UNIVERSITY OF CALIFORNIA SAN DIEGO

Symplectic Numerical Integration at the service of Accelerated Optimization and Structure-Preserving Dynamics Learning

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Mathematics with a Specialization in Computational Science

by

Valentin Duruisseaux

Committee in charge:

Professor Melvin Leok, Chair
Professor Nikolay A. Atanasov
Professor Jorge Cortés
Professor Philip E. Gill
Professor Michael J. Holst

2023

The dissertation of Valentin Duruisseaux is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

DEDICATION

Á ma sœur Roxane.

EPIGRAPH

La meccanica è il paradiso delle scienze matematiche,
perché con quella si viene al frutto matematico.
Mechanics is the paradise of the mathematical sciences,
because by means of it one comes to the fruits of mathematics.
--- Leonardo da Vinci

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my advisor, Professor Melvin Leok, for his invaluable advice, continuous support and guidance during my doctorate. He has been instrumental in my development as an independent researcher, and I could not have asked for a better and more suitable research advisor for my doctorate, and I will always feel very indebted for his generous mentorship.

I am also thankful to the members of my dissertation committee, Professor Nikolay Atanasov, Professor Jorge Cortés, Professor Philip Gill, and Professor Michael Holst, for providing constructive feedback and insightful suggestions to improve the quality of my research. In particular, I would like to thank Professor Michael Holst for helping me navigate the graduate program in the earlier stages of my doctorate, and Professor Nikolay Atanasov for his eagerness to work on joint projects with me and for setting up a research collaboration with his graduate student Thai Duong.

Then, I would like to express my gratitude to the professors who honored me with their generous guidance during my studies at McGill University and at the University of Cambridge: Professor Gantumur Tsogtgerel for introducing me to the world of Geometric Numerical Integration, Professor Antony Humphries for entrusting me with the first research project I ever worked on, and Professor Arieh Iserles for his inspiring words and spirit while supervising my dissertation on highly oscillatory quadrature. I would also like to thank Professor Axel Hundemer for providing me with my first opportunity to tutor and mentor undergraduate students at McGill University.

I would also like to thank all the instructors of the courses I have taken at McGill University, at the University of Cambridge, and at UC San Diego. The research presented in this dissertation was empowered by the material I learnt there. I am thinking in particular of Professor Guillaume Gervais, Professor Arieh Iserles, Professor Vojkan Jaksic, Professor Prakash Panangaden, Professor Charles Roth, Professor Gantumur Tsogtgerel, and Professor David Wolfson, who have been excellent and inspiring instructors.

I would also like to express my sincere gratitude to Joshua W. Burby and Qi Tang who have been supervising me during my research internship within the Applied Mathematics and Plasma Physics group at Los Alamos National Laboratory. This very enriching opportunity for me to explore conducting research at a national laboratory was made possible thanks to Joshua's openness and enthusiasm to collaborate with me. I am very grateful for the continued support and friendliness Joshua has shown me throughout the internship. I am also very grateful for the time and effort Qi has invested in me on a daily basis by sharing his knowledge, experience, and wisdom. I have developed and refined many new computational skills thanks to his mentorship, and also gained very valuable insight through his generous advice into existing career opportunities as a research scientist. I feel very fortunate to have had such a dedicated and caring supervisor and friend during my internship, and will always feel indebted for his guidance.

Additionally, I would like to express my appreciation to Amit Chakraborty for supervising and mentoring me during my research internship at Siemens thanks to whom I have acquired new skills and knowledge by considering different types of problems, and enhanced my understanding of research in a more industrial setting. I would also like to thank members of the Physics-Informed AI research group at Siemens, Olympia Brikis, Biswadip Dey, Shinjan Ghosh, Tongtao Zhang, and Yaofeng Desmond Zhong, for welcoming me into their team and for enlightening conversations.

Next, I would like to express my deepest heartfelt gratitude to all the friends I was very fortunate to make along the way in Canada, in the United Kingdom, and in the United States, who have made this journey considerably easier, more pleasant and memorable with their company and support. I am thinking in particular (in alphabetical order) of Philip Amortilla, Diana Arévalo, Alp Aydınoğlu, Francis Aznaran, Étienne Bilocq, Etienne Bolduc, Juan Cerviño, Efstathios–Konstantinos Chrontsios–Garitsis, Serte Donderwinkel, Thai Duong, Woojing Kyle Jang, Rusiru Gambheera, Soumya Ganguly, Rahul Garg, Pamela Garza–Báez, Tanguy Grall, Joseph Groga, Andrea Helfant, Tudor Manole, Stefano Misino, Alec Poulin, Andrés Rodriguez Rey, Magid Sabbagh, Christian Seitz, Gilles Seropian, Rajat Sethi, Deepthi Sunny, and Ji Zeng.

I would like to express my deepest gratitude to all my family who has always been unconditionally supportive of my endeavors. In particular, I would like to express my gratitude to my mother Sandrine because it was surely not easy to let me move so far away so young, and to my father Philippe who has been discreetly archiving all my publications and probably has made his own version of my doctoral dissertation by now. I would also like to express my heartfelt gratefulness to my grandmother Nicole who has always been very kind and caring, but also too curious for her own good and now regularly has nightmares with Schrödinger's cat, Chaos theory, and Fermat's last theorem. Additionally, I wish to express my sincere appreciation to my grandfathers Asgar and Karim who have always been very generous with their love and wisdom. Last but not least, I am extremely grateful to my sister and perpetual partner in crime, Roxane, to whom I am honored to dedicate this dissertation.

Finally, I would like to gratefully acknowledge the sources of funding which have supported the research presented in this dissertation. Specifically,

- the National Science Foundation (NSF) under grants CCF-2112665, DMS-1411792, DMS-1345013, DMS-1813635,
- the Air Force Office of Scientific Research (AFOSR) under grant FA9550-18-1-0288,
- the U.S. Department of Defense (DOD) under grants HQ00342010023 and 13106725 (Newton Award for Transformative Ideas during the COVID-19 Pandemic).
- the U.S. Department of Energy (DOE), the Office of Science and the Office of Advanced Scientific Computing Research (ASCR) under grant DOE-FOA-2493. Note that the research presented in Chapter 7 used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. DOE Office of Science user facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award ASCR-ERCAP0020162.

This dissertation contains original work extracted from publications as follows:

- Chapter 2 contains original material from

  ① "Adaptive Hamiltonian Variational Integrators and Applications to Symplectic Accelerated Optimization" by V. Duruisseaux, J. Schmitt, and M. Leok. *SIAM Journal on Scientific Computing,* Vol.43, No.4, A2949-A2980, 2021

  ② "Accelerated Optimization on Riemannian Manifolds via Discrete Constrained Variational Integrators" by V. Duruisseaux and M. Leok. *Journal of Nonlinear Science*, Vol.32, No.42, 2022

  ③ "Time-adaptive Lagrangian Variational Integrators for Accelerated Optimization on Manifolds" by V. Duruisseaux and M. Leok. *Journal of Geometric Mechanics*, Vol.15, Issue 1, pages 224-255, 2023.

  The dissertation author was the primary investigator and author of this paper.

- Chapter 3 contains original material from

  ① "Adaptive Hamiltonian Variational Integrators and Applications to Symplectic Accelerated Optimization" by V. Duruisseaux, J. Schmitt, and M. Leok. *SIAM Journal on Scientific Computing,* Vol.43, No.4, A2949-A2980, 2021

  ② "Time-adaptive Lagrangian Variational Integrators for Accelerated Optimization on Manifolds" by V. Duruisseaux and M. Leok. *Journal of Geometric Mechanics*, Vol.15, Issue 1, pages 224-255, 2023.

  The dissertation author was the primary investigator and author of these papers.

- Chapter 4 contains original material from

  ① "A Variational Formulation of Accelerated Optimization on Riemannian Manifolds" by V. Duruisseaux and M. Leok. *SIAM Journal on Mathematics of Data Science*, Vol.4, No.2, pages 649-674, 2022

  ② "Accelerated Optimization on Riemannian Manifolds via Discrete Constrained Variational Integrators" by V. Duruisseaux and M. Leok. *Journal of Nonlinear Science*, Vol.32, No.42, 2022

③ "Accelerated optimization on Riemannian manifolds via Projected Variational Integrators" by V. Duruisseaux and M. Leok, 2022

④ "Time-adaptive Lagrangian Variational Integrators for Accelerated Optimization on Manifolds" by V. Duruisseaux and M. Leok. *Journal of Geometric Mechanics*, Vol.15, Issue 1, pages 224-255, 2023.

The dissertation author was the primary investigator and author of these papers.

➢ Chapter 5 contains original material from

① "Practical Perspectives on Symplectic Accelerated Optimization" by V. Duruisseaux and M. Leok. *Optimization Methods and Software*, Vol.38, Issue 6, pages 1230-1268, 2023

The dissertation author was the primary investigator and author of this paper.

