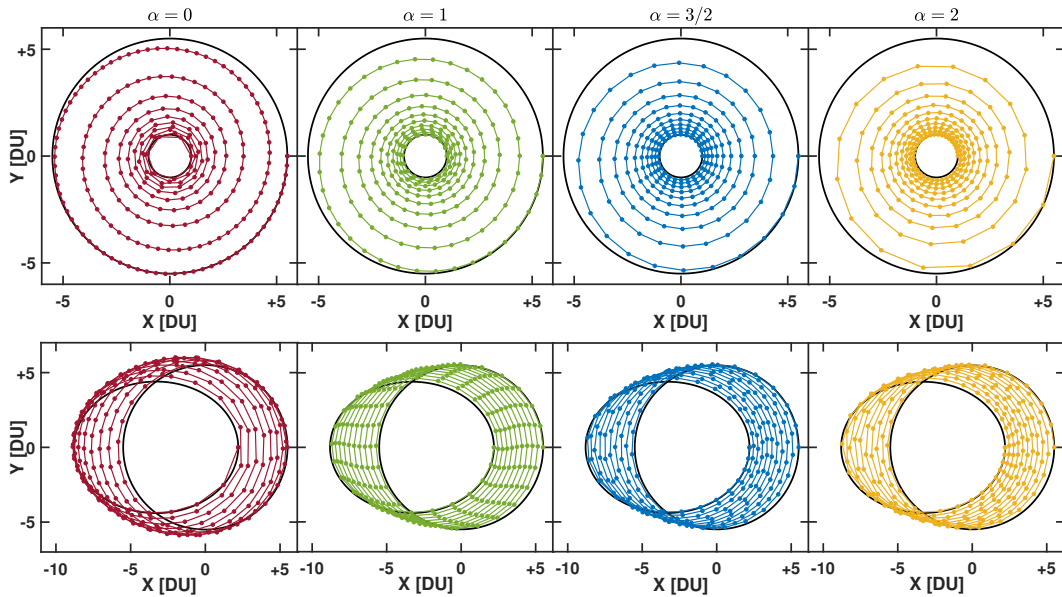


Figure 4.9: Example 3: Trajectories that primarily vary a only (top) and e only (bottom) for various PST exponents α at a fixed time of flight. $(a, e) = (1.0 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.0)$ [top], $(5.5 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.6)$ [bottom]; (240 segments [both], 10 revs [both], 220 TU TOF [top], 820 TU TOF [bottom]).

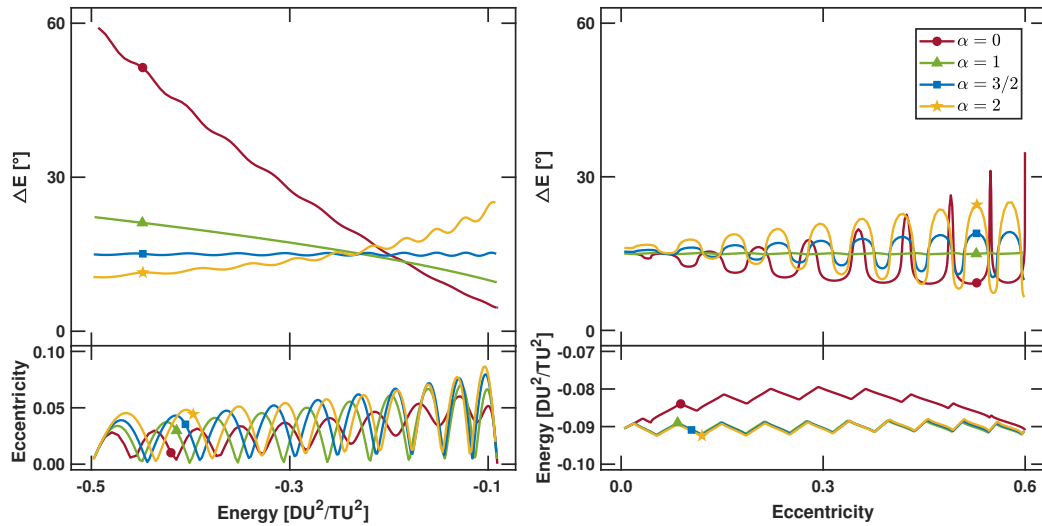


Figure 4.10: Example 3: Histories of ΔE , e , and \mathcal{E} for the spacecraft trajectories of the top row (left) and bottom row (right) of Fig. 4.9.

Table 4.3: Segment count of initial and final rev of example 3 shown in Figs. 4.9 and 4.10.

	Regularization Scenario	Time $\alpha = 0$	Sundman $\alpha = 1$	Sundman $\alpha = 3/2^b$	Sundman $\alpha = 2$
Orbit Transfer for a Only, $(a, e) = (1.0 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.0)$					
Number of Segments	Initial Rev	6	16	24	32
	Final Rev	73	36	24	16
	Segment Ratio	$12.2 \times^a$	$2.3 \times$	$1.0 \times$	$0.5 \times$
Orbit Transfer for e Only, $(a, e) = (5.5 \text{ DU}, 0.0)$ to $(5.5 \text{ DU}, 0.6)$					
Number of Segments	Initial Rev	24	24	23	22
	Final Rev	24	24	26	28
	Segment Ratio	$1.0 \times$	$1.0 \times$	$1.1 \times$	$1.3 \times$

^aSegment ratio is final divided by initial segment count. | ^bHighlighted for emphasis.

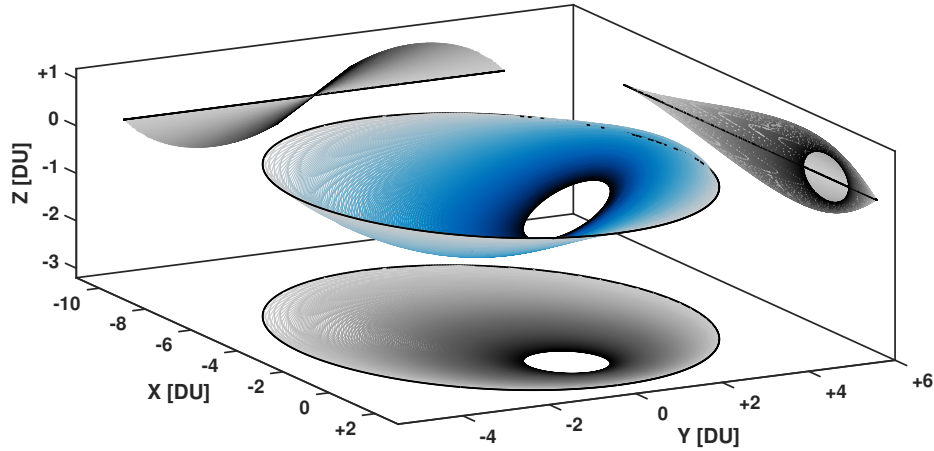


Figure 4.11: Example 3: Time-free, energy optimal spacecraft trajectory that significantly changes a , e , and i . $(a, e, i): (1.0 \text{ DU}, 0.0, 30^\circ)$ to $(5.5 \text{ DU}, 0.6, 0^\circ)$; (12288 nodes, 512 revs, 12814 TU TOF).

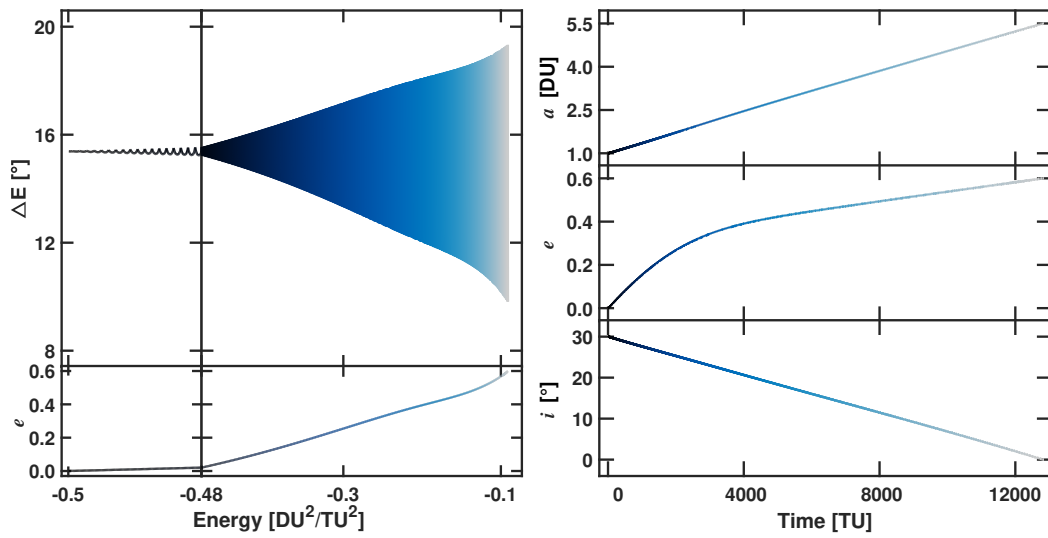


Figure 4.12: Example 3: History of ΔE and e vs. \mathcal{E} (left), as well as a , e , and i vs. t (right) for Fig. 4.11.

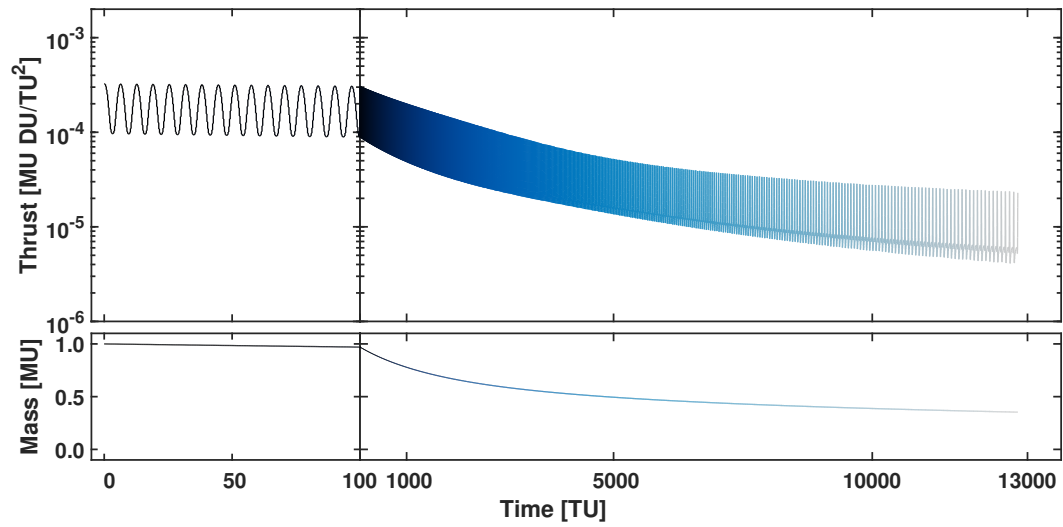


Figure 4.13: Example 3: History of T and m vs. t for Fig. 4.11. Note $m_{n+} = 0.353$ MU.

Table 4.4: Runtimes for the continuation method for the third sub-example of example 3 in Figs. 4.11 to 4.13.

Case	Runtime	Case	Runtime
Coast \rightarrow $12n, 2r$	< 0.1 s	Prev. \rightarrow $768n, 128r$	70.7 s (1.2 m)
Prev. \rightarrow $24n, 4r$	0.1 s	Prev. \rightarrow $1536n, 256r$	1090.1 s (18.2 m)
Prev. \rightarrow $48n, 8r$	0.2 s	Prev. \rightarrow $3072n, 512r$	3704.4 s (61.7 m, 1.0 h)
Prev. \rightarrow $96n, 16r$	0.4 s	Prev. \rightarrow $6144n, 512r$	10161.2 s (169.4 m, 2.8 h)
Prev. \rightarrow $192n, 32r$	1.7 s	Prev. \rightarrow $12288n, 512r$	126730.5 s (2112.2 m, 35.2 h)
Prev. \rightarrow $384n, 64r$	14.3 s	TOTAL	141773.7 s (2362.9 m, 39.4 h)

n is for nodes and r is for revs.

Example 4: Circular to Hyperbolic Orbit Transfer using the PST

A spacecraft trajectory can transfer through the parabolic boundary from a circular to a hyperbolic orbit using the PST with $\alpha = 3/2$ for an efficient discretization. Traversing this boundary with the PST draws a parallel to one of the original purposes of the DST. The DST was a time transformation that produced a universal anomaly for all Keplerian conics from circular to hyperbolic. See page 191 of Bate et al. [79]. To demonstrate this transfer, Fig. 4.14 shows the energy cost vs. total time of flight for different energy optimal trajectories from a circular to hyperbolic orbit, $(a, e) = (1.0 \text{ DU}, 0.0)$ to $(-2.4142 \text{ DU}, 1.4142)$. Each solution for Fig. 4.14 is characterized by a different number of revs between 1 and 10, and a different time of flight between 10 and 1000 TU. In particular, a time-free solution is most obvious near the minimum of the 1-rev curve at approximately 30 TU total time of flight. Other energy optimal solutions using more than 1000 TU total time of flight are not shown, nor desirable, because the last rev approaches a a of infinity. Each solution of Fig. 4.14 contains 240 segments and optimizes in a few seconds. Overall hundreds of time-fixed solutions are solved in about one hour. One solution, termed ‘plotted example’, is illustrated in Fig. 4.15 as an energy optimal solution and a homotopy is

performed to the fuel optimal solution per Eq. (3.1). For this example, the evolution of \mathcal{E} and \mathbf{e} , as well as $T_{i(i+1)}$ and m_i , with respect to t_i and node i are shown in Fig. 4.16 and Fig. 4.17, respectively. Note the straight lines (red) attached to the trajectory in Fig. 4.15 are the Δv 's for this solution. The homotopy places the large Δv 's of the fuel optimal solution ($k = 1$) in the center of the smaller Δv 's of the energy optimal solution ($k = 0$) for the right-half figures that plot node history. The squares in Fig. 4.15 are at the initial time of a hyperbolic segment. There are one and nine hyperbolic segments for the fuel and energy optimal solutions, respectively.

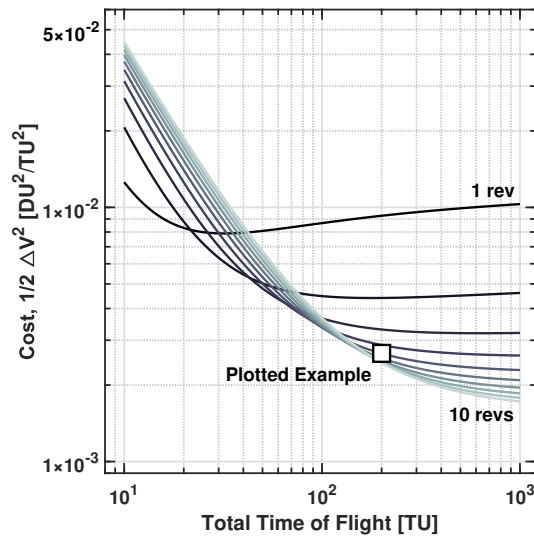


Figure 4.14: Example 4: J vs. $\Delta t_{\text{total},s}$: Energy optimal solutions with varying total flight times for 1 to 10 revs. The square is the energy optimal, plotted example in Figs. 4.15 to 4.17.

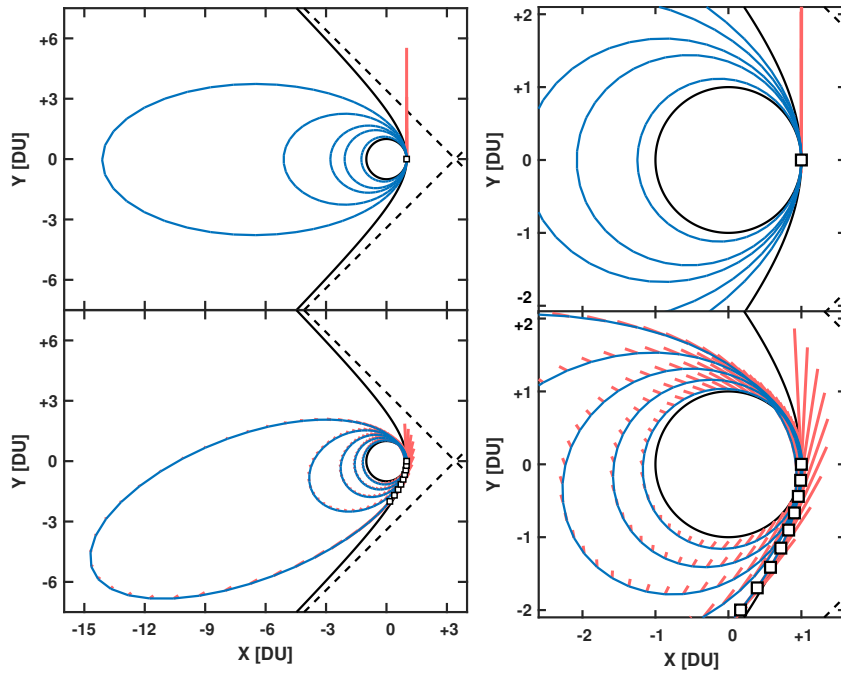


Figure 4.15: Example 4: Fuel (top) and energy (bottom) optimal spacecraft trajectories plotted with the $\Delta\mathbf{v}$ (red lines) for a circular to hyperbolic orbit transfer. The squares are at the initial time of a hyperbolic segment and the right side is a magnification. (a, e): (1.0 DU, 0.0) to (-2.4142 DU, 1.4142); (241 nodes, 5 revs, 200 TU TOF) [both].

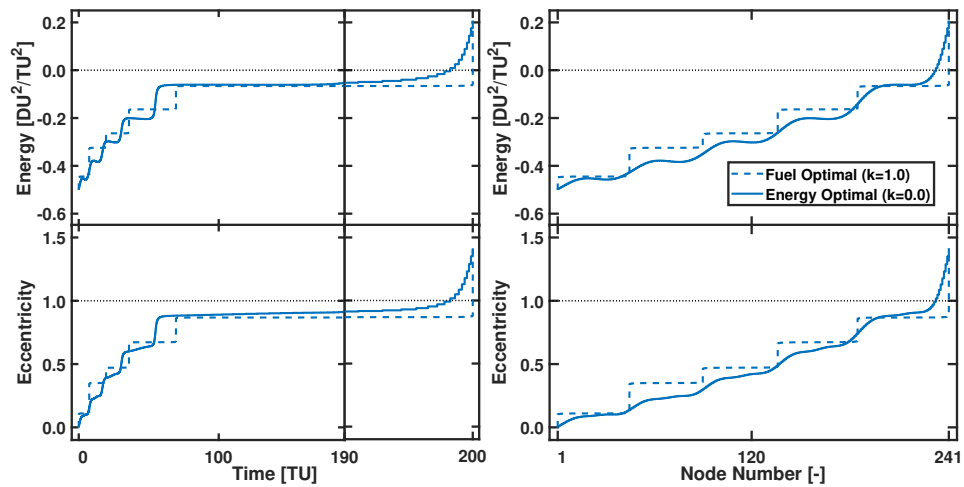


Figure 4.16: Example 4: t (left) and node i (right) history for \mathcal{E} and e for energy and fuel optimal spacecraft trajectories in Fig. 4.15.

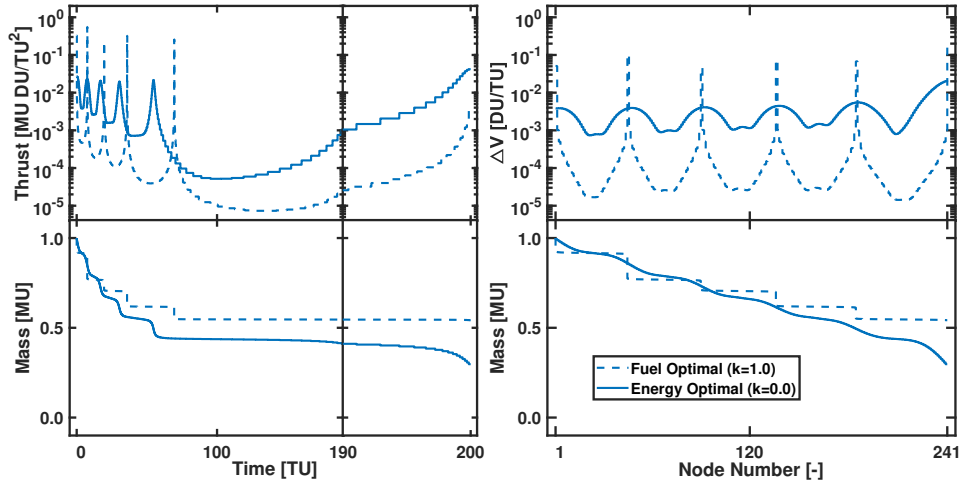


Figure 4.17: Example 4: t (left) and node i (right) history for the energy and fuel optimal spacecraft trajectories in Fig. 4.15. The left and right side show the evolution of T and m , as well as Δv and m , respectively. Note $m_{n+} = 0.542$ and 0.297 MU for $k = 1.0$ and 0.0 , respectively.

4.5 Conclusions

For a trajectory discretized into many segments, a piecewise Sundman transformation produces flight time functions that approximately model the regularizing behavior of differential transformations, while retaining time as the independent variable for the dynamics. Two applications use these flight time functions: a forward propagation problem to numerically integrate a Keplerian orbit, and four optimization problems that use many embedded Lambert boundary value problems for two-body dynamics. In the propagation problem, the functions either approximate arc-length or the Sundman transformation. The functions with piecewise-constant coefficients trend toward the differential transformation as the number of segments of a trajectory increases. In the optimization problems, the piecewise Sundman transformation improves low-fidelity, spacecraft trajectory optimization. The piecewise transformation enables the embedded Lambert technique to outperform a state of the art that

uses an initial value technique. The similar, low-fidelity, initial value technique uses many solutions to the Kepler problem instead of the Lambert problem for each segment. The embedded Lambert problem technique is faster for problems larger than 72 segments. In fact, problems bigger than approximately 360 segments are runtime intractable for the initial value technique. At 360 segments, the embedded Lambert technique converges in seconds, while the initial value technique takes minutes.

Additionally, in the optimization problems, it is shown spacecraft trajectories using approximate continuous thrust in a two-body system benefit from an exponent of $3/2$ because the transfers significantly change semimajor axis and eccentricity. This $3/2$ exponent produces a periodic discretization about a fixed-step size approximately proportional to eccentric anomaly. Other exponents, particularly 1 and 2, fail to produce a discretization proportional to eccentric anomaly if both semimajor axis and eccentricity significantly change.

These approximate time of flight functions are not exclusive to the presented scenarios and are especially practical for applications that require time as the independent variable. This piecewise Sundman transformation facilitates spacecraft trajectory optimization that uses the solution to many embedded Lambert problems, affording solutions to new classes of challenging problems. Besides enabling trajectories that can significantly vary semimajor axis and eccentricity, the trajectories can: (i) contain hundreds of revolutions and thousands of segments, (ii) transfer from circular to hyperbolic orbits, and (iii) be fuel or energy optimal.

Chapter 5

Direct-to-Indirect Mapping for Optimal Low-Thrust Trajectories

5.1 Introduction

Low-thrust electric propulsion for spacecraft enables efficient orbital transfers because of the high-specific-impulse engines [63]. The corresponding trajectories trade long flight times to minimize fuel. Optimization of these trajectories is difficult. The optimal control problems can include many unknowns, constraints, revolutions, non-smooth behavior, and a sensitive design space. Many solution approaches exist [63] to make low-thrust problems more tractable. One popular approach is to use an orbital element set for the coordinate system [212]. These approaches often also use orbital averaging, where the dynamics are simplified by averaging over a period to find near-optimal solutions. It is common to average the equinoctial elements [99] or the modified equinoctial elements [101, 213, 214, 215, 216], but not all approaches [103, 104] do. Scheel and Conway [99] in 1994 found minimum-time, low-thrust, many-rev trajectories around Earth, modeled with equinoctial elements in a Runge-Kutta parallel-shooting method for the NLP problem. Kluever and Ole-

son[214] in 1998 found near-optimal, minimum-time, low-thrust transfers too, but used extremal feedback control to maximize the time rates of averaged modified equinoctial elements. The research incorporated Earth shadowing, J2, and solar cell degradation. In 2016, Graham and Rao [103] solved similar minimum-time, many-rev problems in two types of phases. For the many single-phases, an initial guess for the trajectory is generated from the geometry of the shadow region. The total multi-phase optimal control problems use *hp*-adaptive Legendre-Gauss-Radau orthogonal collocation to find high-accuracy solutions. In 2020, Shannon et al. [104] paired a Q-law Lyapunov function with direct collocation, splitting the low-thrust, many-rev problem into two parts. The model used non-averaged modified equinoctial elements, efficiently finding near-optimal minimum-time or -mass solutions.

These approaches using modified equinoctial elements are fast and useful, but they are often complex and less physically intuitive. Instead, in this current work, the solution approach to solve the low-thrust, many-rev, two-body problem uses the simpler, physically-intuitive, singularity-free, Cartesian coordinates, while importantly maintaining efficiency standards. Furthermore, in this coordinate system, the solution approach is a *mapping* from a direct solution to its indirect equivalent. The mapping is for the class of minimum thrust-acceleration-squared problems governed by symmetric and skew-symmetric dynamics [217]. The direct solution is a coast-impulse trajectory and the indirect solution is a continuous-thrust trajectory. The coast-impulse model uses many Keplerian segments separated by many impulses (impulsive maneuvers) and is otherwise known as the Sims-Flanagan [77, 90] model.

A qualitative, but accurate visual of the mapping is presented in Fig. 5.1. Both trajectories are an optimal, thrust-acceleration-squared, Hohmann-like trans-

fer from two different circular orbits in the two-body problem. While the original motivation of this current work is to solve low-thrust, many-rev, two-body problems, this visual provides clarity on form. The impulses of the direct solution are approximately equivalent to the continuous-thrust profile, both multiplied by $\times 10$ for emphasis. Additionally, for the optimal solution, the direct and magnitude (but not units) of the thrust-acceleration vectors are exactly the same as the relative position and velocity vectors of a particular neighboring trajectory. This thrust-acceleration equivalency to the neighboring path is used for the mapping. In fact, this discrete, direct model converges to the indirect, continuous model as the number of segments and impulses increase, similar to how a finite sum is approximately equal to a continuous integral. This convergence makes the mapping remarkably

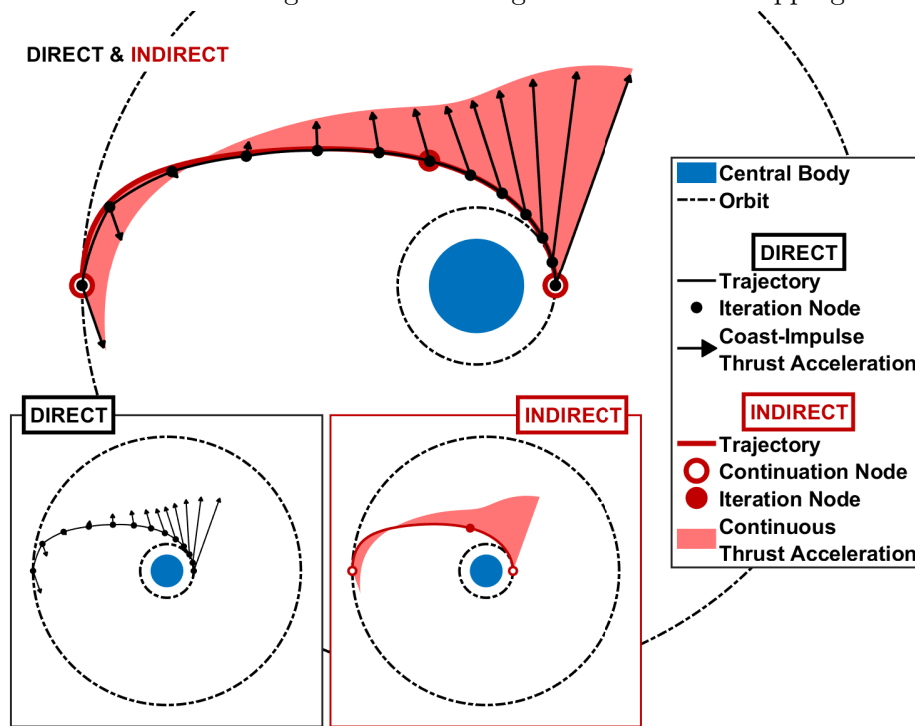


Figure 5.1: Hohmann-like transfer for a minimum thrust-acceleration-squared problem using a direct, coast-impulse model or indirect, continuous-thrust model.

simple, only needing as input a small set of positions, velocities, and flight times associated with any one time along the direct solution. The output is the nearly-optimal state and control for the indirect problem at this same time, efficiently updated to optimality to complete the mapping.

The direct-to-indirect mapping allows for the benefits of both models. Coast-impulse models are simple, fast, and reliable, notably implemented in the software packages MALTO [90], GALLOP [210], PyKEP [211], and EMTG [218, 219, 220], and Copernicus [221]. Some techniques to find coast-impulse solutions include (1) multi-shooting that enforces position continuity with either sequentially connected segments or as a constraint in a NLP; (2) collocation techniques that can be structured to assume position continuity, but dynamics are enforced with constraints; (3) differential dynamic programming [64, 222, 65, 66, 67], or (4) an inverse dynamics approach that uses many solutions to embedded Lambert problems [16, 17] and the velocity discontinuities between segments, as demonstrated by the authors [173, 223]. The mapping is indifferent to the source of the coast-impulse solution. Furthermore, the convenience and large radius of convergence of the direct model circumvents the hyper-sensitivity [56, 61] of the indirect model. The indirect model simultaneously enables the minimization of the size of the NLP problem through a shooting method [99] and improves the resolution of the state and control history.

The mapping determines the notoriously difficult co-states of the indirect problem via primer vector theory, where the linear space around an optimal trajectory is used but typically modeled with fewer impulsive maneuvers. Uniquely, the mapping takes advantage of Lambert’s problem to facilitate the modeling of the linear space. The first research using a primer vector with only a few impulsive maneuvers in a high-thrust application is by Lawden [44] (pp. 5–69) in 1963.

Prussing [224] in 1969 and Jezewski [111] in 1975 also used primer vector theory to optimize a spacecraft trajectory using a few impulsive maneuvers. Closer to the present, in 2012, Hou-yuan and Chang-yin [225] considered a simple optimal control problem that connects low thrust and the co-states coupled to the use of Lambert problems. In 2017, Restrepo and Russell [217] worked on a shadow trajectory [158] approach, leveraging the linear space around an optimal trajectory. In Restrepo and Russell's shadow-trajectory work, the forward propagation of the trajectory is efficiently performed as a low-thrust initial value problem with many sequential impulsive maneuvers. The initial co-states are connected to the physical linear variations to produce the optimal thrust direction for problems that assume constant thrust or bang-bang thrust.