➢ Chapter 6 contains original material from

① "Lie Group Forced Variational Integrator Networks for Learning and Control of Robot Systems" by V. Duruisseaux, T. Duong, M. Leok, and N. Atanasov. *Proceedings of the 5th Learning for Dynamics and Control Conference (L4DC),* PMLR 211:731-744, 2023

The dissertation author and Thai Duong were the primary investigators and authors of this paper.

➢ Chapter 7 contains original material from

① "Approximation of Nearly-Periodic Symplectic Maps via Structure-Preserving Neural Networks" by V. Duruisseaux, J. W. Burby, and Q. Tang. *Scientific Reports,* Vol.13, No.8351, 2023

The dissertation author was the primary investigator and author of this paper.

➢ The Appendix contains original material from

① "Adaptive Hamiltonian Variational Integrators and Applications to Symplectic Accelerated Optimization" by V. Duruisseaux, J. Schmitt, and M. Leok. *SIAM Journal on Scientific Computing,* Vol.43, No.4, A2949-A2980, 2021

② "A Variational Formulation of Accelerated Optimization on Riemannian Manifolds" by V. Duruisseaux and M. Leok. *SIAM Journal on Mathematics of Data Science*, Vol.4, No.2, pages 649-674, 2022

③ "Accelerated Optimization on Riemannian Manifolds via Discrete Constrained Variational Integrators" by V. Duruisseaux and M. Leok. *Journal of Nonlinear Science*, Vol.32, No.42, 2022

④ "Time-adaptive Lagrangian Variational Integrators for Accelerated Optimization on Manifolds" by V. Duruisseaux and M. Leok. *Journal of Geometric Mechanics*, Vol.15, Issue 1, pages 224-255, 2023.

⑤ "Lie Group Forced Variational Integrator Networks for Learning and Control of Robot Systems" by V. Duruisseaux, T. Duong, M. Leok, and N. Atanasov. *Proceedings of the 5th Learning for Dynamics and Control Conference (L4DC),* PMLR 211:731-744, 2023

The dissertation author was the primary investigator and author of these papers.

VITA

2017	B.S. Joint Honours in Mathematics and Physics with *First Class Honours and Distinction* McGill University, Montréal, Canada
2018	Masters of Advanced Study in Mathematics, with *Merit* University of Cambridge, Cambridge, United Kingdom
2023	Ph.D. in Mathematics with a Specialization in Computational Science University of California San Diego, United States of America

PUBLICATIONS

Adaptive Hamiltonian Variational Integrators and Applications to Symplectic Accelerated Optimization
Valentin Duruisseaux, Jeremy Schmitt, and Melvin Leok
SIAM Journal on Scientific Computing, Vol.43, No.4, A2949-A2980, 2021.
[Duruisseaux et al., 2021], eprint: doi.org/10.1137/20M1383835

A Variational Formulation of Accelerated Optimization on Riemannian Manifolds
Valentin Duruisseaux and Melvin Leok
SIAM Journal on Mathematics of Data Science, Vol.4, No.2, pages 649-674, 2022.
[Duruisseaux and Leok, 2022d], eprint: doi.org/10.1137/21M1395648

Variational Accelerated Optimization on Riemannian Manifolds
Valentin Duruisseaux and Melvin Leok
IEICE Proceedings Series 71 (B1L-D-03), NOLTA International Symposium, 2022.
[Duruisseaux and Leok, 2022b] , eprint: doi.org/10.34385/proc.71.B1L-D-03

Accelerated Optimization on Riemannian Manifolds via Discrete Constrained Variational Integrators
Valentin Duruisseaux and Melvin Leok
Journal of Nonlinear Science, Vol.32, No.42, 2022.
[Duruisseaux and Leok, 2022a], eprint: doi.org/10.1007/s00332-022-09795-9

Accelerated Optimization on Riemannian Manifolds via Projected Variational Integrators
Valentin Duruisseaux and Melvin Leok, 2022.
[Duruisseaux and Leok, 2022c], eprint: arxiv.org/abs/2201.02904

Bistability, Bifurcations and Chaos in the Mackey-Glass Equation
Valentin Duruisseaux and Antony R. Humphries
Journal of Computational Dynamics, Vol.9, No.3, pages 421-450, 2022.
[Duruisseaux and Humphries, 2022], eprint: doi.org/10.3934/jcd.2022009

Practical Structured Riemannian Optimization with Momentum by using Generalized Normal Coordinates
Wu Lin, **Valentin Duruisseaux**, Melvin Leok, Frank Nielsen, Mohammad Emtiyaz Khan, Mark Schmidt
NeurIPS Workshop on Symmetry and Geometry in Neural Representations, 2022.
[Lin et al., 2022], eprint: openreview.net/forum?id=1aybhSfabqh

Time-adaptive Lagrangian Variational Integrators for Accelerated Optimization on Manifolds
Valentin Duruisseaux and Melvin Leok
Journal of Geometric Mechanics, Vol.15, Issue 1, pages 224-255, 2023.
[Duruisseaux and Leok, 2023a], eprint: doi.org/10.3934/jgm.2023010

Practical Perspectives on Symplectic Accelerated Optimization
Valentin Duruisseaux and Melvin Leok
Optimization Methods and Software, Vol.38, Issue 6, pages 1230-1268, 2023.
[Duruisseaux and Leok, 2023b], eprint: doi.org/10.1080/10556788.2023.2214837

Approximation of Nearly-Periodic Symplectic Maps via Structure-Preserving Neural Networks
Valentin Duruisseaux, Joshua W. Burby, and Qi Tang
Scientific Reports, Vol.13, No.8351, 2023.
[Duruisseaux et al., 2023a], eprint: doi.org/10.1038/s41598-023-34862-w

Lie Group Forced Variational Integrator Networks for Learning and Control of Robot Systems
Valentin Duruisseaux, Thai Duong, Melvin Leok, and Nikolay Atanasov
Proceedings of the 5th Learning for Dynamics and Control Conference (L4DC), PMLR 211:731-744, 2023.
[Duruisseaux et al., 2023c], eprint: proceedings.mlr.press/v211/duruisseaux23a.html

Simplifying Momentum-based Positive-definite Submanifold Optimization with Applications to Deep Learning
Wu Lin, **Valentin Duruisseaux**, Melvin Leok, Frank Nielsen, Mohammad Emtiyaz Khan, Mark Schmidt
Proceedings of the 40th International Conference on Machine Learning (ICML), PMLR 202:21026-21050, 2023.
[Lin et al., 2023], eprint: proceedings.mlr.press/v202/lin23c.html

An Operator Learning Framework for Spatiotemporal Super-resolution of Scientific Simulations
Valentin Duruisseaux and Amit Chakraborty, 2023.
[Duruisseaux and Chakraborty, 2023], eprint: arxiv.org/abs/2311.02328

ABSTRACT OF THE DISSERTATION

Symplectic Numerical Integration
at the service of
Accelerated Optimization and Structure-Preserving Dynamics Learning

by

Valentin Duruisseaux
Doctor of Philosophy in Mathematics with a Specialization in Computational Science
University of California San Diego, 2023
Professor Melvin Leok, Chair

Symplectic numerical integrators for Hamiltonian systems form the paramount class of geometric numerical integrators, and have been very well investigated in the past forty years. By preserving the symplecticity of the flow of Hamiltonian systems, symplectic integrators generate discrete solutions which enjoy many desirable qualitative properties such as excellent long-time near-energy preservation over exponentially-long time intervals, and as a result typically exhibit superior numerical stability and better long-time fidelity to the continuous dynamics they are resolving.

The purpose of this dissertation is to establish, explore and exploit connections between symplectic numerical integration and two prominent disciplines of scientific computing: accelerated optimization for machine learning applications, and structure-preserving dynamics learning. By leveraging the well-established theory of symplectic integration, we aspire to design new algorithms for accelerated optimization and dynamics learning with superior numerical and computational properties.

Most machine learning algorithms are designed around the minimization of a loss function or the maximization of a likelihood function. Due to the ever-growing size of the datasets, obtaining efficient optimization algorithms which can be executed on parallel and distributed processing architectures is of critical importance. Numerous accelerated optimization algorithms limit to differential equations as the step size goes to zero, and the objective function typically converges to its optimal value at an accelerated rate along the trajectories of these limiting differential equations. As a result, the optimization problem can be replaced by the problem of evolving dynamics governed by appropriately defined differential equations. In Part II, we construct novel optimization algorithms via symplectic integration of carefully designed Hamiltonian systems which converge to the minimizer at an accelerated rate, both on normed vector spaces and Riemannian manifolds.

Identifying accurate and efficient models for dynamical systems based on observed trajectories is crucial for predicting and controlling their future behavior. Incorporating the structure underlying the dynamics in deep learning architectures has proven to be an efficient approach to learning structured dynamical systems. In Part III, we design novel deep learning architectures incorporating the geometric structure of nearly-periodic Hamiltonian systems and controlled Lie group Hamiltonian systems to learn structure-preserving surrogate evolution maps.