The direct-to-indirect mapping in this research draws a parallel to the relationship between discrete Lagrange multipliers of direct NLP problems to the equivalent continuous co-states of indirect optimal control problems, extensively studied in literature [226, 227, 228, 229, 230]. The direct problems in this work can verify optimality by the approximation to the co-states determined from the mapping. In 1992, Enright and Conway [226] were the first to show that the Lagrange multipliers for the defects of direct transcription (Hermite interpolation with Simpson Quadrature or Runge-Kutta parallel shooting) are a discrete approximation to co-states of the continuous optimal control problem. Later in 2001, Fahroo and Ross [227] extended Enright and Conway's work by using a Legendre pseudospectral method to estimate the co-states to the same order of accuracy as the states. A little earlier in 2000, Hager [228] looked at convergence rates of various Runge-Kutta discretizations of optimal control problems, utilizing the equivalency of the Lagrange multipliers of the NLP to the co-states. Benson [229] developed a Gauss pseudospectral method

in 2006 that connected the KKT multipliers (Lagrange multipliers) of the NLP to an accurate estimate of the co-state at the Legendre–Gauss points. Mapping these Lagrange multipliers of the NLP to co-states was formalized by Gong et al. [230], presenting a set of sufficient conditions for pseudospectral methods in which a solution to the discretized optimal control problem converges to the continuous solution.

Other works to determine co-states include: Yam and Longuski [231] who explored different formulations of the impulsive-maneuver model coupled to Chebyshev polynomials to reduce the numbers of unknowns for low-thrust gravity-assist trajectories with optimization runtimes on the order of minutes to hours, depending on the formulation; as well as Lee and Bang [232] who proposed an initial guess structure for the co-states of fuel optimal spiral trajectories for planar, initially-circular, specific-energy targeting problems. Adjoint control transformations exist to translate from a more physical parameter space to the adjoint space (co-state space). For example, Ranieri and Ocampo [233] produced optimal spiral trajectories by estimating co-states in a two-step process. The research shows how to first convert from the thrust unit direction to the co-states, then use a curve fit extrapolation to produce longer spirals trajectories. Ayyanathan and Taheri [234] extend adjoint control transformations to map from Cartesian coordinates to modified equinoctial elements.

Lastly, thrust-acceleration-squared solutions can serve as an initial guess for practical, higher-fidelity problems. Many previous works combine indirect methods with a homotopy for high-fidelity modeling. Some homotopies in literature minimize total flight time [235, 103, 236], fuel [100, 237, 238], or energy [100, 237, 238] using realistic constraints. In 2004, Haberkorn et al. [100] produced an optimal, energy-to-fuel low-thrust transfer from low-Earth orbit to geosynchronous orbit (GSO) for

various thrust maximums. Haberkorn then continued this work in 2011 with the inclusion of eclipses [239]. In 2012, Jiang et al. [237] successfully used an energy-to-fuel homotopy for two globally-optimal, low-thrust, gravity-assist problems. In 2016, Zhao and Zhang [238] produced a homotopy for energy-to-fuel optimal station changes in geosynchronous equatorial orbit that considered two-body perturbations. The homotopy leverages a normalization technique and a genetic algorithm for the initial co-states of the indirect method. In 2020, Taheri et al. [240] produced a high-fidelity homotopy for fuel-optimal spacecraft trajectories with variable specific impulse and bang-bang thrust. In 2021, Wu et al. [241] created state-based, energy-optimal control laws for rapid generation of low-thrust Earth-centric trajectories to be used as initial guesses for higher-fidelity analysis.

The primary contributions of this current work are as follows: (1) The first contribution is a simple, increasingly accurate direct-to-indirect mapping for the minimum thrust-acceleration-squared problem using primer vector theory and Lambert’s problem. The mapping is from a coast-impulse model to a continuous-thrust model. Importantly, the mapping is indifferent to the origin of the direct model. (2) The second contribution is a finite sum approximation to minimize thrust-acceleration-squared using the coast-impulse model. (3) The third contribution is the unprecedented optimization of low-thrust transfers with up to 500 revolutions in Cartesian coordinates. The solution process used a multi-shooting method coupled to the mapping, solving problems with various orbital element changes, both large and small, in runtimes of about one hour.

The chapter is organized as follows. After this introduction as the first section, the second section is on the discrete and continuous spacecraft trajectory models used for the direct and indirect problems. The third section outlines two

approximately equivalent, direct and indirect, optimal control problems. After the necessary preliminary discussion on the models and optimal control problems, the details of direct-to-indirect mapping is in the fourth section. In the fifth section, examples are presented and performance is analyzed. The first set of examples demonstrate the increasingly accurate mapping as the number of segments per rev increase. The second set of examples are many-rev, many-segment transfers changing semimajor axis, eccentricity, and inclination. Two of these examples slightly change semimajor axis or eccentricity, and the third significantly changes semimajor axis and eccentricity in a geosynchronous transfer orbit (GTO) to GSO problem.

5.2 Models

The overall goal of this current work is to map an optimal, direct state history $\mathbf{x}_{\mathbb{D}}$ to the optimal, indirect state history $\mathbf{x}_{\mathbb{I}}$ for an optimal control problem minimizing thrust-acceleration-squared and subject to two-body dynamics. Note the superscript ‘*’ means optimal. This mapping with the associated control is compactly

$$\mathbf{x}_{\mathbb{D}}^* \rightarrow \mathbf{x}_{\mathbb{I}}^* \quad (5.1) \qquad \mathbf{\Gamma}_{\mathbb{D}}^* \rightarrow \mathbf{\Gamma}_{\mathbb{I}}^* \quad (5.2)$$

The mapping of Eqs. (5.1) and (5.2) is possible because the direct and indirect solutions are approximately equivalent. In general, $\mathbf{x}_{\mathbb{D}}$ and $\mathbf{x}_{\mathbb{I}}$ are in a Cartesian space and are functions of time, $\mathbf{x}_{\mathbb{D}}(t)$ and $\mathbf{x}_{\mathbb{I}}(t)$, respectively. The states are defined as

$$\mathbf{x}_{\mathbb{D}} = \begin{bmatrix} \mathbf{r}_{\mathbb{D}}^{\top} & \mathbf{v}_{\mathbb{D}}^{\top} \end{bmatrix}^{\top} \quad (5.3) \qquad \mathbf{x}_{\mathbb{I}} = \begin{bmatrix} \mathbf{r}_{\mathbb{I}}^{\top} & \mathbf{v}_{\mathbb{I}}^{\top} \end{bmatrix}^{\top} \quad (5.4)$$

The vectors $\mathbf{r}_{\mathbb{D}}$ and $\mathbf{r}_{\mathbb{I}}$ are positions, and $\mathbf{v}_{\mathbb{D}}$ and $\mathbf{v}_{\mathbb{I}}$ are velocities.

The state $\mathbf{x}_{\mathbb{D}}$ is subject to the dynamics, $\dot{\mathbf{f}}_{\mathbb{D}} = \mathbf{f}_{\mathbb{D}}(t, \mathbf{x}_{\mathbb{D}})$, as

$$\dot{\mathbf{x}}_{\mathbb{D}} = \mathbf{f}_{\mathbb{D}} \iff \begin{bmatrix} \dot{\mathbf{r}}_{\mathbb{D}}^{\top} \\ \dot{\mathbf{v}}_{\mathbb{D}}^{\top} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{\mathbb{D}}^{\top} \\ \mathbf{a}_{\mathbb{D}}^{\top} \end{bmatrix} \quad (5.5)$$

where acceleration $\mathbf{a}_{\mathbb{D}}$ is equal to two-body acceleration, $\mathbf{a}_{\mathbb{D}} = \mathbf{a}_{2\text{BP},\mathbb{D}} = -(\mu/r_{\mathbb{D}}^3)\mathbf{r}_{\mathbb{D}}$. Notably, the control for the direct model, $\mathbf{\Gamma}_{\mathbb{D}}$, is not in the dynamics. The segments are assumed to be Keplerian. Instead, thrust acceleration is discrete. At node i ,

$$\mathbf{\Gamma}_{i,\mathbb{D}} = \frac{\Delta v_{i,\mathbb{D}}}{\Delta t_{\text{avg},i,\mathbb{D}}} \quad (5.6)$$

where $\Delta v_{i,\mathbb{D}}$ is the magnitude of an impulsive maneuver and average flight time is $\Delta t_{\text{avg},i,\mathbb{D}} = (\Delta t_{(i-1)i,\mathbb{D}} + \Delta t_{i(i+1),\mathbb{D}})/2$. The flight times of the segments that neighbor node i are $\Delta t_{(i-1)i,\mathbb{D}}$ and $\Delta t_{i(i+1),\mathbb{D}}$, before and after, respectively.

The state $\mathbf{x}_{\mathbb{I}}$ is subject to the dynamics, $\dot{\mathbf{f}}_{\mathbb{I}} = \mathbf{f}_{\mathbb{I}}(t, \mathbf{x}_{\mathbb{I}})$, as

$$\dot{\mathbf{x}}_{\mathbb{I}} = \mathbf{f}_{\mathbb{I}} \iff \begin{bmatrix} \dot{\mathbf{r}}_{\mathbb{I}}^{\top} \\ \dot{\mathbf{v}}_{\mathbb{I}}^{\top} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{\mathbb{I}}^{\top} \\ \mathbf{a}_{\mathbb{I}}^{\top} \end{bmatrix} \quad (5.7)$$

where acceleration $\mathbf{a}_{\mathbb{I}}$ is equal to two-body acceleration plus control, $\mathbf{a}_{\mathbb{I}} = \mathbf{a}_{2\text{BP},\mathbb{I}} + \mathbf{\Gamma}_{\mathbb{I}} = -(\mu/r_{\mathbb{I}}^3)\mathbf{r}_{\mathbb{I}} + \mathbf{\Gamma}_{\mathbb{I}}$. The control for the indirect model, $\mathbf{\Gamma}_{\mathbb{I}}$, is derived next in the section for the optimal control problems.

5.3 Optimal Control Problems

The overviews for both optimal control problems follow:

Direct Problem Overview

$$\begin{array}{llll}
\text{minimize} & \text{cost function} & J_{\mathbb{D}} = \frac{1}{2} \sum_{i=1}^n \Gamma_{\mathbb{D}}^2 \Delta t_{\text{avg},i} & a \\
\mathbf{x}_d & & & \\
\text{subject to} & \text{dynamics} & \mathbf{a}_{\mathbb{D}} = \mathbf{a}_{2\text{BP},\mathbb{D}} & b \\
& \text{boundary conditions} & \mathbf{x}_{1+,\mathbb{D}} = \mathbf{x}_{0,s}, \quad \mathbf{x}_{n-,\mathbb{D}} = \mathbf{x}_{f,s} & c \\
& \text{path constraints} & \Delta \mathbf{r}_{i,\mathbb{D}} = \mathbf{r}_{i+,\mathbb{D}} - \mathbf{r}_{i-,\mathbb{D}} = \mathbf{0} & d
\end{array}$$

^athrust-acceleration-squared | ^btwo-body | ^cfixed \mathbf{x} -state to fixed \mathbf{x} -state | ^dposition continuity

Indirect Problem Overview

$$\begin{array}{llll}
\text{minimize} & \text{cost function} & J_{\mathbb{I}} = \frac{1}{2} \int_{t_1}^{t_m} \Gamma_{\mathbb{I}}^2 dt & a \\
\mathbf{\Gamma}_{\mathbb{I}} & & & \\
\text{subject to} & \text{dynamics} & \mathbf{a}_{\mathbb{I}} = \mathbf{a}_{2\text{BP},\mathbb{I}} + \mathbf{\Gamma}_{\mathbb{I}} & b \\
& \text{boundary conditions} & \mathbf{x}_{1+,\mathbb{I}} = \mathbf{x}_{0,s}, \quad \mathbf{x}_{m-,\mathbb{I}} = \mathbf{x}_{f,s} & c \\
& \text{path constraints} & \Delta \mathbf{y}_{j,\mathbb{I}} = \mathbf{y}_{j+,\mathbb{I}} - \mathbf{y}_{j-,\mathbb{I}} = \mathbf{0} & d
\end{array}$$

^athrust-acceleration-squared | ^btwo-body plus control | ^cfixed \mathbf{x} -state to fixed \mathbf{x} -state
^d \mathbf{y} -state continuity

The optimal states through time, $\mathbf{x}_{\mathbb{D}}^*$ and $\mathbf{x}_{\mathbb{I}}^*$, are otherwise known as the solutions to these direct and indirect problems. The problems are formed to be approximately equivalent. As mentioned, both problems use two-body dynamics $\mathbf{a}_{2\text{BP}}$ and minimize thrust-acceleration-squared Γ^2 . The transfer is between fixed, terminal, and specified states:

$$\mathbf{x}_{0,s} = \begin{bmatrix} \mathbf{r}_{0,s}^\top \\ \mathbf{v}_{0,s}^\top \end{bmatrix} \quad (5.8) \quad \mathbf{x}_{f,s} = \begin{bmatrix} \mathbf{r}_{f,s}^\top \\ \mathbf{v}_{f,s}^\top \end{bmatrix} \quad (5.9)$$

where the specified position and velocities at the initial time are $\mathbf{r}_{0,s}$ and $\mathbf{v}_{0,s}$, respectively, and likewise $\mathbf{r}_{f,s}$ and $\mathbf{v}_{f,s}$, respectively, for the final time. The total flight time for both problems is from an initial time t_0 to a final time t_f .

5.3.1 Cost

The direct and indirect quadratic cost functions, $J_{\mathbb{D}}$ and $J_{\mathbb{I}}$, respectively, are as follows. The quadratic cost function of the direct problem is the finite sum of a

discrete form of the thrust acceleration $\Gamma_{i,\mathbb{D}}$:

$$J_{\mathbb{D}} = \frac{1}{2} \sum_{i=1}^n \Gamma_{i,\mathbb{D}}^2 \Delta t_{\text{avg},i,\mathbb{D}} = \frac{1}{2} \sum_{i=1}^n \frac{\Delta v_{i,\mathbb{D}}^2}{\Delta t_{\text{avg},i,\mathbb{D}}} \quad (5.10)$$

where the substitution of $\Gamma_{i,\mathbb{D}}$ is from Eq. (5.6). Note for $\Delta t_{\text{avg},1,\mathbb{D}} = \Delta t_{12,\mathbb{D}}/2$ and $\Delta t_{\text{avg},n,\mathbb{D}} = \Delta t_{(n-1)n,\mathbb{D}}/2$ because there is only one neighboring segment for nodes 1 and n , respectively. Also, $t_{1,\mathbb{D}} = t_0$ and $t_{n,\mathbb{D}} = t_f$. The quadratic cost function for the indirect problem is the integral of a continuous form of the thrust acceleration $\Gamma_{\mathbb{I}}$. This cost function is

$$J_{\mathbb{I}} = \frac{1}{2} \int_{t_{1,\mathbb{I}}}^{t_{m,\mathbb{I}}} \Gamma_{\mathbb{I}}^2 dt \quad (5.11)$$

where $t_{1,\mathbb{I}} = t_0$ and $t_{m,\mathbb{I}} = t_f$. Both the direct and indirect representations for thrust acceleration and flight time have the same approximate form at t_i : $\Gamma_{i,\mathbb{D}} = \Delta v_{i,\mathbb{D}}/\Delta t_{\text{avg},i,\mathbb{D}} \approx \Gamma_{i,\mathbb{I}} = [dv/dt]_{i,\mathbb{I}}$ and $\Delta t_{\text{avg},i,\mathbb{D}} \approx [dt]_{i,\mathbb{I}}$. This approximation is valid because finite differences (Δ) are related to differentials (d) of a generic variable x as $\Delta x = dx + O(d^2x)$. It is now apparent the direct and indirect costs, Eqs. (5.10) and (5.11), are approximately equivalent.

5.3.2 Augmented Cost, First Differential, and Two-Point Boundary Value Problem

The direct problem cost $J_{\mathbb{D}}$ of Eq. (5.10) is assumed minimized such that $\mathbf{x}_{\mathbb{D}} = \mathbf{x}_{\mathbb{D}}^*$ and $\mathbf{\Gamma}_{\mathbb{D}} = \mathbf{\Gamma}_{\mathbb{D}}^*$. For the minimization of the indirect problem, the cost of Eq. (5.11) is augmented with Lagrange multipliers or co-states to enforce dynamics and boundary conditions in Eq. (5.12). The first differential of Eq. (5.12) then yields conditions for an optimal solution. These conditions are enforced with a two-point boundary value problem (TPBVP).

Augmented Cost

The augmented cost is

$$\begin{aligned}
\mathcal{G}_{\mathbb{I}} = & \boldsymbol{\nu}_{\mathbf{r}_{1,\mathbb{I}}}^{\top} (\mathbf{r}_{1+,\mathbb{I}} - \mathbf{r}_{0,s}) + \boldsymbol{\nu}_{\mathbf{v}_{1,\mathbb{I}}}^{\top} (\mathbf{v}_{1+,\mathbb{I}} - \mathbf{v}_{0,s}) \\
& + \boldsymbol{\nu}_{\mathbf{r}_{m,\mathbb{I}}}^{\top} (\mathbf{r}_{f,s} - \mathbf{r}_{m-,\mathbb{I}}) + \boldsymbol{\nu}_{\mathbf{v}_{m,\mathbb{I}}}^{\top} (\mathbf{v}_{f,s} - \mathbf{v}_{m-,\mathbb{I}}) \\
& + \int_{t_1}^{t_m} \left[\frac{1}{2} \Gamma_{\mathbb{I}}^2 + \boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{I}}}^{\top} (\mathbf{v}_{\mathbb{I}} - \dot{\mathbf{r}}_{\mathbb{I}}) + \boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}}^{\top} \left(-\frac{\mu}{r_{\mathbb{I}}^3} \mathbf{r}_{\mathbb{I}} + \boldsymbol{\Gamma}_{\mathbb{I}} - \dot{\mathbf{v}}_{\mathbb{I}} \right) \right] dt \quad (5.12)
\end{aligned}$$

where the specified positions and velocities at t_1 and t_m are shown in Eqs. (5.8) and (5.9), respectively, and the variable positions and velocities at t_1 and t_m are $\mathbf{r}_{1+,\mathbb{I}}$ and $\mathbf{v}_{1+,\mathbb{I}}$, and $\mathbf{r}_{m-,\mathbb{I}}$ and $\mathbf{v}_{m-,\mathbb{I}}$, respectively. The Lagrange multipliers $\boldsymbol{\nu}_{\mathbf{r}_{1,\mathbb{I}}}$, $\boldsymbol{\nu}_{\mathbf{v}_{1,\mathbb{I}}}$, $\boldsymbol{\nu}_{\mathbf{r}_{m,\mathbb{I}}}$, and $\boldsymbol{\nu}_{\mathbf{v}_{m,\mathbb{I}}}$ enforce initial and final, position and velocity boundary conditions, respectively. The time-varying Lagrange multipliers $\boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{I}}}$ and $\boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}}$ are known as co-position and co-velocity, respectively, and ensure the time-derivative of the state conforms to the dynamics, Eq. (5.7).

First Differential

For convenience, a \mathcal{G} and \mathcal{H} term are defined with respect to Eq. (5.12):

$$\begin{aligned}
\mathcal{G} = & \boldsymbol{\nu}_{\mathbf{r}_{1,\mathbb{I}}}^{\top} (\mathbf{r}_{1+,\mathbb{I}} - \mathbf{r}_{0,s}) + \boldsymbol{\nu}_{\mathbf{v}_{1,\mathbb{I}}}^{\top} (\mathbf{v}_{1+,\mathbb{I}} - \mathbf{v}_{0,s}) \\
& + \boldsymbol{\nu}_{\mathbf{r}_{m,\mathbb{I}}}^{\top} (\mathbf{r}_{f,s} - \mathbf{r}_{m-,\mathbb{I}}) + \boldsymbol{\nu}_{\mathbf{v}_{m,\mathbb{I}}}^{\top} (\mathbf{v}_{f,s} - \mathbf{v}_{m-,\mathbb{I}}) \quad (5.13)
\end{aligned}$$

$$\mathcal{H} = \frac{1}{2} \Gamma_{\mathbb{I}}^2 + \boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{I}}}^{\top} \mathbf{v}_{\mathbb{I}} + \boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}}^{\top} \left(-\frac{\mu}{r_{\mathbb{I}}^3} \mathbf{r}_{\mathbb{I}} + \boldsymbol{\Gamma}_{\mathbb{I}} \right) \quad (5.14)$$

\mathcal{H} is also known as the Hamiltonian and the Lagrangian is $\mathcal{L} = \frac{1}{2} \Gamma_{\mathbb{I}}^2$. Taking the first differential of Eq. (5.12) and using Eqs. (5.13) and (5.14), the necessary conditions for a minimum produce the extrema of \mathcal{H} with respect to the control, Eq. (5.15); the

Euler-Lagrange equations, Eqs. (5.17) and (5.18); and the initial and final boundary conditions, Eqs. (5.19) to (5.22).

The extrema of \mathcal{H} with respect to the control, thrust acceleration, yields the optimal control law $\mathbf{\Gamma}_{\mathbb{I}}^*$. Note $\mathbf{\Gamma}_{\mathbb{I}}^\top \mathbf{\Gamma}_{\mathbb{I}} = \mathbf{\Gamma}_{\mathbb{I}}^2$. Take

$$\mathbf{0} = \frac{\partial \mathcal{H}^\top}{\partial \mathbf{\Gamma}_{\mathbb{I}}} = \mathbf{\Gamma}_{\mathbb{I}} + \boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}} \quad (5.15)$$

and rearrange to define the optimal control law:

$$\mathbf{\Gamma}_{\mathbb{I}}^* \triangleq -\boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}} \quad (5.16)$$

Equation (5.16) satisfies Pontryagin's minimum principle [242] (ch. 1, sec. 3, pp. 17–21) and Weierstrass (Legendre-Clebsch too) necessary conditions [205] (sec. 10.2–10.3, pp. 167–170). Proof not shown here for brevity.

Substituting Eq. (5.16) into Eq. (5.14), the Euler-Lagrange equations (the dynamics for the co-states) are

$$\dot{\boldsymbol{\lambda}}_{\mathbf{r}_{\mathbb{I}}} = -\frac{\partial \mathcal{H}^\top}{\partial \mathbf{r}_{\mathbb{I}}} = -\mathbf{G}_{\mathbb{I}} \boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}} \quad (5.17) \quad \dot{\boldsymbol{\lambda}}_{\mathbf{v}_{\mathbb{I}}} = -\frac{\partial \mathcal{H}^\top}{\partial \mathbf{v}_{\mathbb{I}}} = -\boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{I}}} \quad (5.18)$$

where the position-derivative of acceleration $\mathbf{G}_{\mathbb{I}}$ is defined as

$$\mathbf{G}_{\mathbb{I}} = \frac{\partial \mathbf{a}_{\mathbb{I}}}{\partial \mathbf{r}_{\mathbb{I}}} = \frac{-\mu}{r_{\mathbb{I}}^5} \left(r_{\mathbb{I}}^2 \mathbf{I}_{3 \times 3} - 3 \mathbf{r}_{\mathbb{I}} \mathbf{r}_{\mathbb{I}}^\top \right)$$

The initial and final boundary conditions, respectively, are

$$\boldsymbol{\lambda}_{\mathbf{r}_{1+, \mathbb{I}}} = \frac{\partial \mathcal{G}}{\partial \mathbf{r}_{1+, \mathbb{I}}}^\top = \boldsymbol{\nu}_{\mathbf{r}_{1, \mathbb{I}}} \quad (5.19) \quad \boldsymbol{\lambda}_{\mathbf{r}_{m-, \mathbb{I}}} = \frac{\partial \mathcal{G}}{\partial \mathbf{r}_{m-, \mathbb{I}}}^\top = -\boldsymbol{\nu}_{\mathbf{r}_{m, \mathbb{I}}} \quad (5.21)$$

$$\boldsymbol{\lambda}_{\mathbf{v}_{1+, \mathbb{I}}} = \frac{\partial \mathcal{G}}{\partial \mathbf{v}_{1+, \mathbb{I}}}^\top = \boldsymbol{\nu}_{\mathbf{v}_{1, \mathbb{I}}} \quad (5.20) \quad \boldsymbol{\lambda}_{\mathbf{v}_{m-, \mathbb{I}}} = \frac{\partial \mathcal{G}}{\partial \mathbf{v}_{m-, \mathbb{I}}}^\top = -\boldsymbol{\nu}_{\mathbf{v}_{m, \mathbb{I}}} \quad (5.22)$$

where the Lagrange multipliers $\boldsymbol{\nu}_{\mathbf{r}_{1,\mathbb{I}}}$, $\boldsymbol{\nu}_{\mathbf{v}_{1,\mathbb{I}}}$, $\boldsymbol{\nu}_{\mathbf{r}_{m,\mathbb{I}}}$, and $\boldsymbol{\nu}_{\mathbf{v}_{m,\mathbb{I}}}$ are arbitrary constants, trivially enforced. Their values are simply the value of the associated co-positions or co-velocities, $\boldsymbol{\lambda}_{\mathbf{r}_{1+,\mathbb{I}}}$, $\boldsymbol{\lambda}_{\mathbf{r}_{1+,\mathbb{I}}}$, $-\boldsymbol{\lambda}_{\mathbf{r}_{m-,\mathbb{I}}}$, and $-\boldsymbol{\lambda}_{\mathbf{v}_{m-,\mathbb{I}}}$, respectively, after propagation to either t_1 or t_m .

Two-Point Boundary Value Problem

A TPBVP is formed to enforce the necessary conditions of the first differential, particularly Eqs. (5.8), (5.9), (5.15), (5.17) and (5.18). Propagation of an augmented state $\mathbf{y}_{\mathbb{I}}$,

$$\mathbf{y}_{\mathbb{I}} = \left[\mathbf{x}_{\mathbb{I}}^{\top}, \boldsymbol{\lambda}_{\mathbf{x}}^{\top} \right]^{\top} = \left[\mathbf{r}_{\mathbb{I}}^{\top}, \mathbf{v}_{\mathbb{I}}^{\top}, \boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{I}}}^{\top}, \boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}}^{\top} \right]^{\top} \quad (5.23)$$

using Eqs. (5.17) and (5.18) (Euler-Lagrange equations) and the optimal control law, Eq. (5.16) from Eq. (5.15), enforces these necessary conditions. To meet the boundary conditions, Eqs. (5.8) and (5.9), a shooting method is devised. Define a decision state $\mathbf{y}_{d_{k,\mathbb{I}}}$, similar to Eq. (5.23), at a time $t_{k,\mathbb{I}}$ for each leg $j(j+1)$ of the shooting method,

$$\mathbf{y}_{d_{k,\mathbb{I}}} = \left[\mathbf{r}_{k,\mathbb{I}}^{\top}, \mathbf{v}_{k,\mathbb{I}}^{\top}, \boldsymbol{\lambda}_{\mathbf{r}_{k,\mathbb{I}}}^{\top}, \boldsymbol{\lambda}_{\mathbf{v}_{k,\mathbb{I}}}^{\top} \right]_{1 \times 12}^{\top} \quad (5.24)$$

The decision state $\mathbf{y}_{d_{k,\mathbb{I}}}$ is integrated backward and forward to $t_{j,\mathbb{I}}$ and $t_{j+1,\mathbb{I}}$, respectively, and updated by an optimizer until the boundary conditions are satisfied. If one leg is used for the TPBVP ($m = 2$), the total decision state is simply $\mathbf{y}_{d_{\mathbb{I}}} = \mathbf{y}_{d_{1,\mathbb{I}}}$. Using more than one leg ($m \geq 3$), the total decision state $\mathbf{y}_{d_{\mathbb{I}}}$ is expanded to $\mathbf{y}_{d_{\mathbb{I}}} = \left[\mathbf{y}_{d_{1,\mathbb{I}}}^{\top}, \dots, \mathbf{y}_{d_{m-1,\mathbb{I}}}^{\top} \right]_{1 \times 12(m-1)}^{\top}$.

The constraint vector $\mathbf{c}_{\mathbb{I}}$ for a one-leg TPBVP is

$$\mathbf{c}_{\mathbb{I}} = \left[\mathbf{r}_{1+, \mathbb{I}}^{\top} - \mathbf{r}_{0, s}^{\top}, \quad \mathbf{v}_{1+, \mathbb{I}}^{\top} - \mathbf{v}_{0, s}^{\top}, \quad \mathbf{r}_{f, s}^{\top} - \mathbf{r}_{m-, \mathbb{I}}^{\top}, \quad \mathbf{v}_{f, s}^{\top} - \mathbf{v}_{m-, \mathbb{I}}^{\top} \right]_{1 \times 12}^{\top} \quad (5.25)$$

and expanded to

$$\mathbf{c}_{\mathbb{I}} = \left[\begin{array}{l} \mathbf{r}_{1+, \mathbb{I}}^{\top} - \mathbf{r}_{0, s}^{\top}, \quad \mathbf{v}_{1+, \mathbb{I}}^{\top} - \mathbf{v}_{0, s}^{\top}, \\ \cdots, \quad \mathbf{r}_{j+, \mathbb{I}}^{\top} - \mathbf{r}_{j-, \mathbb{I}}^{\top}, \quad \mathbf{v}_{j+, \mathbb{I}}^{\top} - \mathbf{v}_{j-, \mathbb{I}}^{\top}, \\ \lambda_{\mathbf{r}_{j+, \mathbb{I}}}^{\top} - \lambda_{\mathbf{r}_{j-, \mathbb{I}}}^{\top}, \quad \lambda_{\mathbf{v}_{j+, \mathbb{I}}}^{\top} - \lambda_{\mathbf{v}_{j-, \mathbb{I}}}^{\top}, \quad \cdots, \\ \mathbf{r}_{f, s}^{\top} - \mathbf{r}_{m-, \mathbb{I}}^{\top}, \quad \mathbf{v}_{f, s}^{\top} - \mathbf{v}_{m-, \mathbb{I}}^{\top} \end{array} \right]_{1 \times 12(m-1)}^{\top} \quad (5.26)$$

for a TPBVP with more than one leg. The internal boundary conditions in Eq. (5.26) are otherwise known as path constraints or internal continuity constraints. Assuming $m \geq 3$, these continuity constraints are for each continuity node j between legs $(j-1)j$ and $j(j+1)$, such that j satisfies $2 \leq j \leq m-1$.

The auxiliary cost for the TPBVP is quadratic, using the constraint vector $\mathbf{c}_{\mathbb{I}}$ as

$$J_{\text{TPBVP}, \mathbb{I}} = \frac{1}{2} \mathbf{c}_{\mathbb{I}}^{\top} \mathbf{c}_{\mathbb{I}} \quad (5.27)$$

At $J_{\text{TPBVP}, \mathbb{I}} = 0$, the boundary and continuity constraints are satisfied, yielding the optimal solution: $\mathbf{y}_{\mathbb{I}}^*$, which uses $\mathbf{\Gamma}_{\mathbb{I}}^*$.

5.4 Direct-to-Indirect Mapping

The direct-to-indirect mapping connects the optimal states and controls of the two problems, Eqs. (5.1) and (5.2). The direct model represents the control $\mathbf{\Gamma}_{i, \mathbb{D}}^*$ discretely and the continuous state $\mathbf{x}_{\mathbb{D}}^*$ at each $t_{i, \mathbb{D}}$, so $\mathbf{x}_{i, \mathbb{D}}^*$. This increasingly accurate

approximate discretization as the number of impulses and segments increase for a solution enables the mapping which generates an initial guess for Eq. (5.24) for each leg $j(j + 1)$. The mapping leverages the shared space between the variational or linearized dynamics of both the direct and indirect models, Eqs. (5.28) to (5.31); the co-state dynamics of the indirect model, Eqs. (5.17) and (5.18); and the optimal control law of the indirect model, Eq. (5.16). Hereafter, the superscript ‘*’ is dropped for clarity because all expressions are assumed optimal. Also, the mapping is done at $t_{i,\mathbb{D}}$, but simplified to t_i because both the direct and indirect models use this same t_i .

For the mapping, the variational dynamics of both models is the linearization of the equations of motion in Eqs. (5.5) and (5.7):

$$\delta\dot{\mathbf{x}}_{\mathbb{D}} = \frac{\partial \mathbf{f}_{\mathbb{D}}}{\partial \mathbf{x}_{\mathbb{D}}} \delta \mathbf{x}_{\mathbb{D}} \quad (5.28) \quad \delta\dot{\mathbf{x}}_{\mathbb{I}} = \frac{\partial \mathbf{f}_{\mathbb{I}}}{\partial \mathbf{x}_{\mathbb{I}}} \delta \mathbf{x}_{\mathbb{I}} \quad (5.29)$$

and in matrix form are

$$\begin{bmatrix} \delta\dot{\mathbf{r}}_{\mathbb{D}} \\ \delta\dot{\mathbf{v}}_{\mathbb{D}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{G}_{\mathbb{D}} & \mathbf{0}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \delta \mathbf{r}_{\mathbb{D}} \\ \delta \mathbf{v}_{\mathbb{D}} \end{bmatrix} \quad (5.30) \quad \begin{bmatrix} \delta\dot{\mathbf{r}}_{\mathbb{I}} \\ \delta\dot{\mathbf{v}}_{\mathbb{I}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{G}_{\mathbb{I}} & \mathbf{0}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \delta \mathbf{r}_{\mathbb{I}} \\ \delta \mathbf{v}_{\mathbb{I}} \end{bmatrix} \quad (5.31)$$

The identical form of the linear dynamics of the direct and indirect models, Eqs. (5.30) and (5.31), means any variational expressions formulated in the indirect model are approximately equal to the counterparts in the direct model, assuming both solutions are optimal with the same boundary conditions: $\delta \mathbf{r}_{\mathbb{D}} \approx \delta \mathbf{r}_{\mathbb{I}}$ and $\delta \mathbf{v}_{\mathbb{D}} \approx \delta \mathbf{v}_{\mathbb{I}}$.

The variational dynamics in Eq. (5.31) are similar to Eqs. (5.17) and (5.18), reproduced as Eq. (5.32) for convenience,

$$\begin{bmatrix} \dot{\boldsymbol{\lambda}}_{\mathbf{r}_{\mathbb{I}}} \\ \dot{\boldsymbol{\lambda}}_{\mathbf{v}_{\mathbb{I}}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & -\mathbf{G}_{\mathbb{I}} \\ -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{I}}} \\ \boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}} \end{bmatrix} \quad (5.32)$$

The following equivalency is stated and its time-derivative:

$$\begin{bmatrix} \boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{I}}} \\ \boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}} \end{bmatrix} = \beta \begin{bmatrix} \mathbf{0}_{3 \times 3} & +\mathbf{I}_{3 \times 3} \\ -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \delta \mathbf{r}_{\mathbb{I}} \\ \delta \mathbf{v}_{\mathbb{I}} \end{bmatrix} \quad (5.33)$$

$$\begin{bmatrix} \dot{\boldsymbol{\lambda}}_{\mathbf{r}_{\mathbb{I}}} \\ \dot{\boldsymbol{\lambda}}_{\mathbf{v}_{\mathbb{I}}} \end{bmatrix} = \beta \begin{bmatrix} \mathbf{0}_{3 \times 3} & +\mathbf{I}_{3 \times 3} \\ -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \delta \dot{\mathbf{r}}_{\mathbb{I}} \\ \delta \dot{\mathbf{v}}_{\mathbb{I}} \end{bmatrix} \quad (5.34)$$

where the coefficient β must have units of TU^{-2} for unit consistency between $\boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}}$ and $\delta \mathbf{r}_{\mathbb{I}}$ or $\boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{I}}}$ and $\delta \mathbf{v}_{\mathbb{I}}$. Plugging in Eqs. (5.33) and (5.34) into Eq. (5.32) [or Eqs. (5.17) and (5.18)] reproduces the variational dynamics of Eq. (5.31) and demonstrates β is arbitrary. Thus, the following mapping expression is true¹:

$$\boldsymbol{\Gamma}_{\mathbb{I}} = -\boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}} = \delta \mathbf{r}_{\mathbb{I}} \quad (5.35) \quad \dot{\boldsymbol{\Gamma}}_{\mathbb{I}} = +\boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{I}}} = \delta \mathbf{v}_{\mathbb{I}} \quad (5.36)$$

where $\beta = 1 \text{ TU}^{-2}$ for simplicity. The magnitude and direction (but not units) of $\boldsymbol{\Gamma}_{\mathbb{I}}$ is exactly equal to a particular neighboring trajectory in the linear variational space of the optimal solution. For arbitrary β , $\boldsymbol{\Gamma}_{\mathbb{I}} \propto \delta \mathbf{r}_{\mathbb{I}}$ and $\dot{\boldsymbol{\Gamma}}_{\mathbb{I}} \propto \delta \mathbf{v}_{\mathbb{I}}$.

To apply the result of Eqs. (5.35) and (5.36), the augmented state $\mathbf{y}_{\mathbb{I}}$ is approximated at every t_i using the optimal direct solution, so the mapping to position $\mathbf{r}_{i,\mathbb{I}}$, velocity $\mathbf{v}_{i,\mathbb{I}}$, co-position $\boldsymbol{\lambda}_{\mathbf{v}_{i,\mathbb{I}}}$, and co-velocity $\boldsymbol{\lambda}_{\mathbf{r}_{i,\mathbb{I}}}$ are in Eqs. (5.37) to (5.40), respectively, and visualized in Fig. 5.2. Note this mapping to $\mathbf{y}_{i,\mathbb{I}}$ is summarized in

¹The relationships in Eqs. (5.35) and (5.36) is an extension of eq. (20) in Ref. 217, where the cost function minimizes flight time or fuel usage and the acceleration is a function of both position and velocity, $\mathbf{a} = \mathbf{a}(\mathbf{r}, \mathbf{v})$. For eq. (20), the position and velocity partial derivatives of \mathbf{a} are needed and defined as $\mathbf{G} = \partial \mathbf{a} / \partial \mathbf{r}$ and $\mathbf{H} = \partial \mathbf{a} / \partial \mathbf{v}$. The matrix \mathbf{G} must be symmetric and the matrix \mathbf{H} must be skew-symmetric, $\mathbf{G} = \mathbf{G}^T$ and $\mathbf{H} = -\mathbf{H}^T$. In this current work, acceleration is a function of position only and both $\mathbf{G}_{\mathbb{D}}$ and $\mathbf{G}_{\mathbb{I}}$ are symmetric.

Algorithm 1.

$$\mathbf{r}_{i,\mathbb{I}} \approx \mathbf{r}_{i,\mathbb{D}} \quad (5.37)$$

$$\mathbf{v}_{i,\mathbb{I}} \approx \frac{1}{2}(\mathbf{v}_{i-,\mathbb{D}} + \mathbf{v}_{i+,\mathbb{D}}) \quad (5.38)$$

$$\boldsymbol{\lambda}_{\mathbf{r},i,\mathbb{I}} \approx \frac{1}{2}(\delta\mathbf{v}_{i-,\mathbb{D}} + \delta\mathbf{v}_{i+,\mathbb{D}}) \quad (5.39)$$

$$\boldsymbol{\lambda}_{\mathbf{v},i,\mathbb{I}} \approx -\boldsymbol{\Gamma}_{i,\mathbb{D}} = -\frac{\Delta\mathbf{v}_{i,\mathbb{D}}}{\Delta t_{\text{avg},i,\mathbb{D}}} = -\delta\mathbf{r}_{i,\mathbb{D}} \quad (5.40)$$

Position is trivially approximated in Eq. (5.37). Velocity is chosen as an average in Eq. (5.38), where $\mathbf{v}_{i-,\mathbb{D}}$ and $\mathbf{v}_{i+,\mathbb{D}}$ are the neighboring velocities before and after node i , respectively. Co-position in Eq. (5.39) uses Eq. (5.36) and is an average too, where the velocity variations are determined from the neighboring path of the optimal solution. Co-velocity uses Eq. (5.35) and is straightforward as Eq. (5.40).

In particular for co-position $\boldsymbol{\lambda}_{\mathbf{r},i,\mathbb{I}}$, the velocity variations of the neighboring solution can be found two ways: (1) a finite difference approach or (2) a partial derivative approach. Both approaches are approximately equivalent. The needed quantities of the direct solution for each $\boldsymbol{\lambda}_{\mathbf{r},i,\mathbb{I}}$ are the flight times $\Delta t_{(i-2)(i-1),\mathbb{D}}$, $\Delta t_{(i-1)i,\mathbb{D}}$, $\Delta t_{i(i+1),\mathbb{D}}$, and $\Delta t_{(i+1)(i+2),\mathbb{D}}$; the positions $\mathbf{r}_{i-1,\mathbb{D}}$, $\mathbf{r}_{i,\mathbb{D}}$, and $\mathbf{r}_{i+1,\mathbb{D}}$; and the velocities $\mathbf{v}_{(i-1)-,\mathbb{D}}$, $\mathbf{v}_{(i-1)+,\mathbb{D}}$, $\mathbf{v}_{i-,\mathbb{D}}$, $\mathbf{v}_{i+,\mathbb{D}}$, $\mathbf{v}_{(i+1)-,\mathbb{D}}$, and $\mathbf{v}_{(i+1)+,\mathbb{D}}$.

For approach (1), the velocity-finite differences of the neighboring path relative to the optimal path are approximately equal to the velocity variation as

$$\tilde{\mathbf{v}}_{i-,\mathbb{D}} - \mathbf{v}_{i-,\mathbb{D}} \approx \delta\mathbf{v}_{i-,\mathbb{D}} \quad (5.41)$$

$$\tilde{\mathbf{v}}_{i+,\mathbb{D}} - \mathbf{v}_{i+,\mathbb{D}} \approx \delta\mathbf{v}_{i+,\mathbb{D}} \quad (5.42)$$

where $\tilde{\mathbf{v}}_{i-,\mathbb{D}}$ and $\tilde{\mathbf{v}}_{i+,\mathbb{D}}$ are the neighboring velocities, and $\delta\mathbf{v}_{i-,\mathbb{D}}$ and $\delta\mathbf{v}_{i+,\mathbb{D}}$ are the

velocity variations, both

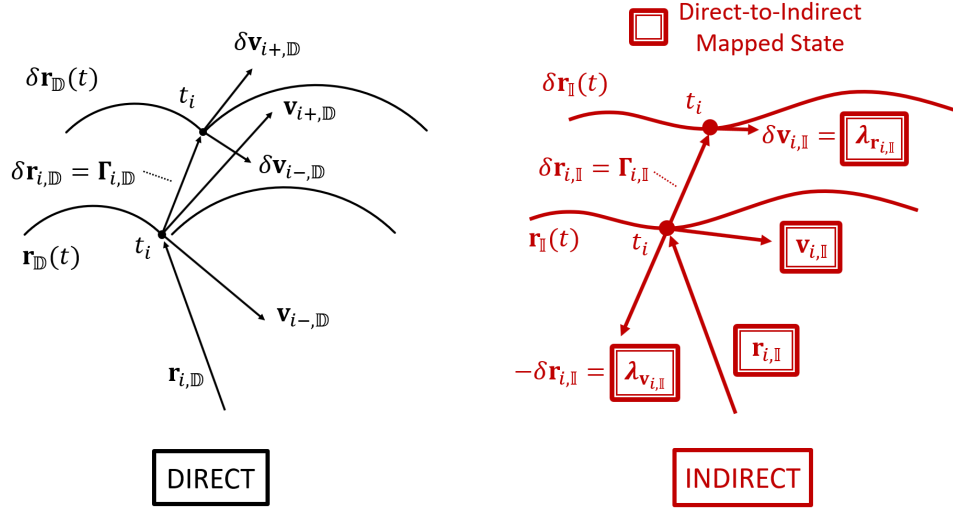


Figure 5.2: Mapping for the state and co-state at node i from the direct (left) to the indirect (right) solution.

before and after node i , respectively. The approximation of the finite-velocity differences to the velocity variations trends better as the finite-velocity differences decrease. To find $\tilde{\mathbf{v}}_{i-,D}$ and $\tilde{\mathbf{v}}_{i+,D}$, the neighboring positions for nodes $i - 1$, i , and $i + 1$ are computed and used as input to two Lambert calls. A Lambert solver is a boundary value solver for the two-body problem. Many Lambert solvers are available, including Battin's [151], Gooding's [152], Izzo's [153], or Russell's [16, 17, 18]. The inputs are terminal positions and flight time. The outputs are the associated terminal velocities. There is a unique solution for the short, zero-revolution case [16].

The neighboring positions are

$$\tilde{\mathbf{r}}_{i-1,\mathbb{D}} = \mathbf{r}_{i-1,\mathbb{D}} + \delta\mathbf{r}_{i-1,\mathbb{D}} \quad (5.43)$$

$$\tilde{\mathbf{r}}_{i,\mathbb{D}} = \mathbf{r}_{i,\mathbb{D}} + \delta\mathbf{r}_{i,\mathbb{D}} \quad (5.44)$$

$$\tilde{\mathbf{r}}_{i+1,\mathbb{D}} = \mathbf{r}_{i+1,\mathbb{D}} + \delta\mathbf{r}_{i+1,\mathbb{D}} \quad (5.45)$$

where

$$\delta\mathbf{r}_{i-1,\mathbb{D}} = \mathbf{\Gamma}_{i-1,\mathbb{D}}/\gamma = (\Delta\mathbf{v}_{i-1}/\Delta t_{\text{avg},i-1})/\gamma \quad (5.46)$$

$$\delta\mathbf{r}_{i,\mathbb{D}} = \mathbf{\Gamma}_{i,\mathbb{D}}/\gamma = (\Delta\mathbf{v}_i/\Delta t_{\text{avg},i})/\gamma \quad (5.47)$$

$$\delta\mathbf{r}_{i+1,\mathbb{D}} = \mathbf{\Gamma}_{i+1,\mathbb{D}}/\gamma = (\Delta\mathbf{v}_{i+1}/\Delta t_{\text{avg},i+1})/\gamma \quad (5.48)$$

If the magnitude of $\mathbf{\Gamma}_{i,\mathbb{D}}$ is large, then a sufficiently large enough $\gamma \geq 1$ can scale down $\mathbf{\Gamma}_{i,\mathbb{D}}$. The velocity-finite difference, Eqs. (5.41) and (5.42), is computed with these smaller position variations, per Eqs. (5.46) to (5.48), then scaled up as $\delta\mathbf{v}_{i-,\mathbb{D}} = (\tilde{\mathbf{v}}_{i-,\mathbb{D}} - \mathbf{v}_{i-,\mathbb{D}})\gamma$ and $\delta\mathbf{v}_{i+,\mathbb{D}} = (\tilde{\mathbf{v}}_{i+,\mathbb{D}} - \mathbf{v}_{i+,\mathbb{D}})\gamma$ to find the velocity variation. This scaling works because linearized dynamics governing variations are proportional. The two Lambert calls to find the terminal velocities are

$$[\tilde{\mathbf{v}}_{(i-1)+,\mathbb{D}}, \tilde{\mathbf{v}}_{i-,\mathbb{D}}] = \text{Lambert}(\Delta t_{(i-1)i,\mathbb{D}}, \tilde{\mathbf{r}}_{i-1,\mathbb{D}}, \tilde{\mathbf{r}}_{i,\mathbb{D}}) \quad (5.49)$$

$$[\tilde{\mathbf{v}}_{i+,\mathbb{D}}, \tilde{\mathbf{v}}_{(i+1)-,\mathbb{D}}] = \text{Lambert}(\Delta t_{i(i+1),\mathbb{D}}, \tilde{\mathbf{r}}_{i,\mathbb{D}}, \tilde{\mathbf{r}}_{i+1,\mathbb{D}}) \quad (5.50)$$

where $\Delta t_{(i-1)i,\mathbb{D}}$ and $\Delta t_{i(i+1),\mathbb{D}}$ are the flight times of the neighboring segments $(i-1)i$ and $i(i+1)$, respectively, to node i . The velocity variations $\delta\mathbf{v}_{i-,\mathbb{D}}$ and $\delta\mathbf{v}_{i+,\mathbb{D}}$ from Eqs. (5.41) and (5.42) can now be used for Eq. (5.39).

For method (2), the expressions for the velocity variations use the partial

derivatives of Lambert’s problem, notated as $\langle \cdot \rangle$:

$$\delta \mathbf{v}_{i-, \mathbb{D}} = \left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_{i-1}} \right\rangle \delta \mathbf{r}_{i-1, \mathbb{D}} + \left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_i} \right\rangle \delta \mathbf{r}_{i, \mathbb{D}} \quad (5.51)$$

$$\delta \mathbf{v}_{i+, \mathbb{D}} = \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_i} \right\rangle \delta \mathbf{r}_{i, \mathbb{D}} + \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_{i+1}} \right\rangle \delta \mathbf{r}_{i+1, \mathbb{D}} \quad (5.52)$$

The Lambert partials $\langle \cdot \rangle$ are known for segments $(i-1)i$ and $i(i+1)$ of the direct optimal solution. References 105, 16, and 18 expand on the derivation of these partials of Lambert’s problem. In brief, for Lambert’s problem, small variations of the independent terminal positions linearly map to small variations of the dependent terminal velocities, per Eqs. (5.51) and (5.52). No scaling of position variations or thrust acceleration vectors is needed. Substitute Eqs. (5.46) to (5.48), where $\gamma = 1$, into Eqs. (5.51) and (5.52) to get the velocity variations $\delta \mathbf{v}_{i-, \mathbb{D}}$ and $\delta \mathbf{v}_{i+, \mathbb{D}}$ for Eq. (5.39).

Both approaches yield the velocity variations using either Eqs. (5.41) and (5.42) or Eqs. (5.51) and (5.52). Approach (1) is convenient because the partial derivatives of Lambert’s problem are not needed. However, method (2) is more reliable. The linear position and velocity variations are exactly dynamically equivalent to the co-states in the indirect model, Eqs. (5.35) and (5.36), and useful to approximate co-states from the direct model, Eqs. (5.39) and (5.40).

5.5 Results

For the computational results (exactly the same as Chapters 3 and 4), the code is written in Fortran 2008 and compiled with Microsoft Visual Studio, Community 2015, Version 14.0.25431.01 with Update 3 and Intel Parallel Studio XE 2017 Composer Edition for Fortran Windows. The code runs on two quad-core Intel Xeon

CPU X5647 2.93 GHz processors and can access 24 GB of memory. Only the two largest optimal orbital transfers using 500 revs use up to approximately 11 GB of memory. Only one processor is used for implementation simplicity. Runtime would improve if parallelized.

The custom-built, gradient-based optimizer (FMINUNC_{UT}) still finds the local solutions in this current research. And as a reminder, the FMINUNC_{UT} employs a BFGS [160, 161, 162, 163] search direction with a line search that uses a golden ratio method with quadratic interpolation at every major iteration. FMINUNC_{UT} takes advantage of fast BLAS [169] matrix and vector operations and parallelizable integration for each leg in a multi-shooting method. The integration uses a Runge-Kutta variable-step method [243] with integration tolerance 10^{-14} . FMINUNC_{UT} uses optimization tolerances related to three inequalities (Gill et al. [209], pp. 306–307): $(J_{\text{prev}} - J)/(1 + |J|) < 10^{-15}$, $\|\mathbf{x}_{d\text{prev}} - \mathbf{x}_d\|/(1 + \|\mathbf{x}_d\|) < \sqrt{10^{-15}}$, and $\|dJ/d\mathbf{x}_d\|/(1 + |J|) \leq \sqrt[3]{10^{-15}}$, where $()_{\text{prev}}$ means the previous iteration and no subscript means current. For more details on FMINUNC_{UT}, see Chapter 3 or Ref. 173.

5.5.1 Results: Relative Error of the Direct Solution to the Indirect Solution

The optimal direct solution is increasingly accurate relative to its equivalent indirect solution as the number of segments and impulses increase. To demonstrate this accuracy, an optimal transfer that significantly changes size and shape in five revolutions is analyzed and defined as $(\mathbf{a}, \mathbf{e}) = (1.0 \text{ DU}, 0.0)$ to $(5.0 \text{ DU}, 0.5)$. Three cases are looked at that increase segment count per rev: 6, 24, and 96 segs/rev. For position, velocity, co-position, and co-velocity, the relative errors ϵ of the direct solution to the indirect solution are defined as

$$\boldsymbol{\epsilon}_{\mathbf{r}} = \frac{\mathbf{r}_{\mathbb{D}} - \mathbf{r}_{\mathbb{I}}}{r_{\mathbb{I}}} \quad (5.53) \quad \boldsymbol{\epsilon}_{\boldsymbol{\lambda}_{\mathbf{r}}} = \frac{\boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{D}}} - \boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{I}}}}{\lambda_{r_{\mathbb{I}}}} \quad (5.55)$$

$$\boldsymbol{\epsilon}_{\mathbf{v}} = \frac{\mathbf{v}_{\mathbb{D}} - \mathbf{v}_{\mathbb{I}}}{v_{\mathbb{I}}} \quad (5.54) \quad \boldsymbol{\epsilon}_{\boldsymbol{\lambda}_{\mathbf{v}}} = \frac{\boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{D}}} - \boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}}}{\lambda_{v_{\mathbb{I}}}} \quad (5.56)$$

In particular for Fig. 5.3, the left column is the time-history of the optimal indirect solution for $\mathbf{r}_{\mathbb{I}}$, $\mathbf{v}_{\mathbb{I}}$, $\boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{I}}}$, and $\boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}}$. For the other three columns, the direct augmented state $\mathbf{y}_{i,\mathbb{I}}$, which includes position $\mathbf{r}_{\mathbb{D}}$, velocity $\mathbf{v}_{\mathbb{D}}$, co-position $\boldsymbol{\lambda}_{\mathbf{r}_{\mathbb{D}}}$, and co-velocity $\boldsymbol{\lambda}_{\mathbf{v}_{\mathbb{I}}}$, is approximated at each t_i for nodes 2 to $n-1$ per Eqs. (5.37) to (5.40) for each of the three cases that use 6, 24, and 96 segs/rev. The $\mathbf{y}_{i,\mathbb{I}}$ at nodes 1 and n are neglected as outliers because the thrust acceleration $\boldsymbol{\Gamma}_{i,\mathbb{D}}$ in Eq. (5.6) is only a function of one neighboring flight time. The relative error is then computed with Eqs. (5.53) to (5.56). The order of accuracy improves by an order of magnitude for each doubling of segment count per rev.

For Fig. 5.4, each $\mathbf{y}_{i,\mathbb{I}}$ per Eqs. (5.37) to (5.40) is integrated backward and forward in time to t_2 and t_{n-1} , respectively. Along the way, each position $\mathbf{r}_{i,\mathbb{I}}$, velocity $\mathbf{v}_{i,\mathbb{I}}$, co-position $\boldsymbol{\lambda}_{\mathbf{r}_{i,\mathbb{I}}}$, and co-velocity $\boldsymbol{\lambda}_{\mathbf{v}_{i,\mathbb{I}}}$ at each t_i is recorded. The left column of Fig. 5.4 shows the continuous, optimal indirect solution in Cartesian space: r_x vs. r_y , v_x vs. v_y , λ_{r_x} vs. λ_{r_y} , and λ_{v_x} vs. λ_{v_y} . The other three columns, left to right, show the integrated approximate augmented states ($\mathbf{r}_{i,\mathbb{I}}$, $\mathbf{v}_{i,\mathbb{I}}$, $\boldsymbol{\lambda}_{\mathbf{r}_{i,\mathbb{I}}}$, and $\boldsymbol{\lambda}_{\mathbf{v}_{i,\mathbb{I}}}$) discretely in Cartesian space at t_i for each of the three cases 6, 24, and 96 segs/rev, respectively. The spread decreases visually as the segment count per rev increases in Fig. 5.4. Moreover, the histograms of Figs. 5.5 to 5.8 plot the relative error for position $\boldsymbol{\epsilon}_{\mathbf{r}_i}$, velocity $\boldsymbol{\epsilon}_{\mathbf{v}_i}$, co-position $\boldsymbol{\epsilon}_{\boldsymbol{\lambda}_{\mathbf{v}_i}}$, and co-velocity $\boldsymbol{\epsilon}_{\boldsymbol{\lambda}_{\mathbf{v}_i}}$ at a particular middle time, depending on the case. The middle times t_i for cases 6, 24, and 96 segs/rev are at nodes $i = 16, 61,$ and $241,$ respectively. Similar to Fig. 5.3, the order of accuracy

improves by an order of magnitude for each doubling of segment count per rev in Figs. 5.5 to 5.8. Interestingly, as a visual metric for accuracy, the patterned order near the origin for the left side of Figs. 5.5 to 5.8 repeats at a different scale for the 24 and 96 segs/rev cases before the chaotic behavior dominates.

The root-mean-square (RMS) of the relative error ϵ vs. segs/rev is presented in Fig. 5.9. Figure 5.9 includes the entire set of augmented states $\mathbf{y}_{i,\mathbb{I}}$ associated with nodes 2 to $n - 1$ and integrated from times t_2 to t_{n-1} . The RMS trends downward from approximately 0.1, 0.01, to less than 0.001 for segment counts per rev of 6, 24, and 96, respectively, further emphasizing the convergence of the discrete direct solution to the continuous indirect solution. Of note, after 70 segs/rev, the curve of Fig. 5.9 is less smooth. The particular technique for the direct solutions used in this research uses many solutions to Lambert problems to represent the coasting

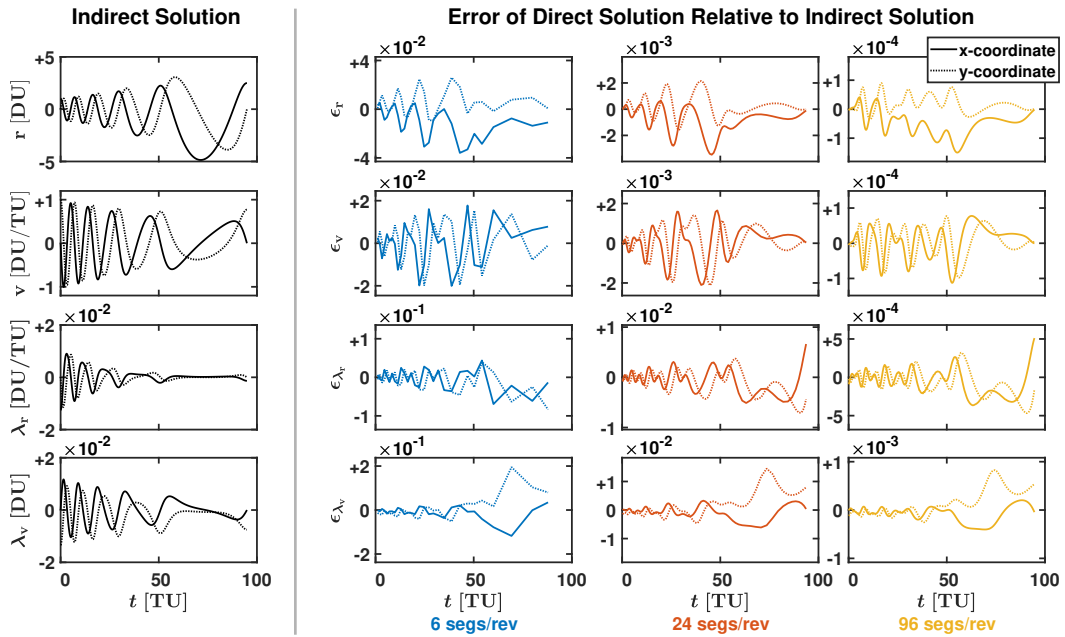


Figure 5.3: Indirect solution (left) and relative error of the direct solution for 6, 24, and 96 segments per rev (mid-left to right, respectively) of an optimal transfer defined by five revs and $(a, e) = (1.0 \text{ DU}, 0.0)$ to $(5.0 \text{ DU}, 0.5)$.