Introduction

Geometric numerical integration has emerged in the 1990s as the field of mathematics concerned with the development of numerical methods to simulate the evolution of dynamical systems governed by differential equations while preserving the geometric attributes of these dynamical systems. These geometric attributes are typically invariants or constants of the flow arising from conservation laws and symmetries of the dynamical system. As a result, these geometric properties constrain the evolution of the dynamical system to a lower-dimensional manifold. Integrators which respect the geometric attributes of the dynamical system and its flow typically exhibit superior numerical stability, and better long-time fidelity to the continuous dynamics they are resolving.

We refer the reader to [Iserles and Quispel, 2018] for a brief recent overview of geometric numerical integration, and to [Hairer et al., 2006; Blanes and Casas, 2017] for a more comprehensive presentation of structure-preserving integration techniques.

There is an abundance of qualitatively distinct dynamical behaviours, each of which possesses its own collection of underlying geometric properties. As a result, it is necessary to develop and use geometric numerical methods tailored to the geometric structure of the dynamical system of interest. These could be symplectic integrators for conservative Hamiltonian systems, symmetric integrators for reversible systems, methods preserving specific conserved quantities (such as energy, volume, momenta, and flux), numerical methods specially designed for problems with periodic and highly oscillatory solutions, and numerical integrators evolving intrinsically on manifolds and Lie groups for constrained dynamical systems.

In this dissertation, we focus predominantly on Hamiltonian systems and their underlying symplectic structure: any solution to a Hamiltonian system is a symplectic flow and any symplectic flow corresponds to an appropriate Hamiltonian system locally. Symplectic numerical integrators form the class of geometric numerical integrators which yield discrete approximations of the flow that preserve the symplectic 2-form when applied to Hamiltonian systems. This preservation of the symplectic 2-form results in many desirable qualitative properties for the resulting discrete flow such as excellent long-time near-energy preservation over exponentially-long time intervals.

Symplectic integrators for Hamiltonian systems are perhaps the most prominent examples of geometric numerical integrators, and have been very thoroughly investigated in the past 40 years. As discussed in [Iserles and Quispel, 2018], the theory of symplectic integration (and more generally of geometric numerical integration) is very well established, and as a result of its success, all the major challenges in symplectic integration might have already been achieved. There are however numerous fields in science and engineering that could benefit from the well-established theory of geometric numerical integration. This dissertation aims to establish, explore and exploit connections between symplectic numerical integration and two prominent disciplines of scientific computing: accelerated optimization for machine learning applications, and structure-preserving dynamics learning.

Part I - Symplectic Numerical Integration

We begin by reviewing the theory of symplectic numerical integration and by establishing a few extensions which will prove very useful when considering applications of symplectic numerical integration to optimization and dynamics learning.

In Chapter 1, we introduce the main definitions and notions from differential and Riemannian geometry, and Lie group theory, that we will use extensively throughout the dissertation when considering dynamical systems which evolve on manifolds and Lie groups. We also introduce several important examples of Riemannian manifolds and Lie groups which will be considered in numerical experiments conducted in this dissertation.

Chapter 2 is devoted to the general theory of geometric numerical integration of Hamiltonian and Lagrangian systems. We first introduce Lagrangian and Hamiltonian mechanics (Section 2.1) and their underlying symplecticity (Section 2.2), and discuss the benefits of preserving symplecticity when numerically integrating these dynamical systems (Section 2.3), such as excellent long-time near-energy preservation. We then describe variational integrators (Section 2.4) which form a class of symplectic integrators, and discuss how external forcing and control can be included in these integrators (Section 2.5). Then, in Section 2.6, we investigate how holonomic constraints can be incorporated into variational integrators to constrain the numerical discretization of a continuous Lagrangian or Hamiltonian system to a certain constraint manifold, and in particular derive new results for the Hamiltonian formulation. Finally, in Section 2.7, we discuss how prescribed variable time-steps can be incorporated in symplectic integrators without losing all the desirable properties associated with these integrators.

Part II - Accelerated Optimization via Geometric Numerical Integration

Efficient optimization has become one of the major concerns in data analysis and artificial intelligence. Most machine learning algorithms require the minimization of a loss function or the maximization of a likelihood function. Due to the ever-growing scale of the data sets and size of the problems, there has been a lot of focus on first-order optimization algorithms because of their low cost per iteration and the ease with which they can be executed on parallel and distributed processing architectures. Standard gradient descent methods typically converge to the minimum of the convex objective function f at a $\mathcal{O}(1/k)$ rate, where k is the iteration number. Several optimization algorithms have however been shown to achieve convergence in $\mathcal{O}(1/k^2)$, which was proven to be optimal among first-order methods using only information about ∇f at consecutive iterates. This phenomenon in which an algorithm displays this improved rate of convergence is referred to as acceleration. It was shown recently that many accelerated optimization algorithms limit to differential equations as the step size goes to 0, and that the objective function converges to its optimal value at an accelerated rate in continuous time along the trajectories of

these differential equations. This effectively replaced the problem of minimizing the convex objective function by the problem of numerically evolving dynamical systems governed by appropriately defined differential equations. In [Wibisono et al., 2016], it is shown that accelerated convergence can be achieved by considering flow maps generated by a family of time-dependent Bregman Lagrangian and Hamiltonian systems.

In Chapter 3, we first review this variational framework introduced in [Wibisono et al., 2016] for accelerated optimization on normed vector spaces. We then introduce a time-adaptive approach to integrate the resulting time-dependent Hamiltonian and Lagrangian dynamical systems, by exploiting the frameworks for variable time-stepping in symplectic integrators and the time-rescaling property of the Bregman family of dynamics. This leads to efficient symplectic algorithms for accelerated optimization, which we carefully study to investigate how time-adaptivity and symplecticity affect their performance.

In Chapter 4, we generalize the accelerated optimization framework presented in Chapter 3 to the Riemannian manifold setting, and establish existence and convergence results for objective functions on Riemannian manifolds that are geodesically convex, weakly quasi-convex, and strongly convex in Section 4.2. The acceleration is achieved along solutions of the Euler–Lagrange and Hamilton's equations corresponding to a family of time-dependent Bregman Lagrangian and Hamiltonian systems on Riemannian manifolds. We show in Section 4.4 that this family of Bregman dynamics on Riemannian manifolds is closed under time rescaling, and take advantage of this invariance property via a Poincaré transformation that allows for the integration of higher-order dynamics with the computational efficiency of integrating lower-order dynamics. We introduce different approaches to design geometric integrators for the Riemannian Bregman family of dynamics. In the Hamiltonian setting (Section 4.5), we take advantage of embedding theorems to reduce Riemannian manifolds to submanifolds of Euclidean spaces, and exploit the structure of the embedding Euclidean spaces using constrained or projection-based variational integrators. In the Lagrangian setting (Section 4.6), we design time-adaptive Lagrangian variational integrators which evolve intrinsically on the Riemannian manifold.

In Chapter 5, we discuss practical considerations which can significantly boost the computational performance and ease the tuning of symplectic optimization algorithms that are constructed by integrating Bregman Lagrangian and Hamiltonian systems coming from the frameworks for accelerated optimization presented in Chapters 3 and 4. We investigate how momentum restarting ameliorates computational efficiency and robustness by reducing the undesirable effect of oscillations (Section 5.4), and ease the tuning process by making time-adaptivity superfluous (Section 5.5). We also discuss how temporal looping helps avoiding instability issues caused by numerical precision (Section 5.8). We then compare the efficiency and robustness of different geometric integrators (Section 5.6), and study the effects of various parameters in the algorithms to inform and simplify tuning in practice (Section 5.7). This computational study leads to symplectic accelerated optimization algorithms, the BrAVO algorithms, whose computational efficiency, stability and robustness have been improved, and which are now much simpler to use and tune for practical applications. The BrAVO algorithms are then tested in Section 5.9 on more challenging optimization problems arising from machine learning with a variety of model architectures, loss functions to minimize, and applications.

Part III - Structure–Preserving Dynamics Learning

Dynamical systems evolve according to the laws of physics, which can usually be described using differential equations. Identifying accurate and efficient dynamic models based on observed trajectories is thus critical not only for predicting future behavior, but also for designing control laws that ensure desirable properties such as safety, stability, and generalization to different operational conditions. We will consider the problem of learning dynamics: given a dataset of observed trajectories followed by a dynamical system, we wish to infer the dynamical law responsible for these trajectories and use it to predict the evolution of the system from different initial states. We are also interested in the surrogate modeling problem: the underlying dynamical system is known, but traditional simulations are either too slow or expensive for some optimization task. This problem can be addressed by learning a less expensive surrogate for the simulations.