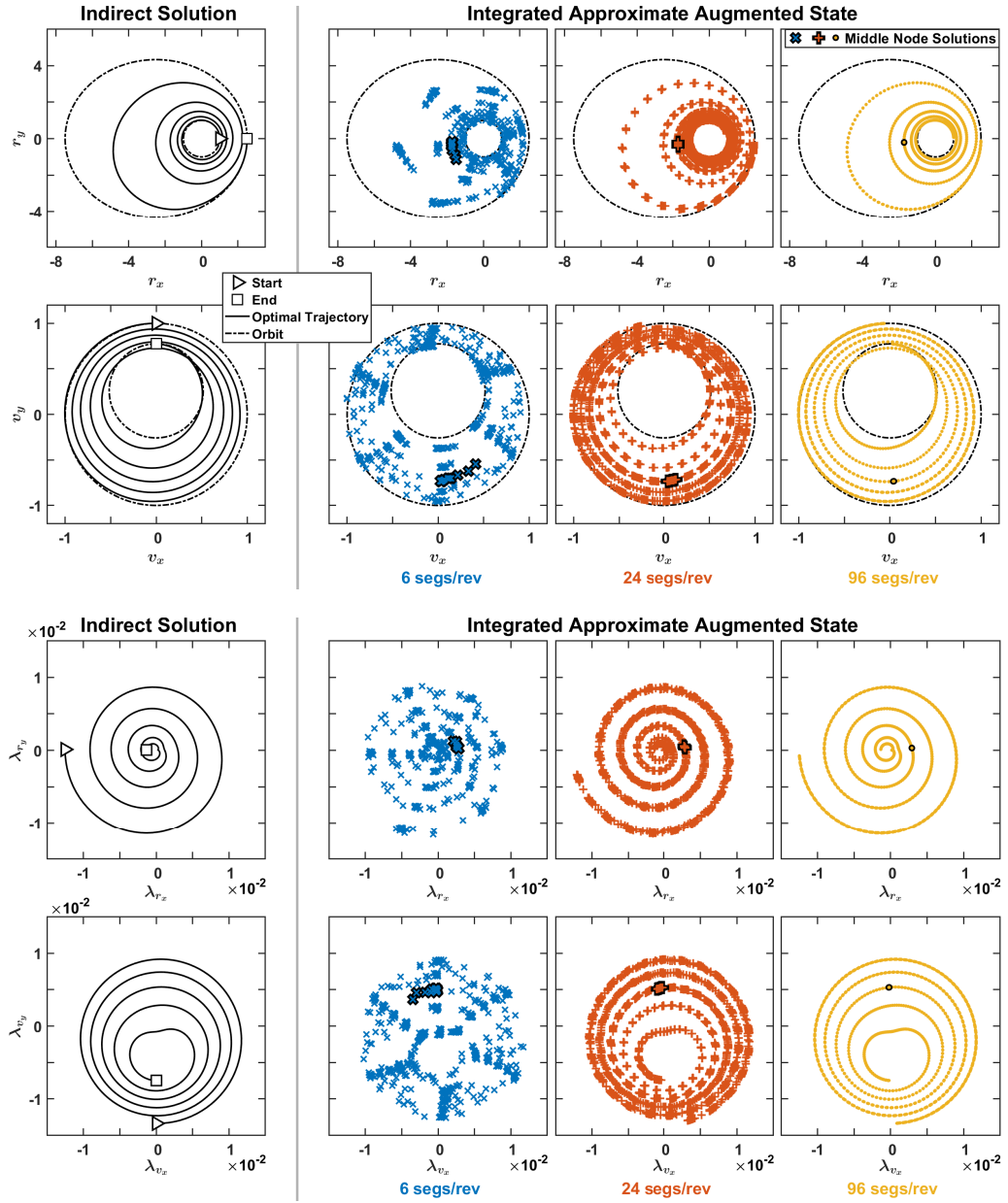


Figure 5.4: Position, velocity, co-position, and co-velocity spaces showing the continuous indirect solution (left) and the integrated approximate augmented state $\mathbf{y}_{i,II}$ (mid-left, mid-right, right). Per Eqs. (5.37) to (5.40), each $\mathbf{y}_{i,II}$ from nodes 2 to $n-1$ is propagated backwards to t_2 and forwards to t_{n-1} .

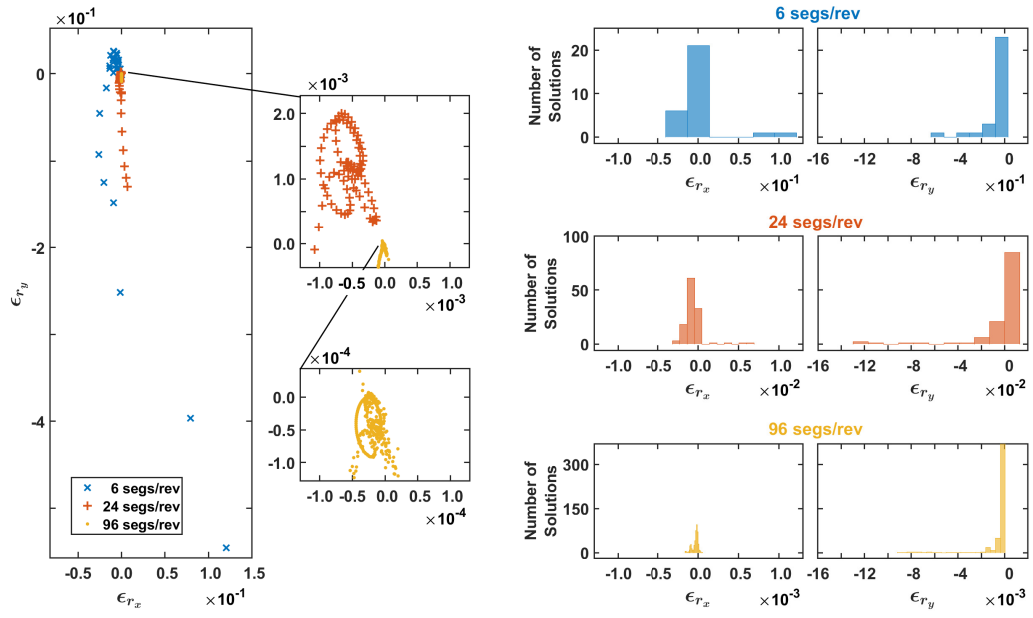


Figure 5.5: Histogram of the r_x vs. r_y solutions at the middle node shown in Fig. 5.4 for the 6, 24, and 96 segs/rev cases.

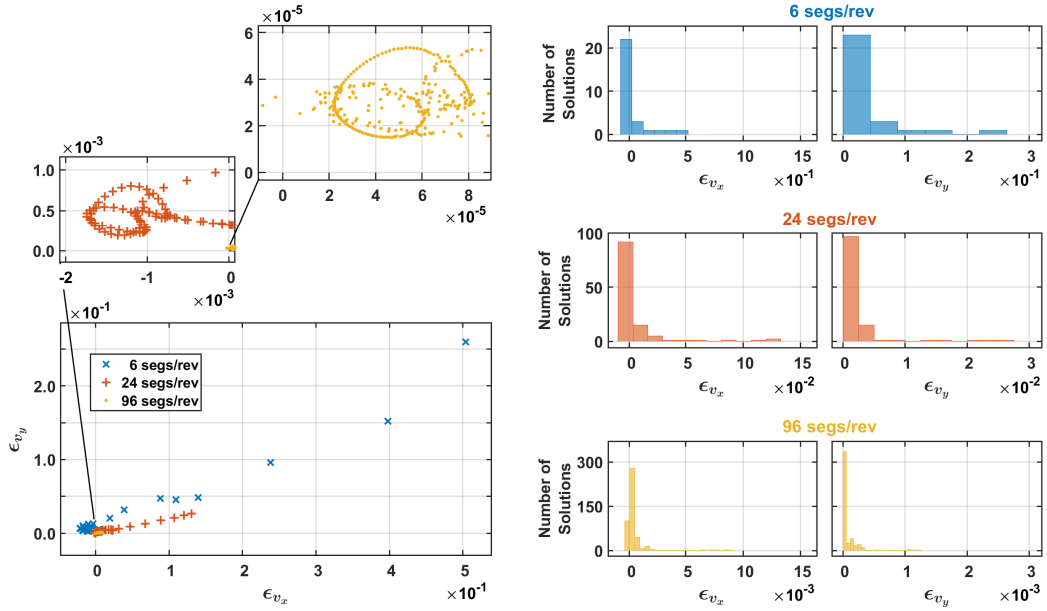


Figure 5.6: Histogram of the v_x vs. v_y solutions at the middle node shown in Fig. 5.4 for the 6, 24, and 96 segs/rev cases.

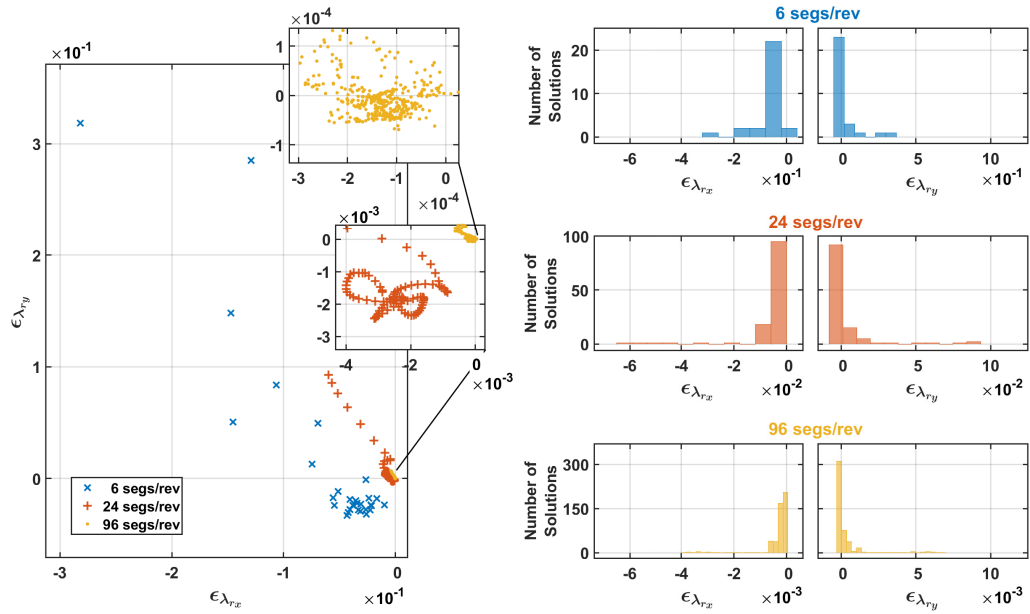


Figure 5.7: Histogram of the λ_{r_x} vs. λ_{r_y} solutions at the middle node shown in Fig. 5.4 for the 6, 24, and 96 segs/rev cases.

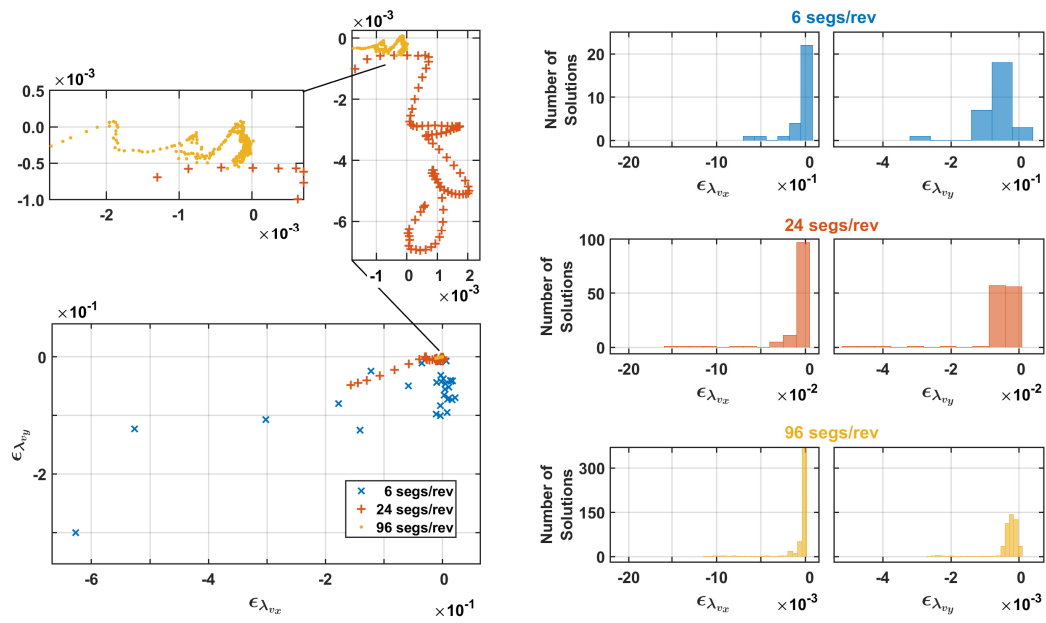


Figure 5.8: Histogram of the λ_{v_x} vs. λ_{v_y} solutions at the middle node shown in Fig. 5.4 for the 6, 24, and 96 segs/rev cases.

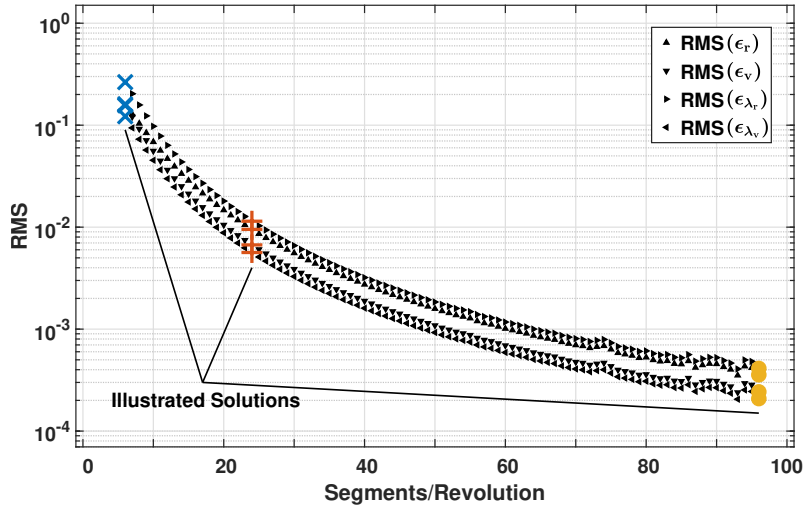


Figure 5.9: RMS of ϵ_r , ϵ_v , ϵ_{λ_r} , and ϵ_{λ_v} , which are computed with direct solution approximations at nodes 2 to $n - 1$ and integrated from t_1 to t_n .

arcs of the coast-impulse model. While fast and robust, this technique encounters a 0° -transfer-angle singularity as terminal positions approach each other close. An initial-value-problem technique using, say solutions to Kepler’s problem, does not encounter this singularity. Only a sufficient number of segments is needed for mapping accuracy, for example the 24 segs/rev case.

5.5.2 Results: Initial Guess Options for Indirect Problem

As shown in Fig. 5.9, the approximate augment state $\mathbf{y}_{i,\mathbb{I}}$ in Eqs. (5.37) to (5.40) is increasingly accurate as the segment count increases for a given solution. The next step is to map $\mathbf{y}_{i,\mathbb{I}}$ to the decision state $\mathbf{y}_{d_{k,\mathbb{I}}}$ per leg of the indirect problem. The most straightforward solution is to simply choose a node i from the direct problem, set $k = i$: $\mathbf{y}_{d_{k,\mathbb{I}}} = \mathbf{y}_{i,\mathbb{I}}$. This initial guess option serves as the baseline for performance to beat versus the other potential improvement options. While this simple strategy seemingly ignores all other segments of the direct solution, the direct solution is

optimal, implicitly affected by every other segment.

The performance analysis uses the same five-rev, one-leg, optimal orbital transfer defined by $(\mathbf{a}, \mathbf{e}) = (1.0 \text{ DU}, 0.0)$ to $(5.0 \text{ DU}, 0.5)$ and the three discretization cases: 6, 24, and 96 segs/revs. The baseline option is called the ‘middle-node solution’ because the chosen node i and its associated time t_i is the middle node for each of the cases, so $i = 16, 61,$ and 241 (in order). For example, the 6-segs/rev case sets $i = 16$ and $k = 1$ for $\mathbf{y}_{d_{k,i}} = \mathbf{y}_{i,\mathbb{I}}$ at $t = t_{16}$. The other options all must determine this 12-state $\mathbf{y}_{d_{k,i}}$ at the same middle times, $t_i = t_{16}, t_i = t_{61},$ or $t_i = t_{241},$ for each of the cases, respectively. For the other options, ‘raw data’ is defined as the data generated and presented for position and velocity at each node i in Fig. 5.4, which is from the integration of all $\mathbf{y}_{i,\mathbb{I}}$ ’s from nodes 2 to $n - 1$ to all t_i ’s from t_2 to t_{n-1} . The data for co-position and co-velocity at each of the node i is similarly generated. Note that in Fig. 5.4, the middle-node solutions for positions and velocity are highlighted. The other options to generate better initial guesses for $\mathbf{y}_{d_{k,i}}$ are as follows. Note the acronyms are defined on the right side of Fig. 5.10.

- MNSO (1–3): Use the middle-node solution only. (baseline option)
- I, RD, Mean and Median (4–9): Compute the mean or median of the raw data at the middle time for each case.
- I, FD, SMAD, Mean and Median (10–15): Compute the mean and median of the raw data at the middle time minus any $\mathbf{y}_{i,\mathbb{I}}$ ’s at this time that are more than three scaled median absolute deviations away.
- I, FD, +/-5N, Mean and Median (16–21): Compute the mean and median of the middle $\mathbf{y}_{i,\mathbb{I}}$ plus the integrated $\mathbf{y}_{i,\mathbb{I}}$ ’s from the five neighboring nodes to the middle time (eleven total), both before and after.
- I, FD, +/-HR, Mean and Median (22–27): Compute the mean and median of

the middle $\mathbf{y}_{i,\mathbb{I}}$ plus the integrated $\mathbf{y}_{i,\mathbb{I}}$'s from the neighboring nodes that make up a half-rev for each case, both before and after. The 6, 24, and 96 segs/rev cases have half-rev node counts of 3, 12, and 48, respectively, so 7, 25, and 97 12-states at the middle time for the three scenarios, respectively.

- LS (28–30): Use the middle-node solution only and improve this guess with a least-squares numerical procedure. The $\mathbf{y}_{d_{k,\mathbb{I}}}$ is integrated and the miss distance at every node i between nodes 2 and $n - 1$ is minimized.

In Fig. 5.10, roughly speaking, all initial guess methods, grouped by counts of segs/rev, trend the same in total iterations for the indirect problem. Despite the variety of possibilities, the straightforward middle-node solution only (options 1, 2, and 3) is recommended. The MNSO is easiest to implement and shows comparable performance to the best options 7–9, which are slightly more complicated to compute.

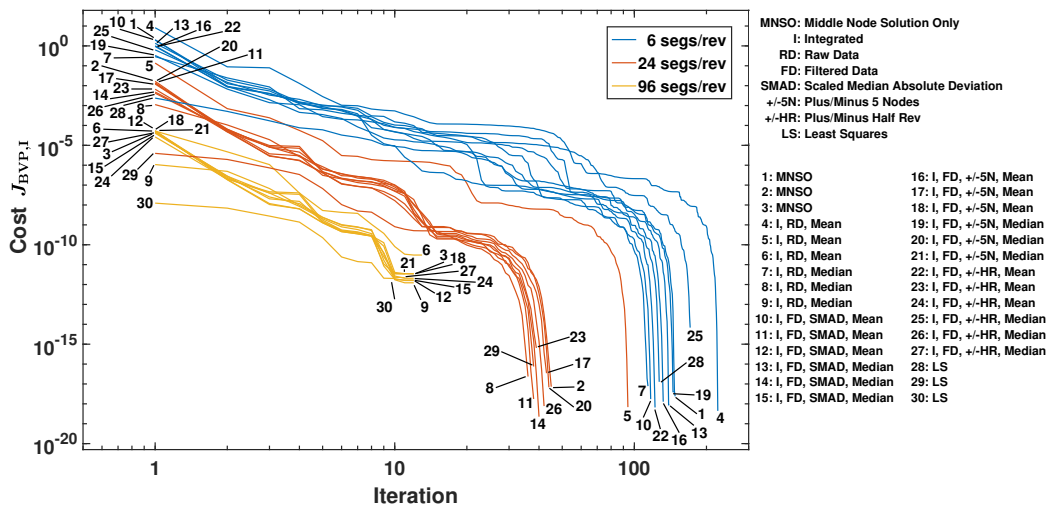


Figure 5.10: Initial guess options for the indirect problem, an optimal thrust-acceleration-squared orbital transfer using five revs and defined by $(a, e) = (1.0 \text{ DU}, 0.0)$ to $(5.0 \text{ DU}, 0.5)$. Three discretization scenarios: 6, 24, and 96 segs/rev are used.

5.5.3 Results: Many-Rev, Multi-Shooting Problems

The next set of examples use a multi-shooting method for the TPBVP, minimizing Eq. (5.27). The first two problems are to find optimal transfers for a small orbital element change: $\Delta \mathbf{a}$ and $\Delta \mathbf{e}$. The third problem is to find an optimal transfer for three large orbital element changes simultaneously: $\Delta \mathbf{a}$, $\Delta \mathbf{e}$, and Δi . More specifically, the transfers of the first two examples are time free and defined by $(\mathbf{a}, \mathbf{e}) = (1.0 \text{ DU}, 0.0)$ to $(2.0 \text{ DU}, 0.0)$ and $(1.5 \text{ DU}, 0.0)$ to $(1.5 \text{ DU}, 0.2)$, respectively. The third example is time free too and defined by $(\mathbf{a}, \mathbf{e}, i) = (3.8305 \text{ DU}, 0.7258, 28.5^\circ)$ to $(6.6107 \text{ DU}, 0.0, 0.0^\circ)$, modeling a transfer from GTO to GSO in canonical units. These problems are time free for convenience, not because of necessity. All three examples vary rev count from 10 to 100 and include a large-rev count solution. The first two examples include a large 500-rev solution, while the largest rev count for the third example is 250. All solutions use the the MNSO initial guess for $\mathbf{y}_{d_{k,\mathbb{I}}}$ of Eq. (5.24) for each leg per five revs. Every optimal direct solution is designed to use 24 segs/rev, deemed sufficiently accurate to balance resolution and problem dimension.

For implementation, each 3-state vector making up the initial guess for $\mathbf{y}_{d_{k,\mathbb{I}}}$ of each leg is scaled to unity to improve convergence: $\mathbf{r}_{k,\mathbb{I},\text{scaled}} = \mathbf{r}_{k,\mathbb{I}}/r_{k,\mathbb{I}}$, $\mathbf{v}_{k,\mathbb{I},\text{scaled}} = \mathbf{v}_{k,\mathbb{I}}/v_{k,\mathbb{I}}$, $\lambda_{\mathbf{r}_{k,\mathbb{I},\text{scaled}}} = \lambda_{\mathbf{r}_{k,\mathbb{I}}}/\lambda_{r_{k,\mathbb{I}}}$, and $\lambda_{\mathbf{v}_{k,\mathbb{I},\text{scaled}}} = \lambda_{\mathbf{v}_{k,\mathbb{I}}}/\lambda_{v_{k,\mathbb{I}}}$. The quadratic TPBVP cost function in Eq. (5.27) is also scaled. Each squared continuity term for position, velocity, co-position, and co-velocity is multiplied by $1/r_{\text{avg}}^2$, $1/v_{\text{avg}}^2$, $1/\lambda_{r_{\text{avg}}}^2$, $1/\lambda_{v_{\text{avg}}}^2$, respectively, where these averages are a function of the initial decision states over all legs: $\mathbf{r}_{k,\mathbb{I}}$, $\mathbf{v}_{k,\mathbb{I}}$, $\lambda_{\mathbf{r}_{k,\mathbb{I}}}$, and $\lambda_{\mathbf{v}_{k,\mathbb{I}}}$. It cannot be overstated; the aforementioned scaling is essential to converge to an optimal solution. A poorly scaled problem can erroneously trigger the tolerances that terminate an optimization algorithm.

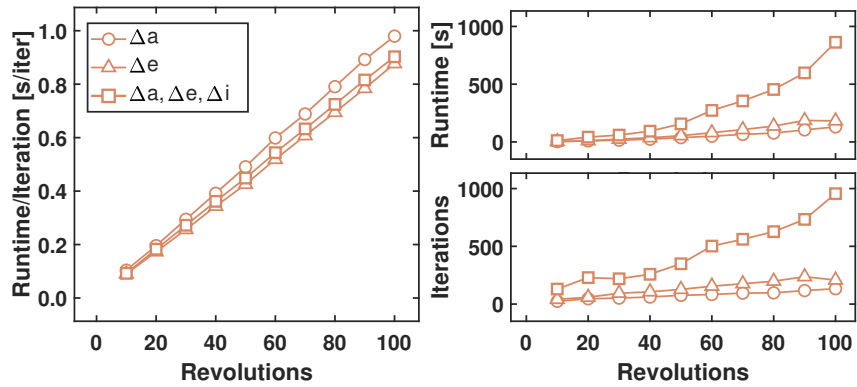


Figure 5.11: Optimization runtimes for the indirect problem for three orbital transfers that use an initial guess derived from a 24 segs/rev direct solution.

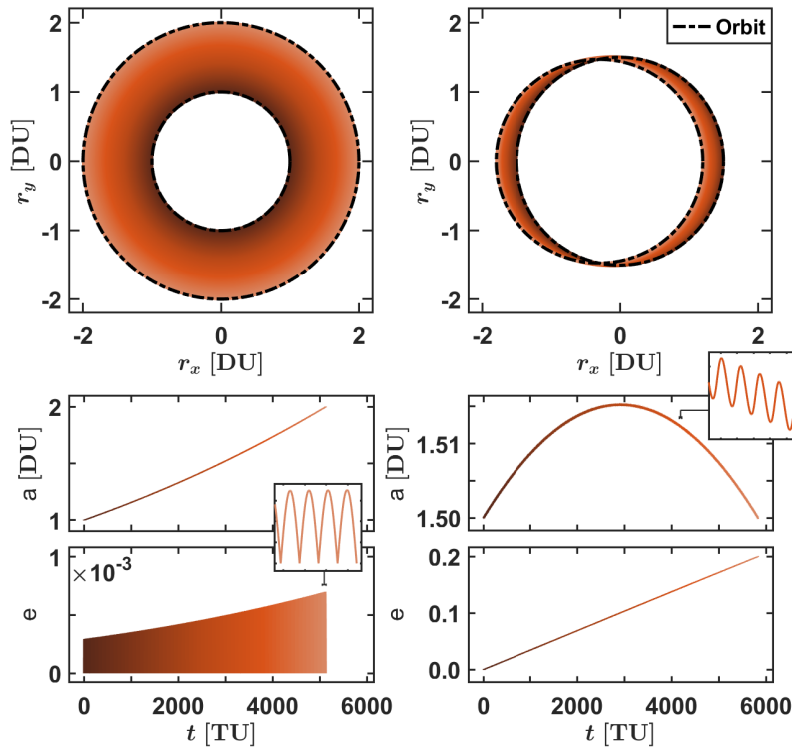


Figure 5.12: Two optimal, time-free, 500-rev transfers defined by $(a, e) = (1.0 \text{ DU}, 0.0)$ to $(2.0 \text{ DU}, 0.0)$ [left] and $(1.5 \text{ DU}, 0.0)$ to $(1.5 \text{ DU}, 0.2)$ [right], respectively.

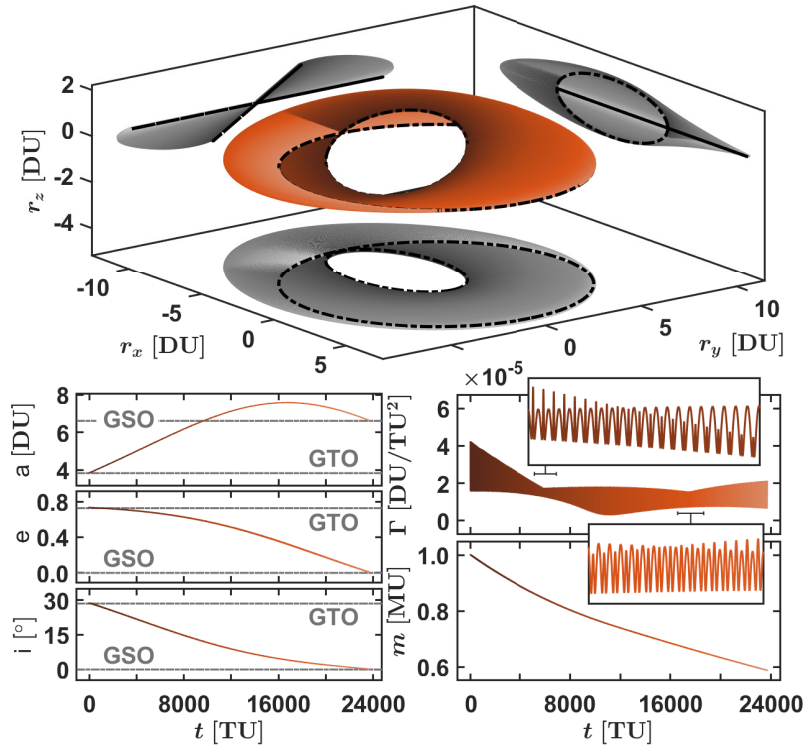


Figure 5.13: Optimal, time-free, 250-rev GTO-to-GSO transfer defined by $(a, e, i) = (3.8305 \text{ DU}, 0.7258, 28.5^\circ)$ to $(6.6107 \text{ DU}, 0.0, 0.0^\circ)$. Note m is for mass.

The optimization runtimes for rev counts from 10 to 100 is shown in Fig. 5.11. Notice the runtimes are seconds to minutes for rev counts of 10 to 100, driven by the number of iterations to optimization. In order, the Δa transfer has the least amount in runtime, then the Δe transfer, to finally the GTO-to-GSO transfer, signaling fast dynamics and larger orbital element changes are numerically more challenging. For the left side of Fig. 5.11, the runtime per iteration of the three examples increases linearly as rev count increases. While these curves are about the same, the Δa transfer curve is above the the GTO-to-GSO transfer curve, which is above the Δe transfer curve. The largest rev-count transfers for Δa -only, Δe -only, and the GTO-to-GSO transfer have runtimes of 30.5 minutes, 1.4 hours, and 3.3 hours, respectively. These runtimes are not shown on Fig. 5.11. See Figs. 5.12 and 5.13 for

a visual on the largest transfers, including magnifications to illuminate sinusoidal behavior. These solutions do not assume a zero-change eccentricity or zero-change semimajor axis for the Δa - or Δe -transfer, respectively. The other curves without magnification have significantly less or no sinusoidal behavior.

5.6 Conclusions

An accurate, physically intuitive, easy-to-implement, direct-to-indirect mapping is derived and demonstrated for optimal spacecraft trajectories minimizing thrust-acceleration-squared in the two-body problem. The direct solution uses a coast-impulse model, where each segment is ballistic and each impulse is the velocity discontinuity between position-continuous segments. Direct coast-impulse solutions are convenient because they are fast, robust, and simple. Furthermore, the mapping is indifferent to the particular technique applied to solve for the direct, coast-impulse solution. The indirect solution uses a continuous-thrust model for a trajectory that satisfies the necessary conditions from a calculus of variations approach. The finite many-thrust-acceleration vectors or impulses of the direct problem converge to the continuous-thrust model. The mapping requires as input from the direct, optimal solution a remarkably small set of positions, velocities, and flight times at only one time. The mapping output is an approximate augmented state that includes the position, velocity, co-position, and co-velocity at the same time as the input. This approximate augmented state is increasingly accurate as the number of segments and impulses increase for a given solution, mitigating hypersensitivity concerns. The mapping derivation leverages primer vector theory to show that the linear space is equivalent to thrust acceleration and the co-states. Notably, Lambert's problem is needed to map the linear position space to the velocity linear space

about the direct, optimal trajectory. The mapping is used in a multi-shooting method to optimize a set of low-thrust transfers using up to 500 revolutions in straightforward Cartesian coordinates. Typically, orbital elements are used for this large of a revolution count. The problems complete various, large and small, orbital element changes in optimization runtimes on the order of an hour. These smooth two-body solutions can serve as the starting state for a homotopy towards a higher-fidelity trajectory with different constraints or objectives.

5.7 Appendix

Algorithm 1 Direct-to-Indirect Mapping for Position, Velocity, Co-Position, and Co-Velocity at Node i .