Dynamical system models obtained from first principles are extensively used in practice, but tend to over-simplify or incorrectly describe the underlying structure of the systems, which usually leads to high prediction errors. Deep learning techniques provide very expressive models for function approximation, but standard neural networks struggle to learn the symmetries and conservation laws underlying dynamical systems, and as a result do not generalize well. They are typically over-parameterized, can be very difficult to interpret, and require large datasets and computational times which make them prohibitively expensive for many applications. A recent research direction has been considering a hybrid approach, where physics laws and geometric properties are encoded in the design of the deep learning architectures or in the learning process. Available physics prior knowledge can be used to construct physics-constrained neural networks with improved design and efficiency and a better generalization capacity, which can take advantage of the function approximation power of deep learning methods to deal with incomplete knowledge. An important example of geometric structure underlying dynamical systems is the symplecticity of the flows of Hamiltonian systems. It is important to have structure-preserving architectures which can learn symplectic maps and ensure that the learnt surrogates preserve symplecticity. Numerous physics-informed machine learning approaches have been proposed to learn Hamiltonian dynamics and symplectic maps.

In Chapter 6, we introduce a new structure-preserving deep learning architecture, the Lie group Forced Variational Integrator Network (LieFVIN), to learn surrogates for the flow maps of controlled Lagrangian or Hamiltonian dynamics evolving on Lie groups, either from position-velocity or position-only data. By design, LieFVINs preserve both the Lie group structure on which the dynamics evolve and the symplectic structure underlying the controlled Hamiltonian systems of interest. The proposed architecture learns surrogate discrete-time flow maps allowing accurate and fast prediction without using a numerical integrator, neural-ODE, or adjoint techniques, which are needed for vector fields. The learnt discrete-time dynamics can also be used with computationally scalable discrete-time (optimal) control strategies. Many mechanical and robotic systems can be modeled as (controlled) Hamiltonian or Lagrangian systems evolving on Lie groups, so the problem of learning this class of dynamical systems is of considerable importance.

In Chapter 7, we construct a novel structure-preserving neural network architecture, capable of approximating nearly-periodic symplectic maps. Nearly-periodic symplectic maps are the discrete-time analogues of nearly-periodic Hamiltonian systems, where a dynamical system with parameter ε is said to be nearly-periodic if all its trajectories are periodic with nowhere-vanishing angular frequency as ε approaches 0. Nearly-periodic symplectic maps possess approximately conserved discrete-time quantities, called adiabatic invariants, and our novel neural network architecture, which we call symplectic gyroceptron, ensures that the resulting surrogate map is nearly-periodic and symplectic, and that it gives rise to a discrete-time adiabatic invariant and a long-time stability as a consequence. Symplectic gyroceptrons provide a promising class of architectures for surrogate modeling of non-dissipative dynamical systems that automatically steps over short timescales without introducing spurious instabilities, and could have future applications for the Klein–Gordon equation in the weakly-relativistic regime, for charged particles moving through a strong magnetic field, and for the rotating inviscid Euler equations in quasi-geostrophic scaling. Symplectic gyroceptrons could also be used for structure-preserving simulation of non-canonical Hamiltonian systems on exact symplectic manifolds, which have numerous applications across the physical sciences.

Part I

Symplectic Numerical Integration

1 Preliminaries

1.1 Riemannian Manifolds

1.1.1 Differential Geometry

We first introduce some standard definitions from differential geometry that we will use throughout this dissertation (see [Lang, 1999; Marsden and Ratiu, 1999; McInerney, 2013] for more details).

Definition 1.1. *Given a set M, a **chart** on M is a pair (U, φ) where $U \subset M$ and φ is a bijective map from U to $\varphi(U) \subset \mathbb{R}^n$. Two charts (U_1, φ_1) and (U_2, φ_2) with $U_1 \cap U_2 \neq \varnothing$ are **compatible** if $\varphi_j(U_1 \cap U_2)$ is an open subset of $\mathbb{R}^n$ and*

$$\varphi_i \circ (\varphi_j)^{-1}\big|_{\varphi_j(U_1 \cap U_2)} \;:\; \varphi_j(U_1 \cap U_2) \to \varphi_i(U_1 \cap U_2) \tag{1.1}$$

*is smooth for $(i, j) \in \{(1,2),(2,1)\}$. A **smooth manifold** is a set M which can be written as a union of compatible charts. Given a chart (U, φ) on a n-dimensional manifold M, we get associated **local coordinates** $(x^1, \ldots, x^n)$ for points in U.*

Definition 1.2. *Two curves $s \mapsto c_1(s)$ and $s \mapsto c_2(s)$ on a manifold M are **equivalent** at $m \in M$ if*

$$c_1(0) = c_2(0) = m \qquad \text{and} \qquad (\varphi \circ c_1)'(0) = (\varphi \circ c_2)'(0) \tag{1.2}$$

*in some chart φ. A **tangent vector** to a manifold M at a point $m \in M$ is an equivalence class of curves at m. The set of tangent vectors to a manifold M at a point $m \in M$ is a vector space called the **tangent space** to M at m and is denoted by T_mM, and the disjoint union of all the tangent spaces to M forms the **tangent bundle** TM of M:*

$$TM = \{(m, v) \mid m \in M,\ v \in T_mM\}. \tag{1.3}$$

Coordinates $(x^1, \ldots, x^n)$ *induce a basis* $\partial_j = \frac{\partial}{\partial x^j}$ *for the tangent spaces, so that any tangent vector can be written in the form* $v = \sum_{j=1}^{n} v^j \partial_j$.

Definition 1.3. *The vector space dual to the tangent space* $T_m M$ *is the* ***cotangent space*** $T_m^* M$*, and the vector bundle over* M *whose fibers are the cotangent spaces of* M *is the* ***cotangent bundle*** T^*M*:*

$$T^*M = \{(m, p) \mid m \in M,\ p \in T_m^* M\}. \tag{1.4}$$

Elements of the tangent bundle T^*M *are called* ***cotangent vectors****. The coordinates* $(x^1, \ldots, x^n)$ *induce a dual basis* $dx^1, \ldots, dx^n$ *for the cotangent spaces by requiring that*

$$dx^j(\partial_k) \;=\; \partial_k(dx^j) \;=\; \delta_k^j \qquad \text{for all } j \text{ and } k, \tag{1.5}$$

where δ_k^j *is the Kronecker delta.*

Definition 1.4. *A* ***vector field*** *on a manifold* M *is a map* $X : M \to TM$ *such that* $X(m) \in T_m M$ *for all* $m \in M$*. The set of all vector fields on* M *is denoted* $\mathcal{X}(M)$*. In local coordinates, a vector field* X *can be represented as* $X(m) = \sum_{j=1}^{n} X^j(m)\partial_j$*. The* ***integral curve*** *at* m *of* $X \in \mathcal{X}(M)$ *is the smooth curve* c *on* M *such that* $c(0) = m$ *and* $c'(t) = X(c(t))$*. The* ***flow*** *of a vector field* $X \in \mathcal{X}(M)$ *is the collection of maps* $\varphi_t : M \to M$ *such that* $\varphi_t(m)$ *is the integral curve of* X *with initial condition* $m \in M$*.*

Definition 1.5. *A smooth function* $f : M \to \mathbb{R}$ *can be differentiated at any point* $m \in M$ *to obtain the* ***tangent map*** $\mathrm{T}_m f : T_m M \to T_{f(m)}\mathbb{R}$*. By identifying the tangent space to* $\mathbb{R}$ *at any point to* $\mathbb{R}$ *itself, we obtain a linear map* $\mathbf{d}f(m) \in T_m^* M$*, and we call* $\mathbf{d}f$ *the* ***differential*** *of* f*.*

Definition 1.6. *A* ***k-form*** *on a manifold* M *is a map which assigns to every point* $m \in M$ *a skew-symmetric* k*-multilinear map on* $T_m M$*. In other words,* α *is a* k*-form if for every point* $m \in M$*,* $\alpha(m) : T_m M \times \ldots \times T_m M \to \mathbb{R}$ *is linear in each of its* k *arguments and has the property that it changes sign whenever two of its arguments are interchanged. A general* k*-form can be written as*

$$\alpha_m(v_1, \ldots, v_k) \;=\; \sum_{i_1, \ldots, i_k} \alpha_{i_1 \ldots i_k}(m)\ v_1^{i_1} \ldots v_k^{i_k} \tag{1.6}$$

where $\alpha_{i_1 \ldots i_k}(m) = \alpha_m(\partial_{i_1}, \ldots, \partial_{i_k})$*, for any vectors* $v_i = \sum_j v_i^j \partial_j \in T_m M$ *for* $i = 1, \ldots, k$*.*

Definition 1.7. *Given a k-form α and a s-form β on a manifold M, we define their* ***tensor product*** *$\alpha \otimes \beta$ at any point $m \in M$ via*

$$(\alpha \otimes \beta)_m(v_1, \ldots, v_{k+s}) \;=\; \alpha_m(v_1, \ldots, v_k)\; \beta_m(v_{k+1}, \ldots, v_{k+s}). \tag{1.7}$$

Definition 1.8. *We define the* ***alternating operator*** Alt *acting on the k-form α via*

$$\mathrm{Alt}(\alpha)(v_1, \ldots, v_k) \;=\; \frac{1}{k!} \sum_{\pi \in S_k} \mathrm{sgn}(\pi)\; \alpha(v_{\pi(1)}, \ldots, v_{\pi(k)}) \tag{1.8}$$

where S_k is the group of all the permutations of $\{1, \ldots, k\}$ and $\mathrm{sgn}(\pi)$ is the sign of the permutation (+ if π even, -1 if π is odd).