Input: $\Delta t_{(i-2)(i-1)}$, $\Delta t_{(i-1)i}$, $\Delta t_{i(i+1)}$, $\Delta t_{(i+1)(i+2)}$
 $\mathbf{r}_{i-1,\mathbb{D}}$, $\mathbf{r}_{i,\mathbb{D}}$, $\mathbf{r}_{i+1,\mathbb{D}}$
 $\mathbf{v}_{(i-1)-,\mathbb{D}}$, $\mathbf{v}_{(i-1)+,\mathbb{D}}$, $\mathbf{v}_{i-,\mathbb{D}}$, $\mathbf{v}_{i+,\mathbb{D}}$, $\mathbf{v}_{(i+1)-,\mathbb{D}}$, $\mathbf{v}_{(i+1)+,\mathbb{D}}$
 γ , ‘Approach No.’

Output: $\mathbf{r}_{i,\mathbb{I}}$, $\mathbf{v}_{i,\mathbb{I}}$, $\boldsymbol{\lambda}_{\mathbf{r}_{i,\mathbb{I}}}$, $\boldsymbol{\lambda}_{\mathbf{v}_{i,\mathbb{I}}}$

- 1: $\Delta \mathbf{v}_{i-1} \leftarrow \mathbf{v}_{(i-1)+,\mathbb{D}} - \mathbf{v}_{(i-1)-,\mathbb{D}}$
- 2: $\Delta \mathbf{v}_i \leftarrow \mathbf{v}_{i+,\mathbb{D}} - \mathbf{v}_{i-,\mathbb{D}}$
- 3: $\Delta \mathbf{v}_{i+1} \leftarrow \mathbf{v}_{(i+1)+,\mathbb{D}} - \mathbf{v}_{(i+1)-,\mathbb{D}}$
- 4: $\Delta t_{\text{avg},i-1} \leftarrow (\Delta t_{(i-2)(i-1)} + \Delta t_{(i-1)i})/2$
- 5: $\Delta t_{\text{avg},i} \leftarrow (\Delta t_{(i-1)i} + \Delta t_{i(i+1)})/2$
- 6: $\Delta t_{\text{avg},i+1} \leftarrow (\Delta t_{i(i+1)} + \Delta t_{(i+1)(i+2)})/2$
- 7: $\delta \mathbf{r}_{i-1,\mathbb{D}} \leftarrow \Delta \mathbf{v}_{i-1} / \Delta t_{\text{avg},i-1}$
- 8: $\delta \mathbf{r}_{i,\mathbb{D}} \leftarrow \Delta \mathbf{v}_i / \Delta t_{\text{avg},i}$
- 9: $\delta \mathbf{r}_{i+1,\mathbb{D}} \leftarrow \Delta \mathbf{v}_{i+1} / \Delta t_{\text{avg},i+1}$
- 10: **if** Approach No. = Case (1) **then**
- 11: $\delta \tilde{\mathbf{r}}_{i-1,\mathbb{D}} \leftarrow \delta \mathbf{r}_{i-1,\mathbb{D}} / \gamma$
- 12: $\delta \tilde{\mathbf{r}}_{i,\mathbb{D}} \leftarrow \delta \mathbf{r}_{i,\mathbb{D}} / \gamma$
- 13: $\delta \tilde{\mathbf{r}}_{i+1,\mathbb{D}} \leftarrow \delta \mathbf{r}_{i+1,\mathbb{D}} / \gamma$
- 14: $\tilde{\mathbf{r}}_{i-1,\mathbb{D}} \leftarrow \mathbf{r}_{i-1,\mathbb{D}} + \delta \tilde{\mathbf{r}}_{i-1,\mathbb{D}}$
- 15: $\tilde{\mathbf{r}}_{i,\mathbb{D}} \leftarrow \mathbf{r}_{i,\mathbb{D}} + \delta \tilde{\mathbf{r}}_{i,\mathbb{D}}$
- 16: $\tilde{\mathbf{r}}_{i+1,\mathbb{D}} \leftarrow \mathbf{r}_{i+1,\mathbb{D}} + \delta \tilde{\mathbf{r}}_{i+1,\mathbb{D}}$
- 17: $\tilde{\mathbf{v}}_{(i-1)+,\mathbb{D}}$, $\tilde{\mathbf{v}}_{i-,\mathbb{D}} \leftarrow \text{Lambert}^a(\Delta t_{(i-1)i}, \tilde{\mathbf{r}}_{i-1,\mathbb{D}}, \tilde{\mathbf{r}}_{i,\mathbb{D}})$
- 18: $\tilde{\mathbf{v}}_{i+,\mathbb{D}}$, $\tilde{\mathbf{v}}_{(i+1)-,\mathbb{D}} \leftarrow \text{Lambert}(\Delta t_{i(i+1)}, \tilde{\mathbf{r}}_{i,\mathbb{D}}, \tilde{\mathbf{r}}_{i+1,\mathbb{D}})$
- 19: $\delta \tilde{\mathbf{v}}_{i-,\mathbb{D}} \leftarrow (\tilde{\mathbf{v}}_{i-,\mathbb{D}} - \mathbf{v}_{i-,\mathbb{D}}) \gamma$
- 20: $\delta \tilde{\mathbf{v}}_{i+,\mathbb{D}} \leftarrow (\tilde{\mathbf{v}}_{i+,\mathbb{D}} - \mathbf{v}_{i+,\mathbb{D}}) \gamma$
- 21: **else if** Approach No. = Case (2) **then**
- 22: $\left\langle \frac{\partial \mathbf{v}_{(i-1)\pm}}{\partial \mathbf{r}_{i-1}} \right\rangle$, $\left\langle \frac{\partial \mathbf{v}_{(i-1)\pm}}{\partial \mathbf{r}_i} \right\rangle$, $\left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_{i-1}} \right\rangle$, $\left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_i} \right\rangle \leftarrow \text{LambertPartials}^b(\Delta t_{(i-1)i}, \mathbf{r}_{i-1,\mathbb{D}}, \mathbf{r}_{i,\mathbb{D}})$
- 23: $\left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_i} \right\rangle$, $\left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_{i+1}} \right\rangle$, $\left\langle \frac{\partial \mathbf{v}_{(i+1)-}}{\partial \mathbf{r}_i} \right\rangle$, $\left\langle \frac{\partial \mathbf{v}_{(i+1)-}}{\partial \mathbf{r}_{i+1}} \right\rangle \leftarrow \text{LambertPartials}(\Delta t_{i(i+1)}, \mathbf{r}_{i,\mathbb{D}}, \mathbf{r}_{i+1,\mathbb{D}})$
- 24: $\delta \tilde{\mathbf{v}}_{i-,\mathbb{D}} \leftarrow \left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_{i-1}} \right\rangle \delta \mathbf{r}_{i-1,\mathbb{D}} + \left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_i} \right\rangle \delta \mathbf{r}_{i,\mathbb{D}}$
- 25: $\delta \tilde{\mathbf{v}}_{i+,\mathbb{D}} \leftarrow \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_i} \right\rangle \delta \mathbf{r}_{i,\mathbb{D}} + \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_{i+1}} \right\rangle \delta \mathbf{r}_{i+1,\mathbb{D}}$
- 26: **end if**
- 27: $\mathbf{r}_{i,\mathbb{I}} \leftarrow \mathbf{r}_{i,\mathbb{D}}$ [Eq. (5.37)]
- 28: $\mathbf{v}_{i,\mathbb{I}} \leftarrow (\mathbf{v}_{i-,\mathbb{D}} + \mathbf{v}_{i+,\mathbb{D}})/2$ [Eq. (5.38)]
- 29: $\boldsymbol{\lambda}_{\mathbf{r}_{i,\mathbb{I}}} \leftarrow (\delta \tilde{\mathbf{v}}_{i-,\mathbb{D}} + \delta \tilde{\mathbf{v}}_{i+,\mathbb{D}})/2$ [Eq. (5.39)]
- 30: $\boldsymbol{\lambda}_{\mathbf{v}_{i,\mathbb{I}}} \leftarrow -\Delta \mathbf{v}_i / \Delta t_{\text{avg},i}$ [Eq. (5.40)]

^{a,b} Functions for Lambert’s problem and the partials of Lambert’s problem, respectively.

Chapter 6

Conclusions

Spacecraft trajectory optimization is a diverse field. Better approaches and better software tools enable mission feasibility now and into the future. The aim of this research is to improve the state of the art of solving trajectory problems. The work (i) significantly improves preliminary low-fidelity design with the embedded boundary value problem (EBVP) trajectory optimization technique, (ii) produces a new flight time function for discretizing a trajectory, and (iii) produces a new mapping procedure to solve many-revolution, low-thrust problems in Cartesian coordinates. The mapping transitions from a direct to an indirect method. These contributions are summarized in Section 1.3 and a list of publications is in Appendix C. The present chapter presents the conclusions of this current work, i.e. summarizes and emphasizes importance, and speculates on future direction.

To solve a trajectory problem, often low-fidelity assumptions are made for efficiency, where complexity and fidelity are added as the solution matures. A low-fidelity coast-impulse model, similar to the Sims-Flanagan model, is typically formulated as an initial value problem. But in this current work, the coast-impulse model is formulated as a boundary value problem. This boundary value problem uses many solutions to Lambert's problem to implicitly enforce position and dy-

namical feasibility at every optimization iteration. In Chapters 3 and 4, this EBVP technique is shown to be robust, fast, and better than the similar initial-value technique using Kepler's problem. The EBVP technique handles many different, even poor, initial guesses. The EBVP technique is parallelizable because the Lambert solutions for each segment are independent from each other. The EBVP technique is also unconstrained, reducing complexity and implementation difficulty. A practitioner can quickly build intuition for optimal trajectory design in the two-body problem.

The two-body design space of the EBVP technique is broad. Transfers can minimize fuel, energy, or thrust acceleration without explicit constraints enforced by an optimizer. The transfers are not limited by the size of the orbital element change and they converge to the true, continuous, optimal solution. More specifically, Chapter 3 presents the solution to an assortment of fuel and energy optimal transfers that produce small changes in their orbital elements using equal flight times for each segment. For large changes, Chapter 4 presents a function which varies flight time per segment. This variation on flight time from the piecewise Sundman transformation (PST) function ensures even spatial spacing per segment of a transfer. Moreover, using an exponent of $3/2$ ensures equal steps in eccentric anomaly for transfers that significantly vary semimajor axis and eccentricity. The regularization provided by the PST affects the discretization of total flight time, not the dynamics, enabling simple implementation in trajectory models that are a function of time. The behavior of this alternative, regularizing, flight-time function mimics the error tracking of Keplerian orbit propagation that uses modified regularized dynamics. The specific error tracking is for Keplerian energy, a constant of integration for the two-body problem. The largest optimization problem in Chapter 4 uses 512 revs, 12287 seg-

ments, and the transfer is defined by (a, e, i) : $(1.0 \text{ DU}, 0.0, 30^\circ)$ to $(5.5 \text{ DU}, 0.6, 0^\circ)$. The PST function also enables the transfer through the parabolic boundary of the two-body problem, often difficult for numerical schemes.

Chapter 5 elegantly connects or maps the direct, discrete EBVP technique to its indirect, continuous-thrust equivalent. Both direct and indirect methods can solve for the same solution. This mapping enables the practitioner to use the robustness of a direct method, increasingly accurate as the resolution of the direct trajectory model increases, to circumvent the sensitivity issues of the continuous indirect methods. In this thesis work, the direct problem is to the indirect problem as the Riemann sum is to integration. Importantly too, the mapping is independent of the solution technique for the direct problem, i.e. the EBVP technique is not necessarily needed. The mapping procedure leverages that the thrust acceleration vectors, the linear space, and the co-states share the same space of the optimal solution. Atypically, the co-states are then physically meaningful. The indirect problems minimizing thrust-acceleration-squared have fast optimization runtimes. The largest three problems change (i) semimajor axis only, (ii) eccentricity only, or (iii) semimajor axis, eccentricity, and inclination simultaneously (GTO-to-GSO transfer). In order, the transfers for the indirect problem use 500 revs, 500 revs, and 250 revs, and the runtimes are 30.5 minutes, 1.4 hours, and 3.3 hours, all from their corresponding direct, initial guess that uses 24 segments per revolution.

6.1 Future Work

Overall, the EBVP technique for spacecraft trajectory optimization is reliable and very efficient for problems with dozens of revs and about a thousand segments, and practically feasible with runtimes for solutions with up to hundreds of revs

and up to ten thousand segments. The EBVP technique is supplemental, even complementary, to many other trajectory design approaches. Future work should improve any current limitations. Some suggestions follow.

First, efficiency of the EBVP technique can be improved. In particular, the number of computations for the quasi-Newton BFGS search direction increases quadratically as the size of the problem increases. This quadratic increase from the search direction computation is necessary and is the result of a matrix-vector operation. The quasi-Newton BFGS method has superlinear convergence, enabling overall lower runtimes when compared to other methods using steepest descent or conjugate gradients. To improve the overall optimization runtime, any quadratic behavior must be minimized.

Second, other state and control representations can be considered. The state and control in this thesis uses Cartesian coordinates. While simple and singularity-free, polar or spherical coordinates, as well as classical or modified equinoctial orbital elements are options. Polar or spherical coordinates are convenient for modeling circular motion, while orbital elements are natural for paths that are Keplerian or nearly-Keplerian, especially if the osculating orbital elements are slowly changing versus time. These alternative representations may improve the trajectory model by minimizing fast variations in the state and control, enabling an increase in revolution count for a transfer and minimizing approximate control error.

Third, improvements to the fidelity of the dynamics and control model are possible and summarized twofold. The first approach to improve fidelity is to use a different BVP solver that incorporates other dynamics besides two-body. In other words, the EBVP technique is not exclusive to the two-body problem. For example, this thesis work demonstrates a general shooting method using Runge-Kutta integra-

tion as a performance benchmark which also enables the modeling of non-analytical dynamics. Though, this shooting method is orders-of-magnitude slower than the Lambert solver. Finding sufficiently fast and robust BVP solvers in other dynamical settings should be considered. The second approach to improve fidelity is to use a homotopy. Transitioning from simpler, easier-to-obtain scenarios is a common tactic in literature. For example, there is research into homotopies from minimum-energy to minimum-fuel problems [244, 245, 100, 237, 238], as well as homotopies from simple to more complicated dynamics [246]. Though, for emphasis, these homotopies to medium or high fidelity can be difficult and not straightforward [247]. The state might need to be transformed from Cartesian to orbital elements. The reference frame can be inertial or rotating. Perturbation inclusion to the two-body problem can be from third-body point-masses, spheroidal central bodies, solar radiation pressure, atmospheric drag, and more. Constraints can be complicated too. For example, a constraint can incorporate ephemerides data, which tracks the state history of relevant space objects. A rendezvous problem can be time-dependent, or power and thrust can be dependent on the location of the Sun, where eclipses degrade performance. The radiation environment is also a concern. A trajectory might need to avoid long-duration dosages.

Fourth, the particularly fast and robust Lambert solver used in this work is applied to optimal control problems. A future direction could be to apply this solver to other scenarios such as interplanetary, asteroid [42], or intermoon path planning, space debris removal [248, 249], constellation reconfiguration [250, 251], rendezvous [252, 253], on-orbit servicing [254], orbit determination [255, 256], and more.

Fifth, the PST discretization scheme can be explored without the use of Lambert’s problem. For example, the flight time discretization can be used to vary

the length of an interval in an adaptive collocation method. Furthermore, other regularization schemes beside the Sundman transformation might be of use, such as arc-length or curvature.

Sixth and finally, for the mapping, the connection between the thrust acceleration and co-states can be explored further. The connection relies on the symmetric and skew-symmetric behavior of the linearized two-body dynamics. The linear space around an optimal trajectory is simultaneously the co-state space and thrust-acceleration space. Similar behavior might exist in other dynamical systems, such as the circular restricted three-body problem, or for other cost functions such as flight-time or fuel-minimizing problems.

Appendices

Appendix A

Optimization Algorithms

Three algorithms for the `FMINUNCUT` optimizer are stated here: the broad `FMINUNCUT` algorithm, the algorithm for the BFGS search direction for the line search, and the algorithm for the line search itself. The unconstrained `FMINUNCUT` is presented in Algorithm 2 along with the BFGS search direction in Algorithm 3 and the line search algorithm in Algorithm 4.

The computation of the search direction leverages the efficient BLAS [169] subroutines. There are (s)ymmetric and (p)acked versions of some subroutines where packed takes approximately half the memory as the symmetric version counterpart, enabling the computational feasibility of larger problems but with no parallelization runtime improvements for an increase in thread count as shown in Section 3.4.2. The BLAS subroutines are: `ddot` for the dot product of two vectors; `dsymv` or `dspmv` for the multiplication of a matrix and vector; `dsyr` and `dsyr2` or `dspr` and `dspr2` for two different forms of an outer-product; and `dnrm2` for the 2-norm of a vector. The search direction is always one unit in magnitude to conveniently limit the step size in the line search. The quadratic growth of the computation shown in Section 3.4.1 is from `dspmv`, `dspr`, and `spr2` that require $2n^2$, n^2 , and $2n^2$ floating-point operations, respectively.

Every line search performs three primary computations. First, the minimum is bracketed but with a bounded max step size. Second, the bracketed space is shrunk via a golden ratio method. Third, the minimization is finalized with quadratic interpolation of the three points closest to the minimum. If the denominator in the formula for the quadratic interpolation equals or is close to zero, then the lowest point of the three is chosen as the actual minimum.

Algorithm 2 FMINUNC_{UT} Algorithm

- 1: Iteration Counter: $i = 0$
- 2: Initial Guess Decision State from: \mathbf{x}_{d_0}
- 3: Initial Guess Cost and Partial: J_0 and $\partial J_0/\partial \mathbf{x}_{d_0}$
- 4: **repeat**
- 5: Update Search Direction from **BFGS METHOD**: $\hat{\mathbf{u}}_i$
- 6: Update Iteration Counter: $i = i + 1$
- 7: Save Previous Decision State, Cost, and Partial before their Update:
 $\mathbf{x}_{d_{i-1}}$, J_{i-1} , and $\partial J_{i-1}/\partial \mathbf{x}_{d_{i-1}}$
- 8: Update Decision State with **LINE SEARCH**: \mathbf{x}_{d_i}
- 9: Update Cost and Partial: J_i and $\partial J_i/\partial \mathbf{x}_{d_i}$
- 10: **until** $|\partial J_i/\partial \mathbf{x}_{d_i}| \leq tol$ or $|J_i - J_{i-1}| \leq tol$ or $i > maxval$

Algorithm 3 BFGS Method for Updated Search Direction

Input: Iteration Counter, Current and Previous Decision State,
Current and Previous Partial, and Approximate Hessian Inverse:

i , \mathbf{x}_{d_i} , $\mathbf{x}_{d_{i-1}}$, $\partial J_i/\partial \mathbf{x}_{d_i}$, $\partial J_{i-1}/\partial \mathbf{x}_{d_{i-1}}$, $\tilde{\mathbf{H}}_{i-1}$

Output: Search Direction, and Approximate Hessian Inverse:

$\hat{\mathbf{u}}_i$, $\tilde{\mathbf{H}}_i$

- 1: **if** $i = 0$ **then**
- 2: $\tilde{\mathbf{H}}_i \leftarrow \mathbf{I}$
- 3: **else**
- 4: $\mathbf{s}_i \leftarrow \mathbf{x}_{d_i} - \mathbf{x}_{d_{i-1}}$
- 5: $\mathbf{y}_i \leftarrow \partial J_i/\partial \mathbf{x}_{d_i} - \partial J_{i-1}/\partial \mathbf{x}_{d_{i-1}}$
- 6: $t_{1_i} \leftarrow \mathbf{s}_i^T \mathbf{y}_i$ (ddot)
- 7: $\mathbf{t}_{n_i} \leftarrow \tilde{\mathbf{H}}_{i-1} \mathbf{y}_i$ (dsymv or dspmv)
- 8: $t_{2_i} \leftarrow \mathbf{y}_i^T \mathbf{t}_{n_i}$ (ddot)
- 9: $\tilde{\mathbf{H}}_{i-1} \leftarrow \tilde{\mathbf{H}}_{i-1} + (t_{1_i} + t_{2_i})/t_{1_i}^2 \mathbf{s}_i \mathbf{s}_i^T$ (dsyr or dspr)
- 10: $\tilde{\mathbf{H}}_i \leftarrow \tilde{\mathbf{H}}_{i-1} + (-1/t_{1_i})(\mathbf{t}_{n_i} \mathbf{s}_i^T + \mathbf{s}_i \mathbf{t}_{n_i}^T)$ (dsyr2 or dspr2)
- 11: **end if**
- 12: $\mathbf{t}_{n_i} \leftarrow \tilde{\mathbf{H}}_i \partial J_i/\partial \mathbf{x}_{d_i}$ (dsymv or dspmv)
- 13: $\hat{\mathbf{u}}_i \leftarrow -\mathbf{t}_{n_i}/|\mathbf{t}_{n_i}|$ (dnrm2)

Algorithm 4 LINE SEARCH for Updated Decision State

Input: Current Decision State and Cost:

\mathbf{x}_{d0} , J_0 , 'maxstepsize'

Output: Minimum Decision State of 1D Line Search:

\mathbf{x}_{dmin}

1.0 Bracket Minimum

```
1:  $a_{GR} \leftarrow (3 - \sqrt{5})/2$  ▷ Golden ratio defined
2:  $dx_d \leftarrow 0.01$  ▷ Initial step: start with a small step, but not too small
3:  $\mathbf{x}_{dL} \leftarrow \mathbf{x}_{d0} + dx_d \hat{\mathbf{u}}$ 
4:  $J_L \leftarrow \text{CostFunction}(\mathbf{x}_{dL})$ 
▷ Choose between case one and two. Does the step increase or decrease cost, respectively?
5: if  $J_L > J_0$  then ▷ Case one: step size increases cost
6:    $\mathbf{x}_{dR} \leftarrow \mathbf{x}_{d0}$ 
7:    $J_R \leftarrow J_0$ 
8:   while  $J_L > J_R$  do
9:      $dx_d \leftarrow dx_d a_{GR}$  ▷ Decrease step size
10:    if  $dx_d < \text{small}$  then ▷ Limit minimum step size
11:       $dx_d \leftarrow 0$ 
12:       $\mathbf{x}_{dL} \leftarrow \mathbf{x}_{d0}$ 
13:       $J_L \leftarrow J_0$ 
14:      exit ▷ Optimization has failed
15:    end if
16:     $\mathbf{x}_{dLtemp} \leftarrow \mathbf{x}_{dL}$  ▷ Update decision variables and cost
17:     $J_{Ltemp} \leftarrow J_L$ 
18:     $\mathbf{x}_{dL} \leftarrow \mathbf{x}_{dR} + dx_d \hat{\mathbf{u}}$ 
19:     $J_L \leftarrow \text{CostFunction}(\mathbf{x}_{dL})$ 
20:  end while
21:   $\mathbf{x}_{dL} \leftarrow \mathbf{x}_{dLtemp}$ 
22:   $J_L \leftarrow J_{Ltemp}$ 
23: else if  $J_L < J_0$  then ▷ Case two: step size decreases cost
24:    $\mathbf{x}_{dR} \leftarrow \mathbf{x}_{d0}$ 
25:    $J_R \leftarrow J_0$ 
26:    $dx_{dPrev} \leftarrow [0, dx_d]$ 
27:   while  $J_L < J_R$  do
28:      $dx_d \leftarrow dx_d / a_{GR}^{3/2}$  ▷ Increase step size
29:      $\mathbf{x}_{dLtemp} \leftarrow \mathbf{x}_{dL}$  ▷ Update decision variables and cost
30:      $J_{Ltemp} \leftarrow J_L$ 
31:      $\mathbf{x}_{dL} \leftarrow \mathbf{x}_{dR} + dx_d \hat{\mathbf{u}}$ 
32:      $J_L \leftarrow \text{CostFunction}(\mathbf{x}_{dL})$ 
33:      $\mathbf{x}_{dRtemp} \leftarrow \mathbf{x}_{dR}$ 
34:      $J_{Rtemp} \leftarrow J_R$ 
35:      $\mathbf{x}_{dR} \leftarrow \mathbf{x}_{dLtemp}$ 
36:      $J_R \leftarrow J_{Ltemp}$ 
37:     if  $dx_{dPrev}(1) + dx_d > \text{maxstepsize}$  then ▷ Limit max step size
38:        $dx_d \leftarrow \text{maxstepsize}$ 
39:       exit ▷ Do not minimize. Use max step size.
40:     end if
41:      $dx_{dPrev} \leftarrow [dx_{dPrev}(2), dx_d]$  ▷ Shift the history of previous step sizes
42:   end while
43:    $\mathbf{x}_{dR} \leftarrow \mathbf{x}_{dRtemp}$ 
44:    $J_R \leftarrow J_{Rtemp}$ 
45: else ▷ Minimum was not bracketed
46:   stop ▷ Optimization has failed
47: end if
```

2.0 Golden Ratio Method and 3.0 Quadratic Interpolation ▷ Minimize after minimum is bracketed

48: **if** $dx_d > \text{maxstepsize}$ **then** ▷ Limit the max step size, typically equal to one
49: $dx_d \leftarrow \text{maxstepsize}$
50: $\mathbf{x}_{dmin} \leftarrow \mathbf{x}_{d0} + dx_d \hat{\mathbf{u}}_i$
51: $J_{min} \leftarrow \text{CostFunction}(\mathbf{x}_{dmin})$
52: **else**

2.0 Golden Ratio Method ▷ Minimize

53: $\mathbf{x}_{d1} \leftarrow \mathbf{x}_{dL} + a_{GR}(\mathbf{x}_{dR} - \mathbf{x}_{dL})$
54: $\mathbf{x}_{d2} \leftarrow \mathbf{x}_{dR} - a_{GR}(\mathbf{x}_{dR} - \mathbf{x}_{dL})$
55: $J_1 \leftarrow \text{CostFunction}(\mathbf{x}_{d1})$
56: $J_2 \leftarrow \text{CostFunction}(\mathbf{x}_{d2})$
57: **for** $iter \leftarrow 1$ to 2 **do** ▷ Update the bracket twice
58: **if** $J_1 > J_2$ **then** ▷ Minimum is between \mathbf{x}_{d1} and \mathbf{x}_{dR}
59: $\mathbf{x}_{dL} \leftarrow \mathbf{x}_{d1}$
60: $J_L \leftarrow J_1$
61: $\mathbf{x}_{d1} \leftarrow \mathbf{x}_{d2}$
62: $J_1 \leftarrow J_2$
63: $\mathbf{x}_{d2} \leftarrow \mathbf{x}_{dR} - a_{GR}(\mathbf{x}_{dR} - \mathbf{x}_{dL})$
64: $J_2 \leftarrow \text{CostFunction}(\mathbf{x}_{d2})$
65: **else** ▷ Minimum is between \mathbf{x}_{dL} and \mathbf{x}_{d2}
66: $\mathbf{x}_{dR} \leftarrow \mathbf{x}_{d2}$
67: $J_R \leftarrow J_2$
68: $\mathbf{x}_{d2} \leftarrow \mathbf{x}_{d1}$
69: $J_2 \leftarrow J_1$
70: $\mathbf{x}_{d1} \leftarrow \mathbf{x}_{dL} + a_{GR}(\mathbf{x}_{dR} - \mathbf{x}_{dL})$
71: $J_1 \leftarrow \text{CostFunction}(\mathbf{x}_{d1})$
72: **end if**
73: **end for**

3.0 Quadratic Interpolation ▷ Finalize the minimization

74: **if** $J_1 > J_2$ **then** ▷ Minimum is between \mathbf{x}_{d1} and \mathbf{x}_{dR}
75: **for** $j \leftarrow 1$ to n_{xd} **do**
76: $t_1 \leftarrow J_1 \mathbf{x}_{d2}(j) - J_1 \mathbf{x}_{dR}(j) - J_2 \mathbf{x}_{d1}(j) + J_2 \mathbf{x}_{dR}(j) + J_R \mathbf{x}_{d1}(j) - J_R \mathbf{x}_{d2}(j)$
77: **if** $|t_1| < \text{small}$ **then**
78: $\mathbf{x}_{dmin} \leftarrow \mathbf{x}_{d2}$
79: **exit** ▷ Exit loop
80: **else**
81: $\mathbf{x}_{dmin}(j) \leftarrow (J_1 \mathbf{x}_{d2}(j)^2 - J_1 \mathbf{x}_{dR}(j)^2 - J_2 \mathbf{x}_{d1}(j)^2 + J_2 \mathbf{x}_{dR}(j)^2 + J_R \mathbf{x}_{d1}(j)^2 - J_R \mathbf{x}_{d2}(j)^2) / (2t_1)$
82: **end if**
83: **end for**
84: **else** ▷ Minimum is between \mathbf{x}_{dL} and \mathbf{x}_{d2}
85: **for** $j \leftarrow 1$ to n_{xd} **do**
86: $t_1 \leftarrow J_L \mathbf{x}_{d1}(j) - J_L \mathbf{x}_{d2}(j) - J_1 \mathbf{x}_{dL}(j) + J_1 \mathbf{x}_{d2}(j) + J_2 \mathbf{x}_{dL}(j) - J_2 \mathbf{x}_{d1}(j)$
87: **if** $|t_1| < \text{small}$ **then**
88: $\mathbf{x}_{dmin} \leftarrow \mathbf{x}_{d1}$
89: **exit** ▷ Exit loop
90: **else**
91: $\mathbf{x}_{dmin}(j) \leftarrow (J_L \mathbf{x}_{d1}(j)^2 - J_L \mathbf{x}_{d2}(j)^2 - J_1 \mathbf{x}_{dL}(j)^2 + J_1 \mathbf{x}_{d2}(j)^2 + J_2 \mathbf{x}_{dL}(j)^2 - J_2 \mathbf{x}_{d1}(j)^2) / (2t_1)$
92: **end if**
93: **end for**
94: **end if**

95: **end if**

Appendix B

Sensitivities

The computational sensitivities are essential for a gradient-based, direct optimization method. These partials can be obtained via many methods of varying accuracy, including finite difference methods, complex step methods, automatic differentiation, and more. The variational equations approach in Ref. [257] or Battin’s variational STM implementation [74] on page 467 (implemented in detail in Ref. [105]) are accurate and common in practice. In the current work, the closed-form ivLam for two-body dynamics is faster due to the absence of an integrated solution to an ODE system as shown in examples 1.c and 1.d of Section 3.4.4 and visualized in Fig. 3.3. A variational equation approach to EBVPs is useful to model perturbed two-body dynamics. A faster, but less accurate, alternative is to approximate two-body perturbations as impulsive $\Delta\mathbf{v}$ maneuvers[157].

B.1 Cost Partial and Differentials

The following derivation of the cost partials for the direct problem is based off differentials $d(\cdot)$ to organize dependencies. There are two sets of cost partials: (i) constant flight time per segment partials used in Chapter 3 and (ii) varying flight

time per segment partials used in Chapters 4 and 5.