Definition 1.9. *Given a k-form α and a s-form β on a manifold M, we define their* ***wedge product*** *$\alpha \wedge \beta$ via*

$$\alpha \wedge \beta = \frac{(k+s)!}{k!s!} \mathrm{Alt}(\alpha \otimes \beta). \tag{1.9}$$

Note in particular that the wedge product of two 1-forms α and β is given by

$$(\alpha \wedge \beta)(v_1, v_2) = \alpha(v_1)\beta(v_2) - \alpha(v_2)\beta(v_1). \tag{1.10}$$

Definition 1.10. *The* ***exterior derivative*** *of a smooth function $f : M \to \mathbb{R}$ is its differential $\mathbf{d}f$, and the* ***exterior derivative*** *$\mathbf{d}\alpha$ of a k-form α with $k > 0$ is the $(k+1)$-form defined by*

$$\mathbf{d}\left(\sum_{i_1, \ldots, i_k} \alpha_{i_1 \ldots i_k} \mathbf{d}x^{i_1} \wedge \ldots \wedge \mathbf{d}x^{i_k} \right) \;=\; \sum_{j} \sum_{i_1, \ldots, i_k} \partial_j \alpha_{i_1 \ldots i_k}\; \mathbf{d}x^j \wedge \mathbf{d}x^{i_1} \wedge \ldots \wedge \mathbf{d}x^{i_k}. \tag{1.11}$$

Note that $\mathbf{d}\alpha$ is linear in α and that $\mathbf{d}(\mathbf{d}\alpha) = 0$ for any k-form α.

Definition 1.11. *We say that a k-form α is* ***closed*** *if*

$$\mathbf{d}\alpha = 0, \tag{1.12}$$

and ***exact*** *if there is a $(k-1)$-form β such that*

$$\alpha = \mathbf{d}\beta. \tag{1.13}$$

Note that every exact form is closed, but that the converse only holds partially: by the Poincaré Lemma, for any closed form α on M and point $m \in M$, there exists a neighborhood of m on which $\alpha = \mathbf{d}\beta$.

Definition 1.12. *The* ***interior product*** $\iota_X\alpha$ *where* X *is a vector field on* M *and* α *is a* k*-form is the* $(k-1)$*-form defined via*

$$(\iota_X\alpha)_m(v_2,\ldots,v_k) \;=\; \alpha_m(X(m),v_2,\ldots,v_k). \tag{1.14}$$

Definition 1.13. *The* ***pull-back*** ψ^*f *of a function* f *by a smooth map* ψ *is the function defined by*

$$\psi^*f \;=\; f\;\circ\;\psi. \tag{1.15}$$

The ***push-forward*** ψ_*f *of a function* f *by a smooth map* ψ *is the function defined by*

$$\psi_*f \;=\; f\;\circ\;\psi^{-1}. \tag{1.16}$$

The ***push-forward*** ψ_*X *of a vector field* X *by a smooth map* ψ *is defined by*

$$(\psi_*X)(\psi(z)) \;=\; \mathbf{d}\psi(z)\cdot X(z). \tag{1.17}$$

The ***pull-back*** ψ^*X *of a vector field* X *by a smooth map* ψ *is defined by*

$$\psi^*X \;=\; (\psi^{-1})_*X. \tag{1.18}$$

The ***pull-back*** $\psi^*\alpha$ *of a* k*-form* α *by a smooth map* ψ *is the* k*-form defined by*

$$(\psi^*\alpha)_m(v_1,\ldots,v_k) \;=\; \alpha_{\psi(m)}(\mathbf{d}\psi\cdot v_1,\ldots,\mathbf{d}\psi\cdot v_k). \tag{1.19}$$

The ***push-forward*** $\psi_*\alpha$ *of a* k*-form* α *by a smooth map* ψ *is the* k*-form defined by*

$$\psi_*\alpha \;=\; (\psi^{-1})^*\alpha. \tag{1.20}$$

Definition 1.14. *The* ***Lie derivative*** $\mathcal{L}_X\alpha$ *of the* k*-form* α *along a vector field* X *with flow* φ_t *is*

$$\mathcal{L}_X\alpha \;=\; \frac{d}{dt}\Big|_{t=0}\varphi_t^*\alpha, \tag{1.21}$$

and in particular, for a smooth function $f:M\to\mathbb{R}$*,* $\mathcal{L}_Xf$ *is the directional derivative*

$$\mathcal{L}_Xf \;=\; \mathbf{d}f\cdot X. \tag{1.22}$$

1.1.2 Riemannian Geometry

We now introduce the main definitions and objects from Riemannian geometry that will be used throughout this dissertation. See [Lang, 1999; Marsden and Ratiu, 1999; Absil et al., 2008; McInerney, 2013; Godinho and Natário, 2014; Jost, 2017; Lee, 2018; Boumal, 2020] for a more detailed presentation.

Definition 1.15. *A **Riemannian metric** on a manifold $\mathcal{Q}$ is a smooth assignment of an inner product $g_q(\cdot,\cdot) = \langle\cdot,\cdot\rangle_q : T_q\mathcal{Q} \times T_q\mathcal{Q} \to \mathbb{R}$ to every point $q \in \mathcal{Q}$. More precisely, the mapping $q \mapsto \langle X(q), Y(q)\rangle_q$ is smooth for any $X, Y \in \mathcal{X}(\mathcal{Q})$ and for any tangent vectors $u, v \in T_p\mathcal{Q}$, $\langle u, v\rangle_p = \langle v, u\rangle_p$ and $\langle u, u\rangle_p \geq 0$ with equality if and only if $u = 0$.*
*A **Riemannian manifold** is a smooth manifold equipped with a Riemannian metric.*

Definition 1.16. *Let $\mathcal{Q}$ be a Riemannian manifold with Riemannian metric $g(\cdot,\cdot) = \langle\cdot,\cdot\rangle$, represented by the positive-definite symmetric matrix (g_{ij}) in local coordinates. We define the **musical isomorphism** $g^\flat : T\mathcal{Q} \to T^*\mathcal{Q}$ via*

$$g^\flat(u)(v) = g_q(u, v) \qquad \forall q \in \mathcal{Q} \text{ and } \forall u, v \in T_q\mathcal{Q}, \tag{1.23}$$

*and its **inverse musical isomorphism** $g^\sharp : T^*\mathcal{Q} \to T\mathcal{Q}$. The Riemannian metric $g(\cdot,\cdot) = \langle\cdot,\cdot\rangle$ induces a **fiber metric** $g^*(\cdot,\cdot) = \langle\!\langle\cdot,\cdot\rangle\!\rangle$ on $T^*\mathcal{Q}$ via*

$$\langle\!\langle u, v\rangle\!\rangle = \langle g^\sharp(u), g^\sharp(v)\rangle \qquad \forall u, v \in T^*\mathcal{Q}, \tag{1.24}$$

represented by the positive-definite symmetric matrix (g^{ij}) in local coordinates, which is the inverse of the Riemannian metric matrix (g_{ij}).