As mentioned previously, the cost J in Eq. (3.1) is a function of the impulsive maneuvers at each node i , so $J = J(\Delta v_1(\Delta \mathbf{v}_1), \dots, \Delta v_n(\Delta \mathbf{v}_n))$. Using the chain-rule, the expansion of J as a cost differential from nodes 1 to n is

$$dJ = \frac{\partial J}{\partial \Delta v_1} \frac{\partial \Delta v_1}{\partial \Delta \mathbf{v}_1} d\Delta \mathbf{v}_1 + \dots + \frac{\partial J}{\partial \Delta v_n} \frac{\partial \Delta v_n}{\partial \Delta \mathbf{v}_n} d\Delta \mathbf{v}_n \quad (\text{B.1})$$

where the cost partial of node i in terms of the impulsive maneuver, $\Delta \mathbf{v}_i$, is $\frac{\partial J}{\partial \Delta v_i} = 1$ and $\frac{\partial \Delta v_i}{\partial \Delta \mathbf{v}_i} = \frac{\Delta \mathbf{v}_i^T}{\Delta v_i}$ for minimum fuel [$k = 1$ in Eq. (3.1)] or $\frac{\partial J}{\partial \Delta v_i} = \Delta v_i$ and $\frac{\partial \Delta v_i}{\partial \Delta \mathbf{v}_i} = \Delta \mathbf{v}_i^T$ for minimum energy [$k = 0$ in Eq. (3.1)] or a hybrid combination for $0 < k < 1$ in Eq. (3.1). The cost differential, Eq. (B.1), currently terminates at the impulsive maneuver differential $d\Delta \mathbf{v}_i$ for each node i , but as previously mentioned, the impulsive maneuver is a function of the velocity before \mathbf{v}_{i-} and after \mathbf{v}_{i+} node i written as $\Delta \mathbf{v}_i = \mathbf{v}_{i+} - \mathbf{v}_{i-}$ and these velocities are a function of position and time written as $\mathbf{v}_{i-} = \mathbf{v}_{i-}(\mathbf{r}_{i-1}, \mathbf{r}_i, t_{(i-1)+}, t_{i-})$ and $\mathbf{v}_{i+} = \mathbf{v}_{i+}(\mathbf{r}_i, \mathbf{r}_{i+1}, t_{i+}, t_{(i+1)-})$. Thus, the differential of the impulsive maneuver is $d\Delta \mathbf{v}_i = d\mathbf{v}_{i+} - d\mathbf{v}_{i-}$ and the differential of the velocity before $d\mathbf{v}_{i-}$ and after $d\mathbf{v}_{i+}$ node i is

$$d\mathbf{v}_{i-} = \left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_{i-1}} \right\rangle d\mathbf{r}_{i-1} + \left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_i} \right\rangle d\mathbf{r}_i + \left\langle \frac{\partial \mathbf{v}_{i-}}{\partial t_{(i-1)+}} \right\rangle dt_{(i-1)+} + \left\langle \frac{\partial \mathbf{v}_{i-}}{\partial t_{i-}} \right\rangle dt_{i-} \quad (\text{B.2})$$

$$d\mathbf{v}_{i+} = \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_i} \right\rangle d\mathbf{r}_i + \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_{i+1}} \right\rangle d\mathbf{r}_{i+1} + \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial t_{i+}} \right\rangle dt_{i+} + \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial t_{(i+1)-}} \right\rangle dt_{(i+1)-} \quad (\text{B.3})$$

where the $\langle \cdot \rangle$ represents the EBVP partials that assume position continuity is enforced by design by using EBVPs, not the optimizer.

Before substitution of Eqs. (B.2) and (B.3) into Eq. (B.1), careful consid-

eration must be made towards the first node 1 and the last node n . The initial and final nodes are constrained by design, not the optimizer, because they are prescribed by a user-defined path, in this current work this path is a two-body orbit. Thus, the differentials for position at the initial and final orbits are $d\mathbf{r}_1 = \mathbf{v}_{1-}d\tilde{t}_{1-}$ and $d\mathbf{r}_n = \mathbf{v}_{n+}d\tilde{t}_{n+}$ and the differentials for velocity are $d\mathbf{v}_{1-} = \mathbf{a}_{1-}d\tilde{t}_{1-}$ and $d\mathbf{v}_{n+} = \mathbf{a}_{n+}d\tilde{t}_{n+}$. The time-like \tilde{t} to parameterize an orbit is not associated with the time of flight of the transfer trajectory. Furthermore, for every segment $(i+1)i$ from node i to $i+1$, the time of flight is defined to be $\Delta t = t_{(i+1)-} - t_{i+}$ and the time of flight differential is $d\Delta t = dt_{(i+1)-} - dt_{i+}$. For convenience, the initial time t_{i+} of segment $(i+1)i$ is designed to be zero, so the time of flight Δt is the final time $t_{(i+1)-}$ of segment $(i+1)i$, i.e. $\Delta t = t_{(i+1)-}$. The initial time t_{i+} can conveniently be zero because time does not explicitly show up in the dynamics. In other words, for time-free unconstrained spacecraft trajectory optimization problems, only the time of flight Δt of each segment $(i+1)i$ is needed, not the initial and final times of each segment. Thus, when computing the velocity differentials at node i shown in Eqs. (B.2) and (B.3), the time of flight differential is $d\Delta t = dt_{i-} = dt_{(i+1)-}$ because $dt_{(i-1)+} = dt_{i+} = 0$.

Now, substitute Eqs. (B.2) and (B.3) into Eq. (B.1), the aforementioned differentials for time and position, and group similar differential terms to get a cost differential of the form

$$dJ = \frac{\partial J}{\partial \tilde{t}_{1-}}d\tilde{t}_{1-} + \frac{\partial J}{\partial \tilde{t}_{n+}}d\tilde{t}_{n+} + \frac{\partial J}{\partial \Delta t}d\Delta t + \frac{\partial J}{\partial \mathbf{r}_2}d\mathbf{r}_2 + \cdots + \frac{\partial J}{\partial \mathbf{r}_{n-1}}d\mathbf{r}_{n-1} \quad (\text{B.4})$$

where the cost partials are the coefficients of each differential of the initial time-like variable on the initial orbit \tilde{t}_{1-} , the final time-like variable on the final orbit \tilde{t}_{n+} , the time of flight Δt of each segment $(i+1)i$, and all intermediate positions \mathbf{r}_i from

nodes 2 to $n - 1$. The cost partials for Chapter 3 are

$$\frac{\partial J}{\partial \tilde{t}_{1-}} = \frac{\partial J}{\partial \Delta v_1} \frac{\partial \Delta v_1}{\partial \Delta \mathbf{v}_1} \left(\left\langle \frac{\partial \mathbf{v}_{1+}}{\partial \mathbf{r}_1} \right\rangle \mathbf{v}_{1-} - \mathbf{a}_{1-} \right) + \frac{\partial J}{\partial \Delta v_2} \frac{\partial \Delta v_2}{\partial \Delta \mathbf{v}_2} \left(- \left\langle \frac{\partial v_{2-}}{\partial r_1} \right\rangle \mathbf{v}_{1-} \right) \quad (\text{B.5})$$

$$\frac{\partial J}{\partial \tilde{t}_{n+}} = \frac{\partial J}{\partial \Delta v_{n-1}} \frac{\partial \Delta v_{n-1}}{\partial \Delta \mathbf{v}_{n-1}} \left(\left\langle \frac{\partial \mathbf{v}_{(n-1)+}}{\partial \mathbf{r}_n} \right\rangle \mathbf{v}_{n+} \right) + \frac{\partial J}{\partial \Delta v_n} \frac{\partial \Delta v_n}{\partial \Delta \mathbf{v}_n} \left(\mathbf{a}_{n+} - \left\langle \frac{\partial \mathbf{v}_{n-}}{\partial \mathbf{r}_n} \right\rangle \mathbf{v}_{n+} \right) \quad (\text{B.6})$$

$$\begin{aligned} \frac{\partial J}{\partial \Delta t} &= \frac{\partial J}{\partial \Delta v_1} \frac{\partial \Delta v_1}{\partial \Delta \mathbf{v}_1} \left\langle \frac{\partial \mathbf{v}_{1+}}{\partial \Delta t} \right\rangle + \sum_{i=2}^{n-1} \frac{\partial J}{\partial \Delta v_i} \frac{\partial \Delta v_i}{\partial \Delta \mathbf{v}_i} \left(- \left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \Delta t} \right\rangle + \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \Delta t} \right\rangle \right) \\ &\quad + \frac{\partial J}{\partial \Delta v_n} \frac{\partial \Delta v_n}{\partial \Delta \mathbf{v}_n} \left(- \left\langle \frac{\partial \mathbf{v}_{n-}}{\partial \Delta t} \right\rangle \right) \end{aligned} \quad (\text{B.7})$$

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{r}_i} &= \frac{\partial J}{\partial \Delta v_{i-1}} \frac{\partial \Delta v_{i-1}}{\partial \Delta \mathbf{v}_{i-1}} \left\langle \frac{\partial \mathbf{v}_{(i-1)+}}{\partial \mathbf{r}_i} \right\rangle + \frac{\partial J}{\partial \Delta v_i} \frac{\partial \Delta v_i}{\partial \Delta \mathbf{v}_i} \left(\left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_i} \right\rangle - \left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_i} \right\rangle \right) \\ &\quad + \frac{\partial J}{\partial \Delta v_{i+1}} \frac{\partial \Delta v_{i+1}}{\partial \Delta \mathbf{v}_{i+1}} \left(- \left\langle \frac{\partial \mathbf{v}_{(i+1)-}}{\partial \mathbf{r}_i} \right\rangle \right) \end{aligned} \quad (\text{B.8})$$

Note the following useful identities between velocity partials that relates flight time Δt and time t before and after node i : $\left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \Delta t} \right\rangle = \left\langle \frac{\partial \mathbf{v}_{i-}}{\partial t_{i-}} \right\rangle$ and $\left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \Delta t} \right\rangle = \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial t_{(i+1)-}} \right\rangle$. Also note, if the problem is fixed-state to fixed-state, not orbit-to-orbit, then the initial and final time-like variables that parameterize the initial and final orbits do not vary, i.e. $d\tilde{t}_{1-} = 0$ and $d\tilde{t}_{n+} = 0$ and the associated cost partials are zero, i.e. $\frac{\partial J}{\partial \tilde{t}_{1-}} = 0$ and $\frac{\partial J}{\partial \tilde{t}_{n+}} = 0$. Likewise, the time of flight cost partial is zero, $\frac{\partial J}{\partial \Delta t} = 0$, if the total time of flight is fixed.

The cost partials reported in Eqs. (B.5) to (B.8) for Chapter 3 are modified to a more general form to accommodate the varying flight time function, the PST shown in Eq. (4.1). Otherwise, position continuity is still assumed and a cost of the form in Eq. (3.1) is still used. The more general form of the cost partials assumes Eq. (3.1) is a function of $\Delta \tau$ for each segment, instead of Δt , and a function of the intermediate positions, \mathbf{r}_2 to \mathbf{r}_{n-1} , for nodes 2 to $n - 1$, respectively. The examples in Chapters 4 and 5 are fixed six-state to six-state, corresponding to Keplerian orbits

for simplicity. Thus modifications to Eqs. (B.5) and (B.6) are ignored because those cost partials are zero. The cost partials for Chapter 4 with respect to $\Delta\tau$ and \mathbf{r}_i , modifying Eqs. (B.7) and (B.8), respectively, are

$$\begin{aligned} \frac{dJ}{d\Delta\tau} &= \frac{\partial J}{\partial\Delta v_1} \frac{\partial\Delta v_1}{\partial\Delta\mathbf{v}_1} \frac{\partial\mathbf{v}_{1+}}{\partial\Delta\tau} + \sum_{i=2}^{n-1} \frac{\partial J}{\partial\Delta v_i} \frac{\partial\Delta v_i}{\partial\Delta\mathbf{v}_i} \left(\frac{\partial\mathbf{v}_{i+}}{\partial\Delta\tau} - \frac{\partial\mathbf{v}_{i-}}{\partial\Delta\tau} \right) \\ &\quad + \frac{\partial J}{\partial\Delta v_n} \frac{\partial\Delta v_n}{\partial\Delta\mathbf{v}_n} \left(-\frac{\partial\mathbf{v}_{n-}}{\partial\Delta\tau} \right) \end{aligned} \quad (\text{B.9})$$

$$\begin{aligned} \frac{dJ}{d\mathbf{r}_i} &= \frac{\partial J}{\partial\Delta v_{i-1}} \frac{\partial\Delta v_{i-1}}{\partial\Delta\mathbf{v}_{i-1}} \frac{\partial\mathbf{v}_{(i-1)+}}{\partial\mathbf{r}_i} + \frac{\partial J}{\partial\Delta v_i} \frac{\partial\Delta v_i}{\partial\Delta\mathbf{v}_i} \left(\frac{\partial\mathbf{v}_{i+}}{\partial\mathbf{r}_i} - \frac{\partial\mathbf{v}_{i-}}{\partial\mathbf{r}_i} \right) \\ &\quad + \frac{\partial J}{\partial\Delta v_{i+1}} \frac{\partial\Delta v_{i+1}}{\partial\Delta\mathbf{v}_{i+1}} \left(-\frac{\partial\mathbf{v}_{(i+1)-}}{\partial\mathbf{r}_i} \right) \end{aligned} \quad (\text{B.10})$$

derived by also considering the variational effects of node i on its neighboring nodes $i-1$ and $i+1$. The initial τ_{i+} is always zero for convenience, so $\Delta\tau_{i(i+1)} = \tau_{(i+1)-} - \tau_{i+} = \tau_{(i+1)-}$. Moreover, $\Delta\tau = \Delta\tau_{i(i+1)}$ because $\Delta\tau$ is the same for every segment for a particular solution.

These cost partials in Eqs. (B.9) and (B.10) are built from velocity partials that combine the partials of Lambert's problem (the EBVP partials) and the PST. For segment $i(i+1)$, the initial velocity \mathbf{v}_{i+} and final velocity $\mathbf{v}_{(i+1)-}$ are functions of the initial position \mathbf{r}_i and final position \mathbf{r}_{i+1} , and $\Delta t_{i(i+1)}$, which in turn is also a function of \mathbf{r}_i , \mathbf{r}_{i+1} , and $\Delta\tau$. These dependencies are written compactly as

$$\begin{aligned} \mathbf{v}_{i+} &= \mathbf{v}_{i+}(\mathbf{r}_i, \mathbf{r}_{i+1}, \Delta t_{i(i+1)}(\mathbf{r}_i, \mathbf{r}_{i+1}, \Delta\tau)) \\ \mathbf{v}_{(i+1)-} &= \mathbf{v}_{(i+1)-}(\mathbf{r}_i, \mathbf{r}_{i+1}, \Delta t_{i(i+1)}(\mathbf{r}_i, \mathbf{r}_{i+1}, \Delta\tau)) \end{aligned}$$

Applying the chain rule on the aforementioned \mathbf{v}_{i+} and $\mathbf{v}_{(i-1)+}$, the EBVP partials are shown in Table B.1 along with the PST partials in Table B.2 for the various α 's

and r_* 's. Note the partials of Lambert's problem are still distinctly labeled with $\langle \cdot \rangle$ and presented in Ref. [105]. For these more general cost partials in Eqs. (B.9) and (B.10), if $\alpha = 0$, then the cost partials shown in Eqs. (B.7) and (B.8) are reproduced. Also note, the velocity partials of segment $(i-1)i$ are derived similarly to segment $i(i+1)$.

Lastly, to minimize an approximation to thrust-acceleration-squared for Chapter 5, an augmentation to Eqs. (B.9) and (B.10) is needed. The cost function in Eq. (5.10) is repeated here for convenience, dropping the subscript \mathbb{D} for clarity,

$$J = \frac{1}{2} \sum_{i=1}^n \Gamma_i^2 \Delta t_{\text{avg},i} = \frac{1}{2} \sum_{i=1}^n \frac{\Delta v_i^2}{\Delta t_{\text{avg},i}}$$

where $\Gamma_i = \Delta v_i / \Delta t_{\text{avg},i}$ and $\Delta t_{\text{avg},i} = (\Delta t_{(i-1)i} + \Delta t_{i(i+1)})/2$. Notice the numerator of this cost function is Δv_i^2 . According to the chain rule, the partials of Eqs. (B.9) and (B.10) are still relevant, but now an additional term is needed to account for the term in the denominator, $\Delta t_{\text{avg},i}$, that is function of $\Delta\tau$ and \mathbf{r}_i .

$$\begin{aligned} \frac{dJ}{d\Delta\tau} &= \frac{\partial J}{\partial \Delta v_1} \frac{\partial \Delta v_1}{\partial \Delta \mathbf{v}_1} \frac{\partial \mathbf{v}_{1+}}{\partial \Delta\tau} + \sum_{i=2}^{n-1} \frac{\partial J}{\partial \Delta v_i} \frac{\partial \Delta v_i}{\partial \Delta \mathbf{v}_i} \left(\frac{\partial \mathbf{v}_{i+}}{\partial \Delta\tau} - \frac{\partial \mathbf{v}_{i-}}{\partial \Delta\tau} \right) \\ &\quad + \frac{\partial J}{\partial \Delta v_n} \frac{\partial \Delta v_n}{\partial \Delta \mathbf{v}_n} \left(-\frac{\partial \mathbf{v}_{n-}}{\partial \Delta\tau} \right) \\ &\quad + \frac{\partial J}{\partial \Delta t_{\text{avg},1}} \left(\frac{1}{2} \frac{\partial \Delta t_{12}}{\partial \Delta\tau} \right) + \sum_{i=2}^{n-1} \frac{\partial J}{\partial \Delta t_{\text{avg},i}} \left(\frac{1}{2} \frac{\partial \Delta t_{(i-1)i}}{\partial \Delta\tau} + \frac{1}{2} \frac{\partial \Delta t_{i(i+1)}}{\partial \Delta\tau} \right) \\ &\quad + \frac{\partial J}{\partial \Delta t_{\text{avg},n}} \left(\frac{1}{2} \frac{\partial \Delta t_{(n-1)n}}{\partial \Delta\tau} \right) \end{aligned}$$

$$\begin{aligned}
\frac{dJ}{d\mathbf{r}_i} &= \frac{\partial J}{\partial \Delta v_{i-1}} \frac{\partial \Delta v_{i-1}}{\partial \Delta \mathbf{v}_{i-1}} \frac{\partial \mathbf{v}_{(i-1)+}}{\partial \mathbf{r}_i} + \frac{\partial J}{\partial \Delta v_i} \frac{\partial \Delta v_i}{\partial \Delta \mathbf{v}_i} \left(\frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_i} - \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_i} \right) \\
&\quad + \frac{\partial J}{\partial \Delta v_{i+1}} \frac{\partial \Delta v_{i+1}}{\partial \Delta \mathbf{v}_{i+1}} \left(-\frac{\partial \mathbf{v}_{(i+1)-}}{\partial \mathbf{r}_i} \right) \\
&\quad + \frac{\partial J}{\partial \Delta t_{\text{avg},i-1}} \left(\frac{1}{2} \frac{\partial \Delta t_{(i-1)i}}{\partial \mathbf{r}_i} \right) + \frac{\partial J}{\partial \Delta t_{\text{avg},i}} \left(\frac{1}{2} \frac{\partial \Delta t_{(i-1)i}}{\partial \mathbf{r}_i} + \frac{1}{2} \frac{\partial \Delta t_{i(i+1)}}{\partial \mathbf{r}_i} \right) \\
&\quad + \frac{\partial J}{\partial \Delta t_{\text{avg},i+1}} \left(\frac{1}{2} \frac{\partial \Delta t_{i(i+1)}}{\partial \mathbf{r}_i} \right)
\end{aligned}$$

Table B.1: Segment $i(i+1)$ partials for \mathbf{r}_i and τ_i for the PST.

Initial Velocity \mathbf{v}_{i+} Partial	Final Velocity $\mathbf{v}_{(i+1)-}$ Partial	Real Time of Flight $\Delta t_{i(i+1)}$
Initial \mathbf{r}_i and Final \mathbf{r}_{i+1} Position		
$\frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_i} = \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_i} \right\rangle + \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial t_{(i+1)-}} \right\rangle \otimes \frac{\partial \Delta t_{i(i+1)}}{\partial \mathbf{r}_i}$	$\frac{\partial \mathbf{v}_{(i+1)-}}{\partial \mathbf{r}_i} = \left\langle \frac{\partial \mathbf{v}_{(i+1)-}}{\partial \mathbf{r}_i} \right\rangle + \left\langle \frac{\partial \mathbf{v}_{(i+1)-}}{\partial t_{(i+1)-}} \right\rangle \otimes \frac{\partial \Delta t_{i(i+1)}}{\partial \mathbf{r}_i}$	$\frac{\partial t_{(i+1)-}}{\partial \mathbf{r}_i} = \frac{\partial \Delta t_{i(i+1)}}{\partial \mathbf{r}_i} = \frac{\partial \Delta t_{i(i+1)}}{\partial r_*} \frac{\partial r_*}{\partial \mathbf{r}_i}$
$\frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_{i+1}} = \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_{i+1}} \right\rangle + \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial t_{(i+1)-}} \right\rangle \otimes \frac{\partial \Delta t_{i(i+1)}}{\partial \mathbf{r}_i}$	$\frac{\partial \mathbf{v}_{(i+1)-}}{\partial \mathbf{r}_{i+1}} = \left\langle \frac{\partial \mathbf{v}_{(i+1)-}}{\partial \mathbf{r}_{i+1}} \right\rangle + \left\langle \frac{\partial \mathbf{v}_{(i+1)-}}{\partial t_{(i+1)-}} \right\rangle \otimes \frac{\partial \Delta t_{i(i+1)}}{\partial \mathbf{r}_{i+1}}$	$\frac{\partial t_{(i+1)-}}{\partial \mathbf{r}_{i+1}} = \frac{\partial \Delta t_{i(i+1)}}{\partial \mathbf{r}_{i+1}} = \frac{\partial \Delta t_{i(i+1)}}{\partial r_*} \frac{\partial r_*}{\partial \mathbf{r}_{i+1}}$
Initial τ_{i+} and Final $\tau_{(i+1)-}$ Pseudo Time		
$\frac{\partial \mathbf{v}_{i+}}{\partial \tau_{i+}} = - \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial t_{(i+1)-}} \right\rangle \frac{\partial \Delta t_{i(i+1)}}{\partial \Delta \tau}$	$\frac{\partial \mathbf{v}_{(i+1)-}}{\partial \tau_{i+}} = - \left\langle \frac{\partial \mathbf{v}_{(i+1)-}}{\partial t_{(i+1)-}} \right\rangle \frac{\partial \Delta t_{i(i+1)}}{\partial \Delta \tau}$	$\frac{\partial t_{(i+1)-}}{\partial \tau_{i+}} = - \frac{\partial \Delta t_{i(i+1)}}{\partial \Delta \tau}$
$\frac{\partial \mathbf{v}_{i+}}{\partial \tau_{(i+1)-}} = \left\langle \frac{\partial \mathbf{v}_{i+}}{\partial t_{(i+1)-}} \right\rangle \frac{\partial \Delta t_{i(i+1)}}{\partial \Delta \tau}$	$\frac{\partial \mathbf{v}_{(i+1)-}}{\partial \tau_{(i+1)-}} = \left\langle \frac{\partial \mathbf{v}_{(i+1)-}}{\partial t_{(i+1)-}} \right\rangle \frac{\partial \Delta t_{i(i+1)}}{\partial \Delta \tau}$	$\frac{\partial t_{(i+1)-}}{\partial \tau_{(i+1)-}} = \frac{\partial \Delta t_{i(i+1)}}{\partial \Delta \tau}$

Table B.2: Auxiliary segment $i(i+1)$ partials for α and r_* of the PST.

Real Time $\alpha = 0$	Ecc. Anomaly $\alpha = 1$	Intermediate Anomaly $\alpha = 3/2$	True Anomaly $\alpha = 2$
$\frac{\partial \Delta t_{i(i+1)}}{\partial \Delta \tau} = 1$	$\frac{\partial \Delta t_{i(i+1)}}{\partial \Delta \tau} = r_*$	$\frac{\partial \Delta t_{i(i+1)}}{\partial \Delta \tau} = r_*^{3/2}$	$\frac{\partial \Delta t_{i(i+1)}}{\partial \Delta \tau} = r_*^2$
$\frac{\partial \Delta t_{i(i+1)}}{\partial r_*} = 0$	$\frac{\partial \Delta t_{i(i+1)}}{\partial r_*} = \Delta \tau$	$\frac{\partial \Delta t_{i(i+1)}}{\partial r_*} = \frac{3}{2} r_*^{1/2} \Delta \tau$	$\frac{\partial \Delta t_{i(i+1)}}{\partial r_*} = 2 r_* \Delta \tau$
Initial Position $r_* = r_i$	Final Position $r_* = r_{i+1}$	Average Position $r_* = (r_i + r_{i+1})/2$	
$\frac{\partial r_*}{\partial \mathbf{r}_i} = \frac{\mathbf{r}_i}{r_i}$	$\frac{\partial r_*}{\partial \mathbf{r}_i} = \mathbf{0}_{1 \times 3}$	$\frac{\partial r_*}{\partial \mathbf{r}_i} = \frac{1}{2} \frac{\mathbf{r}_i}{r_i}$	
$\frac{\partial r_*}{\partial \mathbf{r}_{i+1}} = \mathbf{0}_{1 \times 3}$	$\frac{\partial r_*}{\partial \mathbf{r}_{i+1}} = \frac{\mathbf{r}_{i+1}}{r_{i+1}}$	$\frac{\partial r_*}{\partial \mathbf{r}_{i+1}} = \frac{1}{2} \frac{\mathbf{r}_{i+1}}{r_{i+1}}$	

B.1.1 Variational State Transition Matrix

The EBVP partials, designated by $\langle \cdot \rangle$, can be derived either directly when using ivLam, or through the variational STM via integration for rk7sh and rk78sh or by forming the Keplerian submatrices, e.g. via Battin’s Kepler variational STM implemented in Arora et al. [105], or other analytic forms of two-body dynamics by Shepperd [258], Goodyear [259], Pitkin [260], Glandorf [261], and more.

The variational STM solves the following variational equations of motion for a segment $\delta\dot{\mathbf{x}} = \mathbf{F}\delta\mathbf{x}$ for dynamics of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ so $\mathbf{F} = \partial\mathbf{f}/\partial\mathbf{x}$. The variational STM can then be written as

$$\delta\mathbf{x}_{(i+1)-} = \mathbf{\Phi}(t_{(i+1)-}, t_{i+})\delta\mathbf{x}_{i+} \quad (\text{B.11})$$

for segment $(i+1)i$ from t_{i+} to $t_{(i+1)-}$ or node i to $i+1$. The variational STM $\mathbf{\Phi} = \mathbf{\Phi}(t_{(i+1)-}, t_{i+})$ describes the effect of the sensitivity of the initial state, time-fixed variation $\delta\mathbf{x}_{i+} = \delta\mathbf{x}(t_{i+})$ on a final state, time-fixed variation $\delta\mathbf{x}_{(i+1)-} = \delta\mathbf{x}(t_{(i+1)-})$.

The variational STM is then either numerically integrated with $\dot{\mathbf{\Phi}} = \mathbf{F}\mathbf{\Phi}$ or analytically known. The integrated solution is decomposed into sub-matrix form as

$$\mathbf{\Phi} = \frac{\partial\mathbf{x}_{(i+1)-}}{\partial\mathbf{x}_{i+}} = \begin{bmatrix} \frac{\partial\mathbf{r}_{(i+1)-}}{\partial\mathbf{r}_{i+}} & \frac{\partial\mathbf{r}_{(i+1)-}}{\partial\mathbf{v}_{i+}} \\ \frac{\partial\mathbf{v}_{(i+1)-}}{\partial\mathbf{r}_{i+}} & \frac{\partial\mathbf{v}_{(i+1)-}}{\partial\mathbf{v}_{i+}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{(i+1)i} & \mathbf{B}_{(i+1)i} \\ \mathbf{C}_{(i+1)i} & \mathbf{D}_{(i+1)i} \end{bmatrix} \quad (\text{B.12})$$

where the sub-matrices $\mathbf{A}_{(i+1)i}$, $\mathbf{B}_{(i+1)i}$, $\mathbf{C}_{(i+1)i}$, and $\mathbf{D}_{(i+1)i}$ are defined for notational simplicity. A closed-form variational STM for two-body dynamics by Battin is not reproduced here for compactness but can be found in Ref. [105] via Eqs. 38–50. A similar form of the variational STM can be made for segment $i(i-1)$. The EBVP partials in terms of the variational STM sub-matrices are shown in Table B.3. The

expressions in Table B.3 are primarily for perturbed two-body dynamics, however the submatrices can be found analytically as previously mentioned for the two-body problem to give the partials of the Lambert problem.

Table B.3: EBVP partials in terms of the variational STM

Position	Time
Velocity Partial for Segment $(i+1)i$ from Node i to $i+1$	
$\left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_i} \right\rangle = -\mathbf{B}_{(i+1)i}^{-1} \mathbf{A}_{(i+1)i}$	$\left\langle \frac{\partial \mathbf{v}_{i+}}{\partial t_{i+}} \right\rangle = \mathbf{a}_{i+} + \mathbf{B}_{(i+1)i}^{-1} \mathbf{A}_{(i+1)i} \mathbf{v}_{i+}$
$\left\langle \frac{\partial \mathbf{v}_{i+}}{\partial \mathbf{r}_{i+1}} \right\rangle = \mathbf{B}_{(i+1)i}^{-1}$	$\left\langle \frac{\partial \mathbf{v}_{i+}}{\partial t_{(i+1)-}} \right\rangle = -\mathbf{B}_{(i+1)i}^{-1} \mathbf{v}_{(i+1)-}$
Velocity Partial for Segment $i(i-1)$ from Node $i-1$ to i	
$\left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_{i-1}} \right\rangle = \mathbf{C}_{i(i-1)} - \mathbf{D}_{i(i-1)} \mathbf{B}_{i(i-1)}^{-1} \mathbf{A}_{i(i-1)}$	$\left\langle \frac{\partial \mathbf{v}_{i-}}{\partial t_{(i-1)+}} \right\rangle = -(\mathbf{C}_{i(i-1)} - \mathbf{D}_{i(i-1)} \mathbf{B}_{i(i-1)}^{-1} \mathbf{A}_{i(i-1)}) \mathbf{v}_{(i-1)+}$
$\left\langle \frac{\partial \mathbf{v}_{i-}}{\partial \mathbf{r}_i} \right\rangle = \mathbf{D}_{i(i-1)} \mathbf{B}_{i(i-1)}^{-1}$	$\left\langle \frac{\partial \mathbf{v}_{i-}}{\partial t_{i-}} \right\rangle = \mathbf{a}_{i-} - \mathbf{D}_{i(i-1)} \mathbf{B}_{i(i-1)}^{-1} \mathbf{v}_{i-}$

B.1.2 Mass-Leak

A mass-leak numerically fixes a singularity of the cost partials for fuel optimal trajectories, not needed for energy optimal trajectories. If a node is not needed, the denominator of partial $\partial \Delta v_i / \partial \Delta \mathbf{v}_i = \Delta \mathbf{v}_i^\top / \Delta v_i$ goes to zero. The mass-leak eliminates this singularity by including a small constant quantity ϵ inside the computation of the magnitude of the impulsive maneuver:

$$\Delta v_i = \sqrt{\Delta v_{x_i}^2 + \Delta v_{y_i}^2 + \Delta v_{z_i}^2 + \epsilon^2} \quad (\text{B.13})$$

where Δv_{x_i} , Δv_{y_i} , and Δv_{z_i} are the x , y , and z -components of the $\Delta \mathbf{v}_i$, respectively. Practical choices for ϵ are $\epsilon = 10^{-4}, 10^{-5}$, and 10^{-6} DU/TU. Also, in practice, the optimizer decreases the value of the $\Delta \mathbf{v}_i$ to approximately ϵ if not needed for optimality because the gradient-based optimizer does not recognize the node as affecting the cost once a partial is approximately zero. This mass-leak causes a slightly higher $\Delta \mathbf{v}_i$ in the cost function of Eq. (3.1), but automatically eliminates nodes. For a fuel optimal solution, the true cost, computed without a mass-leak, can be easily post-processed, using the now known correct number of $\Delta \mathbf{v}$'s.