Definition 1.17. *The **Riemannian gradient** $\operatorname{grad} f(q) \in T_q\mathcal{Q}$ at a point $q \in \mathcal{Q}$ of a smooth function $f : \mathcal{Q} \to \mathbb{R}$ is the tangent vector at q such that*

$$\langle \operatorname{grad} f(q), u\rangle = \mathbf{d}f(q)u \qquad \forall u \in T_q\mathcal{Q}. \tag{1.25}$$

This can also be expressed in terms of the inverse musical isomorphism,

$$\operatorname{grad} f(q) = g^\sharp(\mathbf{d}f(q)). \tag{1.26}$$

Definition 1.18. *A **geodesic** in a Riemannian manifold $\mathcal{Q}$ is defined as a parametrized curve $\gamma : [0,1] \to \mathcal{Q}$ which is of minimal local length. It can be thought of as a curve having zero "acceleration" or constant "speed", that is as a generalization of the notion of straight line from Euclidean spaces to Riemannian manifolds.*

Definition 1.19. *Given two points $q, \tilde{q} \in \mathcal{Q}$, a tangent vector in $T_q\mathcal{Q}$ can be transported to $T_{\tilde{q}}\mathcal{Q}$ along a geodesic γ by an operation $\Gamma(\gamma)_q^{\tilde{q}} : T_q\mathcal{Q} \to T_{\tilde{q}}\mathcal{Q}$ called **parallel transport along γ**. We will simply write $\Gamma_q^{\tilde{q}}$ to denote the parallel transport along some geodesic connecting the two points $q, \tilde{q} \in \mathcal{Q}$, and given $A \in \mathcal{X}(\mathcal{Q})$, we will denote by $\Gamma(A)$ the parallel transport along integral curves of A. Note that parallel transport preserves inner products: given a geodesic γ from $q \in \mathcal{Q}$ to $\tilde{q} \in \mathcal{Q}$,*

$$g_q(u,v) = g_{\tilde{q}}\left(\Gamma(\gamma)_q^{\tilde{q}}u, \Gamma(\gamma)_q^{\tilde{q}}v\right) \qquad \forall u, v \in T_q\mathcal{Q}. \tag{1.27}$$

Definition 1.20. *A function $f : \mathcal{Q} \to \mathbb{R}$ is called **L-smooth** if for any two points $q, \tilde{q} \in \mathcal{Q}$ and geodesic γ connecting them,*

$$\left\|\mathrm{gradf}(q) - \Gamma(\gamma)_{\tilde{q}}^{q}\mathrm{gradf}(\tilde{q})\right\| \leq L \ \mathrm{length}(\gamma). \tag{1.28}$$

Definition 1.21. *The **Riemannian Exponential map** $\mathrm{Exp}_q : T_q\mathcal{Q} \to \mathcal{Q}$ at $q \in \mathcal{Q}$ is defined via*

$$\mathrm{Exp}_q(v) = \gamma_v(1), \tag{1.29}$$

*where γ_v is the unique geodesic in $\mathcal{Q}$ such that $\gamma_v(0) = q$ and $\gamma_v'(0) = v$, for any $v \in T_q\mathcal{Q}$. Exp_q is a diffeomorphism in some neighborhood $U \subset T_q\mathcal{Q}$ containing 0, so we can define its inverse map, the **Riemannian Logarithm map** $\mathrm{Log}_q : \mathrm{Exp}_q(U) \to T_q\mathcal{Q}$.*

Definition 1.22. *A **retraction** on a manifold $\mathcal{Q}$ is a smooth mapping $\mathcal{R} : T\mathcal{Q} \to \mathcal{Q}$, such that for any $q \in \mathcal{Q}$, the restriction $\mathcal{R}_q : T_q\mathcal{Q} \to \mathcal{Q}$ of $\mathcal{R}$ to $T_q\mathcal{Q}$ satisfies*

- *$\mathcal{R}_q(0_q) = q$, where 0_q denotes the zero element of $T_q\mathcal{Q}$,*
- *$\mathrm{T}_{0_q}\mathcal{R}_q = \mathrm{Id}_{T_q\mathcal{Q}}$ with the canonical identification $T_{0_q}T_q\mathcal{Q} \simeq T_q\mathcal{Q}$, where $\mathrm{T}_{0_q}\mathcal{R}_q$ is the tangent map of $\mathcal{R}$ at $0_q \in T_q\mathcal{Q}$ and $\mathrm{Id}_{T_q\mathcal{Q}}$ is the identity map on $T_q\mathcal{Q}$.*

The Riemannian Exponential map $\mathrm{Exp} : T\mathcal{Q} \to \mathcal{Q}$ is a natural example of a retraction on a Riemannian manifold $\mathcal{Q}$.

Definition 1.23. *Given two vector fields $X, Y \in \mathcal{X}(\mathcal{Q})$, the* ***covariant derivative*** *$\nabla_X Y \in \mathcal{X}(\mathcal{Q})$ of Y along X is given by*

$$\nabla_X Y(q) \;=\; \lim_{h\to 0} \frac{\Gamma(\gamma)^q_{\gamma(h)} Y(\gamma(h)) - Y(q)}{h}, \tag{1.30}$$

where γ is the unique integral curve of X such that $\gamma(0) = q$, for any $q \in \mathcal{Q}$.

Definition 1.24. *Given vector fields $X = \sum_{j=1}^n X^j \partial_j$ and $Y = \sum_{j=1}^n Y^j \partial_j$ on a Riemannian manifold $\mathcal{Q}$, the* ***Lie bracket*** *$[X, Y]$ is defined in local coordinates as the vector field*

$$[X, Y] = \sum_{i=1}^n \sum_{j=1}^n (X^j \partial_j Y^i - Y^j \partial_j X^i)\partial_i, \tag{1.31}$$

and we say that X and Y ***commute*** *if $[X, Y] = 0$.*

Definition 1.25. *The* ***curvature*** *R of a Riemannian manifold $\mathcal{Q}$ is a map that associates to each pair of vector fields $X, Y \in \mathcal{X}(\mathcal{Q})$ the map $R(X, Y) : \mathcal{X}(\mathcal{Q}) \to \mathcal{X}(\mathcal{Q})$ defined by*

$$R(X, Y)(Z) = \nabla_X(\nabla_Y Z) - \nabla_Y(\nabla_X Z) - \nabla_{[X,Y]} Z, \tag{1.32}$$

for any vector field $Z \in \mathcal{X}(\mathcal{Q})$.

Definition 1.26. *We can define the* ***Riemannian curvature tensor*** *R as the tensor which takes four vector fields $X, Y, Z, W \in \mathcal{X}(\mathcal{Q})$ and returns a scalar via*

$$R(X, Y, Z, W) = \langle R(X, Y)(Z), W\rangle. \tag{1.33}$$

Definition 1.27. *Suppose we have two vector fields $X, Y \in \mathcal{X}(\mathcal{Q})$ such that for all $q \in \mathcal{Q}$, the tangent vectors $X(q)$ and $Y(q)$ are linearly independent. These tangent vectors span a two-dimensional subspace σ_q of $T_q\mathcal{Q}$ at each $q \in \mathcal{Q}$ and thus generate a* ***plane field*** *σ on $\mathcal{Q}$ of two-dimensional subspaces. The* ***sectional curvature*** *$K(\sigma)$ with respect to σ is given by*

$$K(\sigma) = K(X, Y) = \frac{R(X, Y, Y, X)}{\langle X, X\rangle\langle Y, Y\rangle - \langle X, Y\rangle^2}. \tag{1.34}$$

The sectional curvature can be thought of as a "curvature per unit area".

1.1.3 Convexity on Riemannian Manifolds

Definition 1.28. *Given a Riemannian manifold $\mathcal{Q}$, a set $A \subset \mathcal{Q}$ is called* ***geodesically uniquely convex*** *if every two points of A are connected by a unique geodesic in A.*

Definition 1.29. *A function $f : \mathcal{Q} \to \mathbb{R}$ is called* ***geodesically convex*** *if for any two points $q, \tilde{q} \in \mathcal{Q}$ and a geodesic γ connecting them,*

$$f(\gamma(t)) \;\leq\; (1-t)f(q) + tf(\tilde{q}) \qquad \forall t \in [0,1]. \tag{1.35}$$

Remark 1.1. *If f is a smooth geodesically convex function on a geodesically uniquely convex subset A,*

$$f(q) - f(\tilde{q}) \;\geq\; \langle \mathrm{grad} f(\tilde{q}), \mathrm{Log}_{\tilde{q}}(q) \rangle \qquad \forall q, \tilde{q} \in A. \tag{1.36}$$

Definition 1.30. *A function $f : A \to \mathbb{R}$ is called* ***geodesically λ-weakly quasi-convex*** *with respect to $q \in \mathcal{Q}$ for some $\lambda \in (0,1]$ if*

$$\lambda\,(f(q) - f(\tilde{q})) \;\geq\; \langle \mathrm{grad} f(\tilde{q}), \mathrm{Log}_{\tilde{q}}(q) \rangle \qquad \forall \tilde{q} \in A. \tag{1.37}$$

Definition 1.31. *A function $f : A \to \mathbb{R}$ is called* ***geodesically μ-strongly convex*** *for some $\mu > 0$ if*

$$f(q) - f(\tilde{q}) \;\geq\; \langle \mathrm{gradf}(\tilde{q}), \mathrm{Log}_{\tilde{q}}(q) \rangle + \frac{\mu}{2} \|\mathrm{Log}_{\tilde{q}}(q)\|^2 \qquad \forall q, \tilde{q} \in A. \tag{1.38}$$

Remark 1.2. *Remark 1.1 implies that a geodesically convex or strongly convex function is also λ-weakly quasi-convex with $\lambda = 1$. Thus, algorithms introduced for weakly quasi-convex functions can also be used in the geodesically convex and strongly convex cases.*

Remark 1.3. *A local minimum of a geodesically convex or weakly quasi-convex function is also a global minimum. Furthermore, a geodesically strongly convex function either has no minimum or a unique global minimum.*