Appendix C

List of Publications

C.1 Journal Publications

Chapter 3

D. Ottesen and R. P. Russell, “Unconstrained Direct Optimization of Spacecraft Trajectories Using Many Embedded Lambert Problems,” *Journal of Optimization Theory and Applications*, 2021, pp. 1–41. doi:10.1007/s10957-021-01884-1.

N. Arora, R. P. Russell, N. Strange, and **D. Ottesen**, “Partial Derivatives of the Solution to the Lambert Boundary Value Problem,” *Journal of Guidance, Control, and Dynamics*, Vol. 38, Sept. 2015, pp. 1563–1572. doi:10.2514/1.G001030.

Chapter 4

D. Ottesen and R. P. Russell, “Piecewise Sundman Transformation for Spacecraft Trajectory Optimization Using Many Embedded Lambert Problems,” *Journal of Spacecraft and Rockets*, 2022, pp. 1–18. doi:10.2514/1.A35140.

Miscellaneous

R. P. Russell, S. McArdle, **D. Ottesen**, E. M. Zucchelli, and W. E. Brandenburg, “Global Trajectory Optimization, Pathfinding, and Scheduling for a Multi-Flyby, Multi-Spacecraft Mission,” *Acta Astronautica*, July 2022. doi:10.1016/j.actaastro.2022.07.007.

C.2 Conference Proceedings

Chapter 3

D. Ottesen and R. P. Russell, “Space Trajectory Optimization using Embedded Boundary Value Problems,” *AAS/AIAA Astrodynamics Specialist Conference*, AAS 17-837, Vol. 162, Stevenson, WA, August 2017, pp. 2271–2290.

D. Ottesen and R. P. Russell, “Unconstrained Spacecraft Trajectory Optimization using Embedded Boundary Value Problems,” *AAS/AIAA Astrodynamics Specialist Conference*, AAS 19-929, Portland, ME, August 2019.

Chapter 4

D. Ottesen and R. P. Russell, “A Piecewise-Constant Sundman Transformation for Spacecraft Trajectory Optimization,” *AAS/AIAA Astrodynamics Specialist Conference*, AAS 18-467, Vol. 167, Snowbird, WA, August 2018, pp. 1755–1774.

Chapter 5

D. Ottesen and R. P. Russell, “Direct-to-Indirect Mapping for Optimal Low-Thrust Trajectories,” *AAS/AIAA Astrodynamics Specialist Conference*, AAS 22-227, Charlotte, NC, August 2022.

Bibliography

- [1] Constance McLaughlin Green and Milton Lomask. *Vanguard, A History*. NASA Special Publication, NASA-SP-4202, Washington, DC, 1970.
- [2] Ted Spitzmiller. *The History of Human Space Flight*. University Press of Florida, Gainesville, FL, 2017.
- [3] V. D. Kuznetsov, V. M. Sinelnikov, and S. N. Alpert. Yakov alpert: Sputnik-1 and the first satellite ionospheric experiment. *Advances in Space Research*, 55(12):2833–2839, 2015. doi:10.1016/j.asr.2015.02.033.
- [4] NASA Space Science Data Coordinated Archive. Telstar 1, NSS-DCA/COSPAR ID: 1962-029A Accessed June 15, 2022.
- [5] Richard M. Bentley and Albert T. Owens. Syncom satellite program. *Journal of Spacecraft and Rockets*, 1(4):395–399, 1964. doi:10.2514/3.27667.
- [6] David Whalen. The origins of satellite communications 1945–1965. In *41st Aerospace Sciences Meeting and Exhibit, AIAA 2003-661*, Reno, Nevada, January 2003. doi:10.2514/6.2003-661.
- [7] Kevin C. Ruffner. *Corona: America’s First Satellite Program*. History Staff, Center for the Study of Intelligence, Central Intelligence Agency, Washington, DC, 1995.

- [8] P. Gorin. Zenit—the first soviet photo-reconnaissance satellite. *Journal of the British Interplanetary Society*, 50(11):441, 1997.
- [9] Peter A. Gorin. Zenit: Corona’s soviet counterpart. *Corona between the Sun and the Earth: The First NRO Reconnaissance Eye in Space*. Bethesda, MD: American Society for Photogrammetry and Remote Sensing, pages 85–107, 1997.
- [10] National Coordination Office for Space-Based Positioning, Navigation, and Timing. GPS: Performance Standards & Specifications Accessed June 15, 2022.
- [11] Asif A. Siddiqi. Beyond earth: A chronicle of deep space exploration, 1958–2016. Technical report, NASA SP: 2018-4041, National Aeronautics and Space Administration, Washington, DC, 2018.
- [12] Steven J. Goodman, Timothy J. Schmit, Jaime Daniels, and Robert J. Redmon. Goes-r series summary and look ahead. In *The GOES-R Series*, pages 273–277. Elsevier, 2020. doi:10.1016/B978-0-12-814327-8.00023-8.
- [13] Robert Thirsk, Andre Kuipers, Chiaki Mukai, and David William. The space-flight environment: The international space station and beyond. *CMAJ*, 180(12):1216–1220, 2009. doi:10.1503/cmaj.081125.
- [14] Wiley J. Larson, James Richard Wertz, , Douglas Kirkpatrick, and Donna Klungle. *Space Mission Analysis and Design*. Microcosm Press & Kluwer Academic Publishers, third edition, 2005.
- [15] Nitin Arora, Javier Roa, Anastassios E. Petropoulos, and Nathan Strange. Time-free targeting for direct trajectory optimization. In AAS/AIAA *Astro-*

- dynamics Specialist Conf, AAS 17-842*, volume 162, pages 717–726, Stevenson, WA, August 2017.
- [16] Ryan Paul Russell. On the solution to every lambert problem. *Celestial Mechanics and Dynamical Astronomy*, 131(50), 2019. doi:10.1007/s10569-019-9927-z.
- [17] Ryan Paul Russell. ivlam. (1.06). Zenodo., October 2019.
- [18] Ryan Paul Russell. Complete lambert solver including second-order sensitivities. *Journal of Guidance, Control, and Dynamics*, 45(2):196–212, 2022. doi:10.2514/1.G006089.
- [19] C. E. Kohlhase and P. A. Penzo. Voyager mission description. *Space Science Reviews*, 21(2):77–101, 1977. doi:10.1007/BF00200846.
- [20] Smith, Bradford A., et al. Encounter with saturn: Voyager 1 imaging science results. *Science*, 212(4491):163–191, 1981. doi:10.1126/science.212.4491.163.
- [21] Smith, Bradford A., et al. A new look at the saturn system: The voyager 2 images. *Science*, 215(4532):504–537, 1982. doi:10.1126/science.215.4532.504.
- [22] Ken Galal, Tony Colaprete, Steven Cooley, Brian Kennedy, and Tim McElrath. Trajectory Design and Orbit determination for the Lunar Crater Observation and Sensing Satellite (LCROSS). In *Proceedings of the 20th International Symposium on Space Flight Dynamics*, Annapolis, MD, 2007.
- [23] Paul Tompkins, Rusty Hunt, Matthew D’Ortenzio, Ken Galal, Darin Foreman, James Munger, Mark Shirley, James Strong, Robert Barber, and Eric Drucker. Flight operations for the lcross lunar impactor mission. In *SpaceOps 2010 Conference Delivering on the Dream, Hosted by NASA Marshall Space Flight Cen-*

- ter, *AIAA 2010-1986*, Huntsville, Alabama, April 2010. doi:10.2514/6.2010-1986.
- [24] Steve Cooley, Ken Galal, Kevin Berry, Leigh Janes, Greg Marr, John Carrico, and Cesar Ocampo. Mission Design for the Lunar Crater Observation and Sensing Satellite (LCROSS). In *AIAA/AAS Astrodynamics Specialist Conference, AIAA 2010-8386*, Toronto, Ontario, Canada, August 2010. doi:10.2514/6.2010-8386.
- [25] Case, A. W., et al. The solar probe plus mission: Humanity’s first visit to our star. *Space Science Reviews*, 204(1):7–48, 2016. doi:10.3847/1538-4365/ab5a7b.
- [26] Sean C. Solomon, Ralph L. McNutt, Robert E. Gold, and Deborah L. Domingue. Messenger mission overview. *Space Science Reviews*, 131(1):3–39, 2007. doi:10.1007/s11214-007-9247-6.
- [27] James V. McAdams, David W. Dunham, Robert W. Farquhar, Anthony H. Taylor, and B. G. Williams. Trajectory design and maneuver strategy for the messenger mission to mercury. *Journal of Spacecraft and Rockets*, 43(5):1054–1064, 2006. doi:10.2514/1.18178.
- [28] David Dunham, James McAdams, Dawn Moessner, and David Ottesen. Contingency plans for messenger’s mercury orbit insertion maneuver. August 2010. doi:10.2514/6.2010-8252.
- [29] Kaan Çelikbilek, Zainab Saleem, Ruben Morales Ferre, Jaan Praks, and Elena Simona Lohan. Survey on optimization methods for leo-satellite-based networks with applications in future autonomous transportation. *Sensors*, 22(4):1421, 2022. doi:10.3390/s22041421.

- [30] Marc D. Rayman, Pamela A. Chadbourne, Jeffery S. Culwell, and Steven N. Williams. Mission design for deep space 1: A low-thrust technology validation mission. *Acta Astronautica*, 45(4-9):381–388, 1999. doi:10.1016/S0094-5765(99)00157-5.
- [31] Marc D. Rayman, Philip Varghese, David H. Lehman, and Leslie L. Livesay. Results from the deep space 1 technology validation mission. *Acta Astronautica*, 47(2-9):475–487, 2000. doi:10.1016/S0094-5765(00)00087-4.
- [32] Racca, G.D., et al. Smart-1 mission description and development status. *Planetary and Space Science*, 50(14-15):1323–1337, 2002. doi:10.1016/S0032-0633(02)00123-X.
- [33] Charles E. Garner, Mark M. Rayman, Greg J. Whiffen, John R. Brophy, and Steven C. Mikes. Ion propulsion: An enabling technology for the dawn mission. In *AAS/AIAA Space Flight Mechanics Meeting, AAS 13–342*, Kauai, Hawaii, 2013.
- [34] Christopher T. Russell and Carol A. Raymond. *The Dawn Mission to Vesta and Ceres*, pages 3–23. Springer New York, New York, NY, 2012. doi:10.1007/978-1-4614-4903-4_2.
- [35] Fujiwara, Akira, et al. The rubble-pile asteroid itokawa as observed by hayabusa. *Science*, 312(5778):1330–1334, 2006. doi:10.1126/science.1125841.
- [36] Yuichi Tsuda, Makoto Yoshikawa, Masanao Abe, Hiroyuki Minamino, and Satoru Nakazawa. System design of the hayabusa 2—asteroid sample return mission to 1999 ju3. *Acta Astronautica*, 91:356–362, 2013. doi:10.1016/j.actaastro.2013.06.028.

- [37] Lauretta, D. S., et al. Osiris-rex: sample return from asteroid (101955) bennu. *Space Science Reviews*, 212(1):925–984, 2017. doi:10.1007/s11214-017-0405-1.
- [38] Kenneth M. Getzandanner, Peter G. Antreasian, Michael C. Moreau, Jason M. Leonard, Coralie D. Adam, Daniel Wibben, Kevin Berry, Dolan Highsmith, and Dante Lauretta. Small body proximity operations & tag: Navigation experiences & lessons learned from the osiris-rex mission. In *The 32nd AIAA/AAS Space Flight Mechanics Meeting*, San Diego, CA, 2022. doi:10.2514/6.2022-2387.
- [39] Williams, B., et al. Osiris-rex flight dynamics and navigation design. *Space Science Reviews*, 214(4):1–43, 2018. doi:10.1007/s11214-018-0501-x.
- [40] Antreasian, P. G., et al. Osiris-rex proximity operations and navigation performance at bennu. In *The 32nd AIAA/AAS Space Flight Mechanics Meeting*, San Diego, CA, 2022.
- [41] Global Trajectory Optimization Competition. GTOC Portal. Accessed June 6, 2022.
- [42] Ryan Paul Russell, Sean McArdle, David Ottesen, Enrico M. Zucchelli, and William E. Brandenburg. Global trajectory optimization, pathfinding, and scheduling for a multi-flyby, multi-spacecraft mission. *Acta Astronautica*, July 2022. doi:10.1016/j.actaastro.2022.07.007.
- [43] John V. Breakwell. The optimization of trajectories. *Journal of the Society for Industrial and Applied Mathematics*, 7(2):215–247, 1959.
- [44] Derek F. Lawden. *Optimal Trajectories for Space Navigation*, volume 3. Butterworths and Co., Butterworths, London, 1963.

- [45] Theodore N. Edelbaum. Propulsion requirements for controllable satellites. *ARS Journal*, 31(8):1079–1089, 1961. doi:10.2514/8.5723.
- [46] Theodore N. Edelbaum. Theory of maxima and minima. In *Mathematics in Science and Engineering*, volume 5, pages 1–32. Elsevier, 1962. doi:10.1016/S0076-5392(08)62089-5.
- [47] Theodore N. Edelbaum. Optimum low-thrust rendezvous and station keeping. *Journal of Spacecraft and Rockets*, 40(6):960–965, 2003. doi:10.2514/2.7042.
- [48] Theodore N. Edelbaum. Optimum power-limited orbit transfer in strong gravity fields. *AIAA Journal*, 3(5):921–925, 1965. doi:10.2514/3.3016.
- [49] Theodore N. Edelbaum. An asymptotic solution for optimum power limited orbit transfer. *AIAA Journal*, 4(8):1491–1494, 1966. doi:10.2514/3.3725.
- [50] Louis A. D’Amario and Theodore N. Edelbaum. Minimum impulse three-body trajectories. *AIAA Journal*, 12(4):455–462, 1974. doi:10.2514/3.49268.
- [51] Elijah Polak. An historical survey of computational methods in optimal control. *SIAM review*, 15(2):553–584, 1973. doi:10.1137/1015071.
- [52] Arthur E. Bryson. Optimal control—1950 to 1985. *IEEE Control Systems Magazine*, 16(3):26–33, 1996. doi:10.1109/37.506395.
- [53] Alessandro De Angelis. *Galileo Galilei’s “Two New Sciences”*: for Modern Readers. Springer Nature, 2021. doi:10.1007/978-3-030-71952-4.
- [54] J. Bernoulli. Problema novum ad cujus solutionem mathematici invitantur. *Acta Eruditorum Lipsi*, page 269, 1696. <https://books.google.com/books?id=4q1RAAAAcAAJ>.

- [55] Oskar von Stryk and Roland Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37(1):357–373, 1992. doi:10.1007/BF02071065.
- [56] John T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998. doi:10.2514/2.4231.
- [57] Anil V. Rao. A survey of numerical methods for optimal control. In *AAS/AIAA Astrodynamics Specialist Conference, AAS 09-334*, Pittsburgh, PA, August 2009.
- [58] William Karush. Minima of functions of several variables with inequalities as side conditions. In Giorgio Giorgi and Tinne Hoff Kjeldsen, editors, *Traces and Emergence of Nonlinear Programming*, pages 217–245. Birkhauser, Basel, Basel, 2014. doi:10.1007/978-3-0348-0439-4_10.
- [59] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the 2nd Berkeley Symposium on Math. Stat. and Prob., 1950*, pages 481–492. University of California Press, Berkeley and Los Angeles, 1951.
- [60] Bruce A. Conway. A survey of methods available for the numerical optimization of continuous dynamic systems. *Journal of Optimization Theory and Applications*, 152(2):271–306, February 2012. doi:10.1007/s10957-011-9918-z.
- [61] Abolfazl Shirazi, Josu Ceberio, and Jose A. Lozano. Spacecraft trajectory optimization: A review of models, objectives, approaches, and solutions. *Progress in Aerospace Sciences*, 102:76–98, 2018. doi:10.1016/j.paerosci.2018.07.007.
- [62] Danylo Malyuta, Yue Yu, Purnanand Elango, and Behçet Açıkmeşe. Advances

- in trajectory optimization for space vehicle control. *Annual Reviews in Control*, 52:282–315, 2021. doi:10.1016/j.arcontrol.2021.04.013.
- [63] David Morante, Manuel Sanjurjo Rivo, and Manuel Soler. A survey on low-thrust trajectory optimization approaches. *Aerospace*, 8(3):88, 2021. doi:10.3390/aerospace8030088.
- [64] Gregory J. Whiffen. Static/dynamic control for optimizing a useful objective, 12 2002. US Patent 6,496,741.
- [65] Gregory Lantoine and Ryan Paul Russell. A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 1: Theory. *Journal of Optimization Theory and Applications*, 154(2):382–417, 2012. doi:10.1007/s10957-012-0039-0.
- [66] Gregory Lantoine and Ryan Paul Russell. A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 2: Application. *Journal of Optimization Theory and Applications*, 154(2):418–442, 2012. doi:10.1007/s10957-012-0038-1.
- [67] Jonathan D. Aziz, Jeffrey S. Parkery, Daniel J. Scheeresz, and Jacob A. Englander. Low-thrust many-revolution trajectory optimization via differential dynamic programming and a sundman transformation. *The Journal of the Astronautical Sciences*, 65(2):205–228, 2018. doi:10.1007/s40295-017-0122-8.
- [68] W. H. Clohessy and R. S. Wiltshire. Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, 1960. doi:10.2514/8.8704.

- [69] Roger Broucke. Stability of periodic orbits in the elliptic, restricted three-body problem. *AIAA journal*, 7(6):1003–1009, 1969. doi:10.2514/3.5267.
- [70] George William Hill. Researches in the lunar theory. *American Journal of Mathematics*, 1(1):5–26, 1878. doi:10.2307/2369430.
- [71] Hanspeter Schaub and John L. Junkins. *Analytical Mechanics of Space Systems*. American Institute of Aeronautics and Astronautics, Inc., Reston, VA, fourth edition, 2018.
- [72] M. J. H. Walker, B. Ireland, and Joyce Owens. A set modified equinoctial orbit elements. *Celestial mechanics*, 36(4):409–419, 1985. doi:10.1007/BF01227493.
- [73] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, 2nd edition, 2010. doi:10.1137/1.9780898718577.
- [74] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA Education Series. American Institute of Aeronautics and Astronautics, Reston, VA, revised edition, 1999. doi:10.2514/4.861543.
- [75] Walter Hohmann. *The Attainability of Heavenly Bodies*. NASA Technical Translation F-44. National Aeronautics and Space Administration, Washington, DC, November 1960.
- [76] Howard D. Curtis. *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, first edition, 2005.
- [77] Jon A. Sims and Steve N. Flanagan. Preliminary design of low-thrust interplanetary missions. In *AAS/AIAA Astrodynamics Specialist Conference, AAS 99-338*, Girdwood, AK, August 1999.

- [78] Yaochu Jin and Bernhard Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(3):397–415, 2008. doi:10.1109/TSMCC.2008.919172.
- [79] Roger R. Bate, Donald D. Mueller, and Jerry E. White. *Fundamentals of Astrodynamics*. Dover Pub., NY, 1971.
- [80] David C. Redding and John V. Breakwell. Optimal low-thrust transfers to synchronous orbit. *Journal of Guidance, Control, and Dynamics*, 7(2):148–155, 1984. doi:10.2514/3.8560.
- [81] Charles R. Hargraves and Stephen W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of guidance, control, and dynamics*, 10(4):338–342, 1987. doi:10.2514/3.20223.
- [82] Dimitri Bertsekas. *Dynamic Programming and Optimal Control: Volume I*, volume 1. Athena Scientific, fourth edition, 2012.
- [83] Scott Kirkpatrick, C Daniel Gelatt Jr., and Mario P. Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983. doi:10.1126/science.220.4598.671.
- [84] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995. doi:10.1109/ICNN.1995.488968.
- [85] Russell Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Sym-*

- posium on Micro Machine and Human Science*, pages 39–43. IEEE, 1995. doi:10.1109/MHS.1995.494215.
- [86] Mauro Pontani and Bruce A. Conway. Particle swarm optimization applied to space trajectories. *Journal of Guidance, Control, and Dynamics*, 33(5):1429–1441, 2010. doi:10.2514/1.48475.
- [87] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, September 2003. doi:10.1145/937503.937505.
- [88] Ning Xiong, Daniel Molina, Miguel Leon Ortiz, and Francisco Herrera. A walk into metaheuristics for engineering optimization: Principles, methods and recent trends. *international journal of computational intelligence systems*, 8(4):606–636, 2015. doi:10.1080/18756891.2015.1046324.
- [89] Marco Dorigo and Christian Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2-3):243–278, 2005. doi:10.1016/j.tcs.2005.05.020.
- [90] Jon A. Sims, Paul A. Finlayson, Edward A. Rinderle, Matthew A. Vavrina, and Theresa D. Kowalkowski. Implementation of a low-thrust trajectory optimization algorithm for preliminary design. In *AAS/AIAA Astrodynamics Specialist Conference, AIAA 2006-6746*, Keystone, Colorado, August 2006. doi:10.2514/6.2006-6746.
- [91] John T. Betts and William P. Huffman. Trajectory optimization on a parallel processor. *Journal of Guidance, Control, and Dynamics*, 14(2):431–439, 1991. doi:10.2514/3.20656.

- [92] Yang Gao and Craig Kluever. Low-thrust interplanetary orbit transfers using hybrid trajectory optimization method with multiple shooting. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, AIAA 2004-5088*, Providence, RI, August 2004. doi:10.2514/6.2004-5088.
- [93] Etienne Pellegrini and Ryan Paul Russell. *Multiple-Shooting Differential Dynamic Programming with Applications to Spacecraft Trajectory Optimization*. PhD thesis, 2017.
- [94] C. E. Velez and A. J. Fuchs. Averaging techniques and their application to orbit determination systems. *AIAA Journal*, 13(1):12–16, 1975. doi:10.2514/3.49622.
- [95] Sophie Geffroy and Richard Epenoy. Optimal low-thrust transfers with constraints—generalization of averaging techniques. *Acta astronautica*, 41(3):133–149, 1997. doi:10.1016/S0094-5765(97)00208-7.
- [96] Yang Gao and Craig Kluever. Analytic orbital averaging technique for computing tangential-thrust trajectories. *Journal of Guidance, Control, and Dynamics*, 28(6):1320–1323, 2005. doi:10.2514/1.14698.
- [97] B. Bonnard and G. Janin. Geometric orbital transfer using averaging techniques. *Journal of Dynamical and Control Systems*, 14(2):145–167, 2008. doi:10.1007/s10883-008-9040-1.
- [98] Jean A Kéchichian. Optimal low-thrust transfer in general circular orbit using analytic averaging of the system dynamics. *The Journal of the Astronautical Sciences*, 57(1-2):369–392, 2009. doi:10.1007/BF03321509.
- [99] Wayne A. Scheel and Bruce A. Conway. Optimization of very-low-thrust,

- many-revolution spacecraft trajectories. *Journal of Guidance, Control, and Dynamics*, 17(6):1185–1192, 1994. doi:10.2514/3.21331.
- [100] Thomas Haberkorn, Pierre Martinon, and Joseph Gergaud. Low-thrust minimum-fuel orbital transfer: A homotopic approach. *Journal of Guidance, Control, and Dynamics*, 27(6):1046–1060, 2004. doi:10.2514/1.4022.
- [101] Gao Yang. Direct optimization of low-thrust many-revolution earth-orbit transfers. *Chinese Journal of Aeronautics*, 22(4):426–433, 2009. doi:10.1016/S1000-9361(08)60121-1.
- [102] Kathryn F. Graham and Anil V. Rao. Minimum-time trajectory optimization of multiple revolution low-thrust earth-orbit transfers. *Journal of Spacecraft and Rockets*, 52(3):711–727, 2015. doi:10.2514/1.A33187.
- [103] Kathryn F. Graham and Anil V. Rao. Minimum-time trajectory optimization of low-thrust earth-orbit transfers with eclipsing. *Journal of Spacecraft and Rockets*, 53(2):289–303, 2016. doi:10.2514/1.A33416.
- [104] Jackson L. Shannon, Martin T. Ozimek, Justin A. Atchison, and Christine M. Hartzell. Q-law aided direct trajectory optimization of many-revolution low-thrust transfers. *Journal of Spacecraft and Rockets*, pages 1–11, 2020. doi:10.2514/1.A34586.
- [105] Nitin Arora, Ryan Paul Russell, Nathan Strange, and David Ottesen. Partial derivatives of the solution to the lambert boundary value problem. *Journal of Guidance, Control, and Dynamics*, 38(9):1563–1572, September 2015. doi:10.2514/1.G001030.

- [106] Brandon A Jones and Rodney L Anderson. A survey of symplectic and collocation integration methods for orbit propagation. In *22nd Annual AAS/AIAA Space Flight Mechanics Meeting, AAS 12-214*, Charleston, SC, January 2012. <https://ntrs.nasa.gov/search.jsp?R=20130000293>.
- [107] Chit Hong Yam, Dario Izzo, and Francesco Biscani. Towards a high fidelity direct transcription method for optimisation of low-thrust trajectories. In *4th International Conference on Astrodynamics Tools and Techniques, ICATT*, pages 1–7, April 2010.
- [108] Uri M. Ascher, Robert M. M. Mattheij, and Robert D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. SIAM, 1995. doi:10.1137/1.9781611971231.
- [109] P. M. Lion and M. Handelsman. Primer vector on fixed-time impulsive trajectories. *AIAA Journal*, 6(1):127–132, 1968. doi:10.2514/3.4452.
- [110] Donald J Jezewski and H. L. Rozendaal. An efficient method for calculating optimal free-space n-impulse trajectories. *AIAA Journal*, 6(11):2160–2165, 1968. doi:10.2514/3.4949.
- [111] Donald J. Jezewski. Primer vector theory and applications. Technical report, Technical Report NASA TR R-454, Lyndon B Johnson Space Center, Houston, Texas, 77058, November 1975. <https://ntrs.nasa.gov/search.jsp?R=19760004112>.
- [112] Donald J Jezewski. Primer vector theory applied to the linear relative-motion equations. *Optimal Control Applications and Methods*, 1(4):387–401, 1980. doi:10.1002/oca.4660010408.

- [113] Louis A D’Amario, D V Byrnes, L. L. Sackett, and R. H. Stanford. Optimization of multiple flyby trajectories. Provincetown, Mass., June 1979. <https://ntrs.nasa.gov/search.jsp?R=19790063226>.
- [114] Louis A. D’Amario, D. V. Byrnes, and R. H. Stanford. A new method for optimizing multiple-flyby trajectories. *Journal of Guidance, Control, and Dynamics*, 4(5):591–596, 1981. doi:10.2514/3.56115.
- [115] Damon Landau. Efficient maneuver placement for automated trajectory design. *Journal of Guidance, Control, and Dynamics*, 41(7):1531–1541, 2018. doi:10.2514/1.G003172.
- [116] Kathryn E. Davis, Rodney L. Anderson, Daniel J. Scheeres, and George H. Born. Optimal transfers between unstable periodic orbits using invariant manifolds. *Celestial Mechanics and Dynamical Astronomy*, 109(3):241–264, 2011. doi:10.1007/s10569-010-9327-x.
- [117] Kevin A Bokelmann and Ryan Paul Russell. Optimization of impulsive europa capture trajectories using primer vector theory. *The Journal of the Astronautical Sciences*, May 2019. doi:10.1007/s40295-018-00146-z.
- [118] L A Hiday-Johnston and Kathleen C Howell. Transfers between libration-point orbits in the elliptic restricted problem. *Celestial Mechanics and Dynamical Astronomy*, 58(4):317–337, 1994. doi:10.1007/BF00692008.
- [119] Belinda G. Marchand, Kathleen C. Howell, and Roby S. Wilson. Improved corrections process for constrained trajectory design in the n-body problem. *Journal of Spacecraft and Rockets*, 44(4):884–897, 2007. doi:10.2514/1.27205.
- [120] P. Di Lizia, R. Armellin, and M. Lavagna. Application of high order expansions

- of two-point boundary value problems to astrodynamics. *Celestial Mechanics and Dynamical Astronomy*, 102(4):355–375, 2008. doi:10.1007/s10569-008-9170-5.
- [121] Louis A. D’Amario, Larry E. Bright, and Aron A. Wolf. Galileo trajectory design. *Space Science Reviews*, 60(1-4):23–78, 1992. doi:10.1007/BF00216849.
- [122] Fernando Peralta and Steve Flanagan. Cassini interplanetary trajectory design. *Control Engineering Practice*, 3(11):1603–1610, 1995. doi:10.1016/0967-0661(95)00171-P.
- [123] Nathan Strange, Troy Goodson, and Yungsun Hahn. Cassini tour redesign for the huygens mission. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, AIAA 2002-4720*, Monterey, California, August 2002. doi:10.2514/6.2002-4720.
- [124] NASA Planetary Data System. Cassini-Huygens Mission Information. Accessed Mar. 3, 2021.
- [125] Brent Buffington, Nathan Strange, and John Smith. Overview of the cassini extended mission trajectory. In *AAS/AIAA Astrodynamics Specialist Conference and Exhibit, AIAA 2008-6752*, Honolulu, Hawaii, August 2008. doi:10.2514/6.2008-6752.
- [126] John Smith and Brent Buffington. Overview of the cassini solstice mission trajectory. In *AAS/AIAA Advances in the Astronautical Sciences, AAS 2009-351*, volume 135, Pittsburgh,PA, August 2009.
- [127] Brent Buffington. Trajectory design concept for the proposed europa clipper

- mission. In *AAS/AIAA Astrodynamics Specialist Conference, 2014-4105*, San Diego, CA, August 2014. doi:10.2514/6.2014-4105.
- [128] Ryan Paul Russell and Cesar A Ocampo. Optimization of a broad class of ephemeris model earth-mars cyclers. *Journal of Guidance, Control, and Dynamics*, 29(2):354–367, 2006. doi:10.2514/1.13652.
- [129] Dario Izzo, Victor M Becerra, Darren R Myatt, Slawomir J Nasuto, and J Mark Bishop. Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. *Journal of Global Optimization*, 38(2):283–296, 2007. doi:10.1007/s10898-006-9106-0.
- [130] Ossama Abdelkhalik and Daniele Mortari. N-impulse orbit transfer using genetic algorithms. *Journal of Spacecraft and Rockets*, 44(2):456–460, 2007. doi:10.2514/1.24701.
- [131] Massimiliano Vasile, Matteo Ceriotti, Gianmarco Radice, Victor M Becerra, Slawomir J Nasuto, and James D Anderson. Global trajectory optimisation: Can we prune the solution space when considering deep space manoeuvres? 2008.
- [132] Alfred E. Lynam. Broad-search algorithms for the spacecraft trajectory design of callisto-ganymede-io triple flyby sequences from 2024 to 2040, part ii: Lambert pathfinding and trajectory solutions. *Acta Astronautica*, 94(1):253–261, 2014. doi:10.1016/j.actaastro.2013.07.020.
- [133] Alfred E. Lynam. Broad-search algorithms for the spacecraft trajectory design of callisto-ganymede-io triple flyby sequences from 2024 to 2040, part i: Heuristic pruning of the search space. *Acta Astronautica*, 94(1):246–252, 2014. doi:10.1016/j.actaastro.2013.07.018.

- [134] Daniele Mortari. The theory of connections: Connecting points. *Mathematics*, 5(4):57, 2017. doi:10.3390/math5040057.
- [135] Roberto Furfaro and Daniele Mortari. Least-squares solution of a class of optimal space guidance problems via theory of connections. *Acta Astronautica*, 168:92–103, 2020. doi:10.1016/j.actaastro.2019.05.050.
- [136] Ralph Hoyt Bacon. Logarithmic spiral: An ideal trajectory for the interplanetary vehicle with engines of low sustained thrust. *American Journal of Physics*, 27(3):164–165, 1959. doi:10.1119/1.1934788.
- [137] Ehsan Taheri and Ossama Abdelkhalik. Shape-based approximation of constrained low-thrust space trajectories using fourier series. *Journal of Spacecraft and Rockets*, 49(3):535–546, 2012. doi:10.2514/1.58789.
- [138] Ossama Abdelkhalik and Elsan Taheri. Approximate on-off low-thrust space trajectories using fourier series. *Journal of Spacecraft and Rockets*, 49(5):962–965, 2012. doi:10.2514/1.A32307.
- [139] Qun Fang, Xuefeng Wang, Chong Sun, and Jianping Yuan. A shape-based method for continuous low-thrust trajectory design between circular coplanar orbits. *International Journal of Aerospace Engineering*, 2017, 2017. doi:10.1155/2017/9234905.
- [140] Bradley J. Wall and Bruce A. Conway. Shape-based approach to low-thrust rendezvous trajectory design. *Journal of Guidance, Control, and Dynamics*, 32(1):95–101, 2009. doi:10.2514/1.36848.
- [141] Anastassios E. Petropoulos and James M. Longuski. Shape-based algorithm

- for the automated design of low-thrust, gravity-assist trajectories. *Journal of Spacecraft and Rockets*, 41(5):787–796, 2004. doi:10.2514/1.13095.
- [142] Dario Izzo. Lambert’s problem for exponential sinusoids. *Journal of Guidance, Control, and Dynamics*, 29(5):1242–1245, 2006. doi:10.2514/1.21796.
- [143] Daniel M Novak and Massimiliano Vasile. Improved shaping approach to the preliminary design of low-thrust trajectories. *Journal of Guidance, Control, and Dynamics*, 34(1):128–147, 2011. doi:10.2514/1.50434.
- [144] Javier Roa, Jesús Peláez, and Juan Senent. Spiral lambert’s problem. *Journal of Guidance, Control, and Dynamics*, 39(10):2250–2263, 2016. doi:10.2514/1.G000342.
- [145] Paolo De Pascale and Massimiliano Vasile. Preliminary design of low-thrust multiple gravity-assist trajectories. *Journal of Spacecraft and Rockets*, 43(5):1065–1076, 2006. doi:10.2514/1.19646.
- [146] Anastassios E. Petropoulos and Jon A. Sims. A review of some exact solutions to the planar equations of motion of a thrusting spacecraft. In *2nd International Symposium Low Thrust Trajectories*, Toulouse, France, June 2002.
- [147] C. A. Balafoutis. A survey of efficient computational methods for manipulator inverse dynamics. *Journal of Intelligent and Robotic Systems*, 9(1):45–71, February 1994. doi:10.1007/BF01258313.
- [148] George Boyarko, Marcello Romano, and Oleg Yakimenko. Time-optimal reorientation of a spacecraft using an inverse dynamics optimization method. *Journal of Guidance, Control, and Dynamics*, 34(4):1197–1208, July-Aug 2011. doi:10.2514/1.49449.

- [149] Ping Lu. Inverse dynamics approach to trajectory optimization for an aerospace plane. *Journal of Guidance, Control, and Dynamics*, 16(4):726–732, 1993. doi:10.2514/3.21073.
- [150] Oleg A. Yakimenko. Direct method for rapid prototyping of near-optimal aircraft trajectories. *Journal of Guidance, Control, and Dynamics*, 23(5):865–875, Sept-Oct 2000. doi:10.2514/2.4616.
- [151] Richard H. Battin. Lambert’s problem revisited. *AIAA Journal*, 15(5):707–713, 1977. doi:10.2514/3.60680.
- [152] R. H. Gooding. A procedure for the solution of lambert’s orbital boundary-value problem. *Celestial Mechanics and Dynamical Astronomy*, 48(2):145–165, 1990. doi:10.1007/BF00049511.
- [153] Dario Izzo. Revisiting lambert’s problem. *Celestial Mechanics and Dynamical Astronomy*, 121(1):1–15, 2015. doi:10.1007/s10569-014-9587-y.
- [154] Nitin Arora and Ryan Paul Russell. A fast and robust multiple revolution lambert algorithm using a cosine transformation. In *AAS/AIAA Astrodynamics Specialist Conference, AAS 13-728*, Springfield, VA, 2013.
- [155] David Ottesen and Ryan Paul Russell. Space trajectory optimization using embedded boundary value problems. In *AAS/AIAA Astrodynamics Specialist Conference, AAS 17-837*, volume 162, pages 2271–2290, Stevenson, WA, August 2017.
- [156] David Ottesen and Ryan Paul Russell. A piecewise-constant sundman transformation for spacecraft trajectory optimization. In *AAS/AIAA Astrodynamics*

- Specialist Conference, AAS 18-467*, volume 167, pages 1755–1774, Snowbird, WA, August 2018.
- [157] Noble Hatten and Ryan Paul Russell. Application of the stark problem to space trajectories with time-varying perturbations. In *AAS/AIAA Astrodynamics Specialist Conference, AAS 14-230*, volume 152, pages 439–457, Santa Fe, NM, January 2014.
- [158] Jean Pierre Marec. *Optimal Space Trajectories*, volume 1. Elsevier, 1979.
- [159] P E Gill, W. Murray, and M. A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *Society for Industrial and Applied Mathematics*, 47(1):99–132, March 2005. doi:10.1137/S0036144504446096.
- [160] Charles George Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970. doi:10.1093/imamat/6.1.76.
- [161] Roger Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317–322, 1970. doi:10.1093/comjnl/13.3.317.
- [162] Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109):23–26, 1970. doi:10.1090/S0025-5718-1970-0258249-6.
- [163] David F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970. doi:10.1090/S0025-5718-1970-0274029-X.
- [164] William W. Hager and Hongchao Zhang. A new conjugate gradient method

- with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*, 16(1):170–192, 2005. doi:10.1137/030601880.
- [165] William W. Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*, 2(1):35–58, 2006.
- [166] Richard H Byrd, Jorge Nocedal, and Robert B Schnabel. Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1-3):129–156, 1994. doi:10.1007/BF01582063.
- [167] Reeves Fletcher and Colin M. Reeves. Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154, 1964. doi:10.1093/comjnl/7.2.149.
- [168] Elijah Polak and Gerard Ribiere. Note on convergence of conjugate direction meters. *ESAIM: Mathematical Modeling and Numerical Analysis-Mathematical Mathematics and Numerical Analysis*, 3(R1):35–43, 1969.
- [169] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Basic linear algebra subprograms for fortran usage. *ACM Transactions on Mathematical Software*, 5(3):308–323, Sep 1979. doi:10.1145/355841.355847.
- [170] Leonardo Dagum and Ramesh Menon. Openmp: an industry standard api for shared-memory programming. *Computational Science & Engineering, IEEE*, 5(1):46–55, 1998.
- [171] Anastassios E. Petropoulos and James M. Longuski. Automated design of low-thrust gravity-assist trajectories. In *AIAA*, page 4033, Aug 2000. doi:10.2514/6.2000-4033.
- [172] Bruce A. Conway. *Spacecraft Trajectory Optimization*. Cambridge Aerospace Series. Cambridge University Press, 2010.

- [173] David Ottesen and Ryan Paul Russell. Unconstrained direct optimization of spacecraft trajectories using many embedded lambert problems. *Journal of Optimization Theory and Applications*, pages 1–41, 2021. doi:10.1007/s10957-021-01884-1.
- [174] Philip H. Cowell and A. D. Crommelin. The orbit of jupiter’s eighth satellite. *Monthly Notices of the Royal Astronomical Society*, 68(8):576–582, 1908. doi:10.1093/mnras/68.8.576.
- [175] Karl F. Sundman. Memoire sur le probleme des trois corps. *Acta Mathematica*, 36:105–179, 1913. doi:10.1007/BF02422379.
- [176] Javier Roa. *Regularization in Orbital Mechanics*, volume 42. De Gruyter, 2017. doi:10.1515/9783110559125.
- [177] Tullio Levi-Civita. Sur la résolution qualitative du problème restreint des trois corps. *Acta Mathematica*, 30:305–327, 1906. doi:10.1007/BF02418577.
- [178] Joachim Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 1(1):1–16, 1972. doi:10.1016/0045-7825(72)90018-7.
- [179] Joachim Baumgarte and E. Stiefel. Examples of transformations improving the numerical accuracy of the integration of differential equations. In *Proceedings of the Conference on the Numerical Solution of Ordinary Differential Equations*, volume 362, pages 207–236, Berlin, Heidelberg, 1974. Springer. doi:10.1007/BFb0066593.
- [180] Paul Nacozy. The intermediate anomaly. *Celestial Mechanics and Dynamical Astronomy*, 16(3):309–313, 1977. doi:10.1007/BF01232657.

- [181] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *106(1):25–57*, 2006. doi:10.1007/s10107-004-0559-y.
- [182] Giulio Baù, Hodei Urrutxua, and Jesús Peláez. Edromo: An accurate propagator for elliptical orbits in the perturbed two-body problem. In *AAS/AIAA Spaceflight Mechanics Meeting, AAS 14-227*, volume 152, pages 379–399, Santa Fe, New Mexico, January 2014.
- [183] Giulio Baù, Claudio Bombardelli, Jesús Peláez, and Enrico Lorenzini. Non-singular orbital elements for special perturbations in the two-body problem. *Monthly Notices of the Royal Astronomical Society*, 454(3):2890–2908, 2015. doi:10.1093/mnras/stv2106.
- [184] Giulio Baù, Andrea Milani, Davide Amato, and Claudio Bombardelli. New orbital elements for accurate propagation in the solar system. In *6th International Conference on Astrodynamics Tools and Techniques, ICATT*, Darmstadt, March 2016.
- [185] Javier Roa and Jesús Peláez. Orbit propagation in minkowskian geometry. *Celestial Mechanics and Dynamical Astronomy*, 123(1):13–43, 2015. doi:10.1007/s10569-015-9627-2.
- [186] Giulio Baù and Javier Roa. Uniform formulation for orbit computation: The intermediate elements. *Celestial Mechanics and Dynamical Astronomy*, 132(2):1–31, 2020. doi:10.1007/s10569-020-9952-y.
- [187] Hermann G. Burchard. Splines (with optimal knots) are better. *Applicable Analysis*, 3(4):309–319, 1974. doi:10.1080/00036817408839073.

- [188] Joe F. Thompson. A survey of dynamically-adaptive grids in the numerical solution of partial differential equations. In *AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference*, AIAA 84-1606, pages 1–15, Snowmass, CO, U.S.A., June 1984. doi:10.2514/6.1984-1606.
- [189] Joe F. Thompson, Zahir U. A. Warsi, and C. Wayne Mastin. *Numerical Grid Generation: Foundations and Applications*, volume 45. Elsevier North-Holland, Inc., 1985.
- [190] Joe F. Thompson, Bharat K Soni, and Nigel P Weatherill. *Handbook of Grid Generation*. CRC press, 1998.
- [191] I. Babuska and W. C. Rheinboldt. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615, 1978. doi:10.1002/nme.1620121010.
- [192] I. Babuska and W. C Rheinboldt. Analysis of optimal finite-element meshes in r^1 . *Mathematics of Computation*, 33(146):435–463, 1979. doi:10.2307/2006290.
- [193] I. Babuska and W. Gui. Basic principles of feedback and adaptive approaches in the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 55(1–2):27–42, April 1986. doi:10.1016/0045-7825(86)90084-8.
- [194] J. Hugger. The theory of density representation of finite element meshes. examples of density operators with quadrilateral elements in the mapped domain. *Computer Methods in Applied Mechanics and Engineering*, 109(1-2):17–39, 1993. doi:10.1016/0045-7825(93)90223-K.

- [195] Timothy J. Baker. Mesh generation: Art or science? *Progress in Aerospace Sciences*, 41(1):29–63, 2005. doi:10.1016/j.paerosci.2005.02.002.
- [196] Weizhang Huang and Robert D. Russell. *Adaptive Moving Mesh Methods*, volume 4 of *174*. Springer-Verlag New York, 1 edition, 1993. doi:10.1007/978-1-4419-7916-2.
- [197] William F. Mitchell and Marjorie A. McClain. A comparison of hp-adaptive strategies for elliptic partial differential equations. *ACM Transactions on Mathematical Software*, 41(1):2, 2014. doi:10.1145/2629459.
- [198] John T. Betts, S. L. Campbell, and N. N. Kalla. Initialization of direct transcription optimal control software. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 3802–3807, Maui, HI, December 2003. doi:10.1109/CDC.2003.1271741.
- [199] Yiming Zhao and Panagiotis Tsiotras. Density functions for mesh refinement in numerical optimal control. *Journal of Guidance, Control, and Dynamics*, 34(1):271–277, Jan-Feb 2011. doi:10.2514/1.45852.
- [200] Yiming Zhao and Panagiotis Tsiotras. Analysis of energy-optimal aircraft landing operation trajectories. *Journal of Guidance, Control, and Dynamics*, 36(3):833–845, 2013. doi:10.2514/1.57779.
- [201] Christopher L. Darby, William W. Hager, and Anil V. Rao. Direct trajectory optimization using a variable low-order adaptive pseudospectral method. *Journal of Spacecraft and Rockets*, 48(3):433–445, May-June 2011. doi:10.2514/1.52136.
- [202] Christopher L. Darby, William W. Hager, and Anil V. Rao. An hp-adaptive

- pseudospectral method for solving optimal control problems. *Optimal Control Applications and Methods*, 32:476–502, May-June 2011. doi:10.1002/oca.957.
- [203] Hongfu Liu, Shaofei Chen, Lincheng Shen, and Jing Chen. An integrated multicriterion hp-adaptive pseudospectral method for direct optimal control problems solving. *Mathematical Problems in Engineering*, 2012, 2012. doi:10.1155/2012/760890.
- [204] Haijun Peng, Qiang Gao, Zhigang Wu, and Wanxie Zhong. Symplectic algorithms with mesh refinement for a hypersensitive optimal control problem. *International Journal of Computer Mathematics*, 92(11):2273–2289, 2015. doi:10.1080/00207160.2014.979810.
- [205] David G. Hull. *Optimal Control Theory for Applications*. Springer-Verlag, New York, NY, 2003.
- [206] Guy Janin and Victor R Bond. The elliptic anomaly. Technical report, NASA Technical Memorandum 58228, Lyndon B. Johnson Space Center, Houston, Texas, 77058, 1980.
- [207] Etienne Pellegrini, Ryan Paul Russell, and Vivek Vittaldev. F and G taylor series solutions to the stark and kepler problems with sundman transformations. *Celestial Mechanics and Dynamical Astronomy*, 118(4):355–378, 2014. doi:10.1007/s10569-014-9538-7.
- [208] Peter J. Prince and John R. Dormand. High order embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 7(1):67–75, 1981. doi:10.1016/0771-050X(81)90010-3.

- [209] Philip E Gill, Walter Murray, and Margaret H Wright. Practical optimization, 2019. doi:10.1137/1.9781611975604.
- [210] Troy T. McConaghy, Theresa J. Debban, Anastassios E. Petropoulos, and James M. Longuski. Design and optimization of low-thrust trajectories with gravity assists. *Journal of Spacecraft and Rockets*, 40(3):380–387, 2003. doi:10.2514/2.3973.
- [211] Dario Izzo. PyGMO and PyKEP: Open source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization). In *5th International Conference on Astrodynamics Tools and Techniques*, ICATT, The Netherlands, May–June 2012.
- [212] Anastassios E. Petropoulos, Zahi B. Tarzi, Gregory Lantoine, Thierry Dargent, and Richard Epenoy. Techniques for designing many-revolution electric-propulsion trajectories. *Advances in the Astronautical Sciences*, 152(3):2367–2386, 2014.
- [213] David Jimenez-Lluva and Bart Root. Hybrid optimization of low-thrust many-revolution trajectories with coasting arcs and longitude targeting for propellant minimization. *Acta Astronautica*, 177:232–245, 2020. doi:10.1016/j.actaastro.2020.06.015.
- [214] Craig A. Kluever and Steven R. Oleson. Direct approach for computing near-optimal low-thrust earth-orbit transfers. *Journal of Spacecraft and Rockets*, 35(4):509–515, 1998. doi:10.2514/2.3360.
- [215] John L. Junkins and Ehsan Taheri. Exploration of alternative state vector choices for low-thrust trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 42(1):47–64, 2019. doi:10.2514/1.G003686.

- [216] Di Wu, Wei Wang, Fanghua Jiang, and Junfeng Li. Minimum-time low-thrust many-revolution geocentric trajectories with analytical costates initialization. *Aerospace Science and Technology*, 119:107146, 2021. doi:10.1016/j.ast.2021.107146.
- [217] Ricardo L. Restrepo and Ryan Paul Russell. Shadow trajectory model for fast low-thrust indirect optimization. *Journal of Spacecraft and Rockets*, 54(1):44–54, 2017. doi:10.2514/1.A33611.
- [218] Jacob A. Englander, Bruce A. Conway, and Trevor Williams. Automated mission planning via evolutionary algorithms. *Journal of Guidance, Control, and Dynamics*, 35(6):1878–1887, 2012. doi:10.2514/1.54101.
- [219] Jacob A. Englander, Bruce A. Conway, and Trevor Williams. Automated interplanetary trajectory planning. In *AIAA/AAS Astrodynamics Specialist Conference, AIAA 12-4517*, Minneapolis, Minnesota, August 2012. doi:10.2514/6.2012-4517.
- [220] Jacob A. Englander, Donald H. Ellison, and Bruce A. Conway. Global optimization of low-thrust, multiple-flyby trajectories at medium and medium-high fidelity. In *AIAA/AAS Space-Flight Mechanics Meeting, AAS 14-309*, pages 1539–1558, Santa Fe, New Mexico, January 2014.
- [221] Jacob Williams, Juan S. Senent, Cesar Ocampo, Ravi Mathur, and Elizabeth C. Davis. Overview and software architecture of the copernicus trajectory design and optimization system. In *4th International Conference on Astrodynamics Tools and Techniques*, Madrid, Spain, 2010.
- [222] Gregory Whiffen. Mystic: Implementation of the static dynamic opti-

- mal control algorithm for high-fidelity, low-thrust trajectory design. 2006. doi:10.2514/6.2006-6741.
- [223] David Ottesen and Ryan Paul Russell. Piecewise sundman transformation for spacecraft trajectory optimization using many embedded lambert problems. *Journal of Spacecraft and Rockets*, pages 1–18, 2022. doi:10.2514/1.A35140.
- [224] John E. Prussing. Illustration of the primer vector in time-fixed orbit transfer. *AIAA Journal*, 7(6):1167–1168, 1969. doi:10.2514/3.5297.
- [225] Lin Hou-yuan and Zhao Chang-yin. Optimization of low-thrust trajectories using an indirect shooting method without guesses of initial costates. *Chinese Astronomy and Astrophysics*, 36(4):389–398, 2012. doi:10.1016/j.chinastron.2012.10.004.
- [226] Paul J. Enright and Bruce A. Conway. Discrete approximations to optimal trajectories using direct transcription and nonlinear programming. *Journal of Guidance, Control, and Dynamics*, pages 994–1002, 1992. doi:10.2514/3.20934.
- [227] Fariba Fahroo and I. Michael Ross. Costate estimation by a legendre pseudospectral method. *Journal of Guidance, Control, and Dynamics*, 24(2):270–277, 2001. doi:10.2514/2.4709.
- [228] William W. Hager. Runge-kutta methods in optimal control and the transformed adjoint system. *Numerische Mathematik*, 87(2):247–282, 2000. doi:10.1007/s002110000178.
- [229] David A. Benson, Geoffrey T. Huntington, Tom P. Thorvaldsen, and Anil V. Rao. Direct trajectory optimization and costate estimation via an orthog-

- onal collocation method. *Journal of Guidance, Control, and Dynamics*, 29(6):1435–1440, 2006. doi:10.2514/1.20478.
- [230] Qi Gong, I. Michael Ross, Wei Kang, and Fariba Fahroo. Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control. *Computational Optimization and Applications*, 41(3):307–335, 2008. doi:10.1007/s10589-007-9102-4.
- [231] Chit Hong Yam and James M. Longuski. Reduced parameterization for optimization of low-thrust gravity-assist trajectories: Case studies. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, 2006-6744*, pages 1–16, Keystone, Colorado, August 2006. doi:10.2514/6.2006-6744.
- [232] Donghun Lee and Hyochoong Bang. Efficient initial costates estimation for optimal spiral orbit transfer trajectories design. *Journal of Guidance, Control, and Dynamics*, 32(6):1943–1947, 2009. doi:10.2514/1.44550.
- [233] Christopher L Ranieri and Cesar A Ocampo. Indirect optimization of spiral trajectories. *Journal of Guidance, Control, and Dynamics*, 29(6):1360–1366, 2006. doi:10.2514/1.19539.
- [234] Praveen Jawaharlal Ayyanathan and Ehsan Taheri. Mapped adjoint control transformation method for low-thrust trajectory design. *Acta Astronautica*, 193:418–431, 2022. doi:10.1016/j.actaastro.2021.12.019.
- [235] James D. Thorne and Christopher D. Hall. Minimum-time continuous-thrust orbit transfers using the kustaanheimo-stiefel transformation. *Journal of guidance, control, and dynamics*, 20(4):836–838, 1997. doi:10.2514/2.4125.
- [236] Max Cerf. Fast solution of minimum-time low-thrust transfer with

- eclipses. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(7):2699–2714, 2019. doi:10.1177/0954410018785971.
- [237] Fanghua Jiang, Hexi Baoyin, and Junfeng Li. Practical techniques for low-thrust trajectory optimization with homotopic approach. *Journal of guidance, control, and dynamics*, 35(1):245–258, 2012. doi:10.2514/1.52476.
- [238] ShuGe Zhao and JingRui Zhang. Minimum-fuel station-change for geostationary satellites using low-thrust considering perturbations. *Acta Astronautica*, 127:296–307, 2016. doi:10.1016/j.actaastro.2016.05.028.
- [239] Thomas Haberkorn and Emmanuel Trélat. Convergence results for smooth regularizations of hybrid nonlinear optimal control problems. *SIAM Journal on Control and Optimization*, 49(4):1498–1522, 2011. doi:10.1137/100809209.
- [240] Ehsan Taheri, John L. Junkins, Ilya Kolmanovsky, and Anouck Girard. A novel approach for optimal trajectory design with multiple operation modes of propulsion system, part 1. *Acta Astronautica*, 2020. doi:10.1016/j.actaastro.2020.02.042.
- [241] Di Wu, Lin Cheng, Fanghua Jiang, and Junfeng Li. Rapid generation of low-thrust many-revolution earth-center trajectories based on analytical state-based control. *Acta Astronautica*, 187:338–347, 2021. doi:10.1016/j.actaastro.2021.05.017.
- [242] Lev Semenovich Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. The mathematical theory of optimal processes. In L W Neustadt, editor, *L.S. Pontryagin Selected Works*, volume 4. Gordon and Breach Science Publishers, 1986. doi:10.1201/9780203749319.

- [243] Erwin Fehlberg. Classical fifth-, sixth-, seventh-, and eighth-order runge-kutta formulas with stepsize control. Technical report, Technical Report NASA TR R-287, George C. Marshall Space Flight Center, Huntsville, Alabama, October 1968. <https://ntrs.nasa.gov/citations/19680027281>.
- [244] S. A. Dadebo, K. B. McAuley, and P. J. McLellan. On the computation of optimal singular and bang–bang controls. *Optimal Control Applications and Methods*, 19(4):287–297, 1998. doi:10.1002/(SICI)1099-1514(199807/08)19:4<287::AID-OCA628>3.0.CO;2-S.
- [245] J. B. Caillau, Bilel Daoud, and Joseph Gergaud. Minimum fuel control of the planar circular restricted three-body problem. *Celestial Mechanics and Dynamical Astronomy*, 114(1):137–150, 2012. doi:10.1007/s10569-012-9443-x.
- [246] Max Cerf, Thomas Haberkorn, and Emmanuel Trélat. Continuation from a flat to a round earth model in the coplanar orbit transfer problem. *Optimal Control Applications and Methods*, 33(6):654–675, 2012. doi:10.1002/oca.1016.
- [247] Sven Schäff. Low-thrust multi-revolution orbit transfers. In *Space Engineering*, pages 337–367. Springer, 2016. doi:10.1007/978-3-319-41508-6_13.
- [248] Liam M. Healy. Lambert targeting for on-orbit delivery of debris remediation dust. *Journal of Spacecraft and Rockets*, 51(5):1485–1491, 2014. doi:10.2514/1.A32634.
- [249] Giacomo Borelli, G. Gaias, C. Colombo, and L. Vallini. Rendezvous and proximity operations design of an active debris removal service to a large constellation fleet. In *72nd International Astronautical Congress (IAC 2021)*, pages 1–13, 2021.

- [250] Prado, A. F. B. A. Bi-impulsive control to build a satellite constellation. *Nonlinear Dyn. Syst. Theory*, 5(2):169–175, 2005.
- [251] Mahdi Fakoor, Majid Bakhtiari, and Mahshid Soleymani. Optimal design of the satellite constellation arrangement reconfiguration process. *Advances in Space Research*, 58(3):372–386, 2016. doi:10.1016/j.asr.2016.04.031.
- [252] Haijun Shen and Panagiotis Tsiotras. Optimal two-impulse rendezvous using multiple-revolution lambert solutions. *Journal of Guidance, Control, and Dynamics*, 26(1):50–61, 2003. doi:10.2514/2.5014.
- [253] Gang Zhang, Di Zhou, and Daniele Mortari. Optimal two-impulse rendezvous using constrained multiple-revolution lambert solutions. *Celestial Mechanics and Dynamical Astronomy*, 110(4):305–317, 2011. doi:10.1007/s10569-011-9349-z.
- [254] K Daneshjou, AA Mohammadi-Dehabadi, and Majid Bakhtiari. Mission planning for on-orbit servicing through multiple servicing satellites: A new approach. *Advances in Space Research*, 60(6):1148–1162, 2017. doi:10.1016/j.asr.2017.05.037.
- [255] R. H. Gooding. A new procedure for the solution of the classical problem of minimal orbit determination from three lines of sight. *Celestial Mechanics and Dynamical Astronomy*, 66(4):387–423, 1996. doi:10.1007/BF00049379.
- [256] Hideaki Hinagawa, Hitoshi Yamaoka, and Toshiya Hanada. Orbit determination by genetic algorithm and application to geo observation. *Advances in Space Research*, 53(3):532–542, 2014. doi:10.1016/j.asr.2013.11.051.
- [257] Etienne Pellegrini and Ryan Paul Russell. On the computation and accuracy

- of trajectory state transition matrices. *Journal of Guidance, Control, and Dynamics*, 39(11):2485–2499, 2016. doi:10.2514/1.G001920.
- [258] Stanley W. Shepperd. Universal keplerian state transition matrix. *Celestial Mechanics*, 35(2):129–144, February 1985. doi:10.1007/BF01227666.
- [259] W. H. Goodyear. Completely general closed-form solution for coordinates and partial derivative of the two-body problem. *Astronomical Journal*, 70:189, April 1965. doi:10.1086/109713.
- [260] E. T. Pitkin. Second transition partial derivatives via universal variables. *Journal of the Astronautical Sciences*, 13:204–207, 1966. <http://adsabs.harvard.edu/abs/1966JAnSc..13..204P>.
- [261] D. R. Glandorf. Lagrange multipliers and the state transition matrix for coasting arcs. *AIAA Journal*, 7(2):363–365, 1969. doi:10.2514/3.5109.

Vita

David Ottesen was born in Boise, Idaho, but has spent most of life in Texas. He grew up in Katy, attending elementary, middle, and high school there. He moved to Austin in 2005 to attend The University of Texas at Austin. He received his Bachelor of Science in 2009 and his Master of Science in 2012, both in aerospace engineering. After finishing his Master's work with Dr. Belinda Marchand, he joined Dr. Ryan P. Russell's research group for his Ph.D. Dr. Russell and him have been fortunate to work on many interesting optimization problems together. He has also been provided the opportunity to work at Johns Hopkins University Applied Physics Laboratory, Emergent Space Technologies, and NASA Ames Research Center throughout his graduate studies. Most importantly, his time as a teacher to hundreds of undergraduates will be remembered fondly.

Permanent Address: Austin, TX | davidottesen@gmail.com

This dissertation was typeset with $\text{\LaTeX} 2_{\epsilon}$ ¹ by the author.

¹ $\text{\LaTeX} 2_{\epsilon}$ is an extension of \LaTeX . \LaTeX is a collection of macros for \TeX . \TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.