1.1.4 Examples of Riemannian Manifolds

Example 1.1 (**Unit Sphere** $\mathbb{S}^{n-1}$)**.** *The unit sphere*

$$\boxed{\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n \mid x^\top x = 1\}} \tag{1.39}$$

can be thought of as a Riemannian submanifold with constant positive curvature $K = 1$ *of* $\mathbb{R}^n$ *endowed with the Riemannian metric inherited from the Euclidean inner product* $g_v(u, w) = u^\top w$*. The tangent and normal spaces at any* $v \in \mathbb{S}^{n-1}$ *are given by*

$$T_v\mathbb{S}^{n-1} = \{u \in \mathbb{R}^n \mid u^\top v = 0\}, \qquad \text{and} \qquad (T_v\mathbb{S}^{n-1})^\perp = \{kv \mid k \in \mathbb{R}\}, \tag{1.40}$$

and the orthogonal projections onto $T_v\mathbb{S}^{n-1}$ *and* $(T_v\mathbb{S}^{n-1})^\perp$ *are given by*

$$P_v\xi = (\mathbb{I}_n - xx^\top)\xi \qquad \text{and} \qquad P_v^\perp\xi = xx^\top\xi. \tag{1.41}$$

The obvious choice of projection from $\mathbb{R}^n$ *to* $\mathbb{S}^{n-1}$ *is the rescaling* $v \mapsto \frac{v}{\|v\|_2}$*. This projection can be associated to a retraction on* $\mathbb{S}^{n-1}$*,*

$$\mathcal{R}_x(\xi) = \frac{x + \xi}{\|x + \xi\|_2}. \tag{1.42}$$

More information concerning the geometry of $\mathbb{S}^{n-1}$ *can be found in [Absil et al., 2008].*

Example 1.2 (**Stiefel manifold** $\mathrm{St}(m, n)$)**.** *When endowed with the Riemannian metric* $g_X(A, B) = \mathrm{Trace}(A^\top B)$*, the Stiefel manifold*

$$\boxed{\mathrm{St}(m, n) = \{X \in \mathbb{R}^{n\times m} \mid X^\top X = \mathbb{I}_m\}} \tag{1.43}$$

is a Riemannian submanifold of $\mathbb{R}^{n\times m}$*. The tangent space at any* $X \in \mathrm{St}(m, n)$ *is given by*

$$T_X\mathrm{St}(m, n) = \{Z \in \mathbb{R}^{n\times m} \mid X^\top Z + Z^\top X = 0\}, \tag{1.44}$$

and the orthogonal projection P_X *onto* $T_X\mathrm{St}(m, n)$ *is given by*

$$P_X Z = Z - \frac{1}{2}X(X^\top Z + Z^\top X). \tag{1.45}$$

We can define a projection of any matrix $\tilde{X} \in \mathbb{R}^{n\times m}$ *onto* $\mathrm{St}(m,n)$ *as the solution of the optimization problem*

$$\underset{X\in St(m,n)}{\operatorname{argmin}} \|X - \tilde{X}\|_F. \tag{1.46}$$

From [Hairer et al., 2006], the solution of this minimization problem is given by $X = UV^\top$ *where* $\tilde{X} = U\Sigma V^\top$ *is the Singular Value Decomposition of* $\tilde{X}$ *where* Σ *is a square diagonal* $m\times m$ *matrix. The solution* X *of this problem can also be thought of as the first component of the polar decomposition* $\tilde{X} = XS^{1/2}$ *where* $X \in \mathrm{St}(m,n)$ *and* S *is a* $m\times m$ *symmetric positive-definite matrix. This solution can be written in closed form as* $X = \tilde{X}(\tilde{X}^\top\tilde{X})^{-1/2}$ *(and* $S = \tilde{X}^\top\tilde{X}$*). Thus, a first projection of any given matrix* $Q \in \mathbb{R}^{n\times m}$ *with Singular Value Decomposition* $Q = U\Sigma V^\top$ *onto* $\mathrm{St}(m,n)$ *is given by*

$$Q \mapsto Q(Q^\top Q)^{-1/2} \quad \text{or equivalently} \quad Q \mapsto UV^\top. \tag{1.47}$$

Another method to project a matrix $Y \in \mathbb{R}^{n\times m}$ *onto* $\mathrm{St}(m,n)$ *is obtained via the matrix orthogonalization* $Y \mapsto \mathrm{qf}(Y)$*, which maps the matrix* Y *to the* Q *factor of its* QR *factorization* $Y = QR$ *where* $Q \in \mathrm{St}(m,n)$ *and* R *is an upper triangular* $n\times m$ *matrix with strictly positive diagonal elements [Absil et al., 2008].*

Note that these polar decomposition and matrix orthogonalization can also be used as the basis for the construction of the following two retractions on $\mathrm{St}(m,n)$*:*

$$\mathcal{R}_X(\xi) = (X+\xi)(\mathbb{I}_m + \xi^\top\xi)^{-1/2}, \qquad \mathcal{R}_X(\xi) = \mathrm{qf}(X+\xi). \tag{1.48}$$

More information concerning the geometry of $\mathrm{St}(m,n)$ *can be found in [Absil et al., 2008].*

Example 1.3 (**Lie Groups**)**.** *Consider a Lie group* G *with associated Lie algebra* $\mathfrak{g}$*, and a left trivialization* $TG \simeq G\times\mathfrak{g}$*, obtained via* $(q,\dot{q}) \mapsto (q, \mathrm{L}_{q^{-1}}\dot{q}) \equiv (q,\xi)$*. Suppose* $\mathfrak{g}$ *is equipped with an inner product which induces an inner product on* T_qG *via left trivialization*

$$(v \bullet w)_{T_qG} = (\mathrm{T}_q\mathrm{L}_{q^{-1}}v \bullet \mathrm{T}_q\mathrm{L}_{q^{-1}}w)_{\mathfrak{g}} \qquad \forall v,w \in T_qG. \tag{1.49}$$

With this inner product, we identify $\mathfrak{g} \simeq \mathfrak{g}^*$ *and* $T_qG \simeq T_q^*G \simeq G\times\mathfrak{g}^*$ *via the Riesz representation. Let* $\mathbf{J}:\mathfrak{g}\to\mathfrak{g}^*$ *be chosen such that* $(\mathbf{J}(\xi)\bullet\zeta)$ *is symmetric positive-definite as a bilinear form of* $\xi,\zeta\in\mathfrak{g}$*. Then, the metric* $\langle\cdot,\cdot\rangle : \mathfrak{g}\times\mathfrak{g}\to\mathbb{R}$ *with* $\langle\xi,\zeta\rangle = (\mathbf{J}(\xi)\bullet\zeta)$ *serves as a left-invariant Riemannian metric on* G*. Lie groups are much more structured than Riemannian manifolds and will be described in greater details in the next section.*

1.2 Lie Groups

We now introduce the main definitions and objects from Lie group theory, and some examples of common Lie groups that will be considered throughout this dissertation. See [Varadarajan, 1984; Marsden and Ratiu, 1999; Hall, 2015; Lee et al., 2017; Gallier and Quaintance, 2020] for a more detailed description of Lie Group theory and mechanics on Lie Groups.

1.2.1 Lie Group Theory

Definition 1.32. *A **group** is a nonempty set G with a group operation*

$$G \times G \to G, \qquad (g, h) \mapsto gh,$$

such that the following axioms hold:

1. *Associativity: $(ab)c = a(bc)$ for all $a, b, c \in G$,*
2. *Existence of an identity: $\exists e \in G$ such that $eg = ge = g$ for all $g \in G$,*
3. *Existence of the inverse: for every $g \in G$, there exists $a \in G$ such that $ag = ga = e$.*

The inverse of $g \in G$ is unique and denoted g^{-1}.

Definition 1.33. *A **Lie group** is a differentiable manifold G that has a group structure such that the group operation is a smooth map.*
In a Lie group, the inversion map $g \mapsto g^{-1}$ is smooth.

Definition 1.34. *The **left translation map** and **right translation map** are given by*

$$\mathrm{L}_g : G \to G, \quad h \mapsto gh, \qquad \text{and} \qquad \mathrm{R}_g : G \to G, \quad h \mapsto hg. \tag{1.50}$$

Definition 1.35. *A vector field X on G is called **left-invariant** if*

$$(\mathrm{T}_h \mathrm{L}_g) X(h) = X(gh) \qquad \forall g, h \in G.$$

Definition 1.36. *A **Lie algebra** $\mathfrak{g}$ is the tangent space of the Lie group G at the identity element T_eG equipped with a Lie bracket $[\cdot,\cdot]:\mathfrak{g}\times\mathfrak{g}\to\mathfrak{g}$ which is bilinear, skew symmetric, and satisfies the Jacobi identity*

$$[[a,b],c]+[[c,a],b]+[[b,c],a]=0 \qquad \forall a,b,c\in G. \tag{1.51}$$

Definition 1.37. *Given $\xi\in\mathfrak{g}$, we define a vector field $X_\xi\in\mathcal{X}(G)$ via $X_\xi(g)=\mathrm{T}_e\mathrm{L}_g(\xi)$ for any $g\in G$. Note that for any $\xi\in\mathfrak{g}$, the vector field X_ξ is left-invariant. There is a unique integral curve $\gamma_\xi:\mathbb{R}\to G$ of X_ξ starting at the identity e. The **exponential map** $\exp:\mathfrak{g}\to G$ is defined by*

$$\exp(\xi)=\gamma_\xi(1), \tag{1.52}$$

and satisfies

$$\exp(s\xi)=\gamma_\xi(s). \tag{1.53}$$

The exponential map is a local diffeomorphism from a neighborhood of 0 in $\mathfrak{g}$ onto a neighborhood of e in G.

Definition 1.38. *Given $g\in G$, we define the **inner automorphism** $\mathrm{I}_g:G\to G$ via*

$$\mathrm{I}_g(h)=ghg^{-1} \qquad \forall h\in G. \tag{1.54}$$

*Differentiating the inner automorphism $\mathrm{I}_g(h)$ with respect to h at the identity e gives the **adjoint operator** $\mathrm{Ad}_g:\mathfrak{g}\to\mathfrak{g}$, where*

$$\mathrm{Ad}_g\eta \;=\; \mathrm{T}_e\mathrm{I}_g\eta \;=\; (\mathrm{T}_{g^{-1}}\mathrm{L}_g\cdot\mathrm{T}_e\mathrm{R}_{g^{-1}})\eta \qquad \forall\eta\in\mathfrak{g}, \tag{1.55}$$

*which can be thought of as the linearization of conjugation. Differentiating $\mathrm{Ad}_g\eta$ with respect to g at the identity e gives the **ad operator** $\mathrm{ad}_\xi:\mathfrak{g}\to\mathfrak{g}$, given by*

$$\mathrm{ad}_\xi\eta \;=\; \mathrm{T}_e(\mathrm{Ad}_g\eta)\xi \;=\; [\xi,\eta] \qquad \forall\xi,\eta\in\mathfrak{g}. \tag{1.56}$$

*Now, let $\langle\cdot,\cdot\rangle$ be a pairing between $\mathfrak{g}$ and $\mathfrak{g}^*$. the **coadjoint operator** $\mathrm{Ad}^*_g:\mathfrak{g}^*\to\mathfrak{g}^*$ and the **co-ad operator** $\mathrm{ad}^*_\eta:\mathfrak{g}^*\to\mathfrak{g}^*$ are defined by*

$$\langle\mathrm{Ad}^*_g\alpha,\xi\rangle=\langle\alpha,\mathrm{Ad}_g\xi\rangle \quad \textit{and} \quad \langle\mathrm{ad}^*_\eta\alpha,\xi\rangle=\langle\alpha,\mathrm{ad}_\eta\xi\rangle \quad \forall g\in G,\ \forall\xi,\eta\in\mathfrak{g},\ \forall\alpha\in\mathfrak{g}^*. \tag{1.57}$$

1.2.2 Examples of Lie Groups

Example 1.4 (**Real General Linear Group $\mathbf{GL}(n,\mathbb{R})$**). *The* ***Real General Linear Group*** $\mathrm{GL}(n,\mathbb{R})$ *is the Lie group of isomorphism of* $\mathbb{R}^n$ *to* $\mathbb{R}^n$*, with composition as the group operation. Given a basis in* $\mathbb{R}^n$*, any* $A \in \mathrm{GL}(n,\mathbb{R})$ *can be represented as an invertible* $n \times n$ *matrix, so* $\mathrm{GL}(n,\mathbb{R})$ *can be thought of as the group of real invertible* $n \times n$ *matrices. Then, the group operation is matrix multiplication* $(A,B) \mapsto AB$*, the identity element* e *is the identity matrix* $I_{n\times n}$*, and the inverse operation is matrix inversion* $A \mapsto A^{-1}$*. The Lie algebra of* $\mathrm{GL}(n,\mathbb{R})$ *is the vector space* $\mathfrak{gl}(n)$ *of all linear transformation from* $\mathbb{R}^n$ *to* $\mathbb{R}^n$*, with the commutator bracket*

$$[A,B] = AB - BA. \tag{1.58}$$

The exponential mapping $\exp : \mathfrak{gl}(n) \to \mathrm{GL}(n,\mathbb{R})$ *is given by the usual matrix exponential*

$$\exp A = \sum_{k=0}^{\infty} \frac{A^k}{k!}. \tag{1.59}$$

and $\mathrm{I}_A B = ABA^{-1}$ *for any* $A, B \in \mathrm{GL}(n,\mathbb{R})$*, so the adjoint operator is defined via*

$$\mathrm{Ad}_A \eta = A\eta A^{-1} \qquad \forall \eta \in \mathfrak{gl}(n). \tag{1.60}$$

Definition 1.39. *A* ***matrix Lie group*** *is a closed subgroup of* $\mathrm{GL}(n,\mathbb{R})$ *for some* $n \in \mathbb{N}$*.*

Example 1.5 (**Orthogonal Group $\mathbf{O}(n)$**). *The* ***Orthogonal Group*** $\mathrm{O}(n)$ *is the set of all orthogonal* $n \times n$ *matrices:*

$$\boxed{\mathrm{O}(n) = \{R \in \mathbb{R}^{n\times n} \mid R^{\top} R = \mathbb{I}_n\}.} \tag{1.61}$$

It is a matrix Lie group of dimension $\frac{1}{2}n(n-1)$*. The Orthogonal Group* $\mathrm{O}(n)$ *can be thought of as the special case of the Stiefel manifold* $\mathrm{St}(m,n)$ *where* $m = n$*.*

Its Lie algebra $\mathfrak{o}(n)$ *is the space of skew-symmetric* $n \times n$ *matrices with the matrix commutator* $[A,B] = AB - BA$ *as the Lie bracket.*

Example 1.6 (**Special Orthogonal Group $\mathbf{SO}(n)$**). ***Special Orthogonal Group*** $\mathrm{SO}(n)$ *is the set of all orthogonal* $n \times n$ *matrices with determinant equal to* $+1$*:*

$$\boxed{\mathrm{SO}(n) \;=\; \{R \in \mathbb{R}^{n\times n} \mid R^{\mathsf{T}}R = \mathbb{I}_n,\ \det(R) = +1\} \;=\; \{R \in \mathrm{O}(n) \mid \det(R) = +1\}.} \tag{1.62}$$

It is a matrix Lie group of dimension n*. A Special Orthogonal Group of particular interest is* $\mathrm{SO}(3)$ *since* $R \in \mathrm{SO}(3)$ *can be viewed as the attitude of a rigid body, or as defining a rigid body rotation on* $\mathbb{R}^3$*. Note that any matrix in* $\mathrm{SO}(3)$ *can be written in some orthonormal basis as*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{1.63}$$

for some angle of rotation θ*. In other words,* $\mathrm{SO}(3)$ *is the Lie group of rotations about the origin in* $\mathbb{R}^3$*.*

The Lie algebra $\mathfrak{so}(3)$ *of* $\mathrm{SO}(3)$ *is the space of skew-symmetric* 3×3 *matrices*

$$\mathfrak{so}(3) = \{S \in \mathbb{R}^{3\times 3} \mid S^{\mathsf{T}} = -S\}, \tag{1.64}$$

with the matrix commutator $[A, B] = AB - BA$ *as the Lie bracket. The Lie algebra* $\mathfrak{so}(3)$ *can be identified with* $\mathbb{R}^3$ *via the* ***hat map*** $\hat{\cdot} : \mathbb{R}^3 \to \mathfrak{so}(3)$*,*

$$x = (x_1, x_2, x_3) \quad \mapsto \quad \hat{x} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \tag{1.65}$$

which is such that $\hat{x}y = x \times y$ *for any* $x, y \in \mathbb{R}^3$*. The hat map is also sometimes denoted by* $S(\cdot) : \mathbb{R}^3 \to \mathfrak{so}(3)$*, that is* $S(x)y = x \times y$ *for any* $x, y \in \mathbb{R}^3$*. The inverse of the hat map is called the* ***vee map*** $(\cdot)^{\vee} : \mathfrak{so}(3) \to \mathbb{R}^3$*.*

On $\mathrm{SO}(3)$*, for any* $u, v \in \mathbb{R}^3$ *and* $F \in \mathrm{SO}(3)$*,*

$$\mathrm{ad}_{\hat{u}}\hat{v} = [\hat{u}, \hat{v}] = \hat{u}\hat{v} - \hat{v}\hat{u} = \widehat{u \times v}, \qquad \mathrm{Ad}_F\hat{u} = F\hat{u}F^{\mathsf{T}} = \widehat{Fu}. \tag{1.66}$$

Identifying $\mathfrak{so}(3)^* \simeq \mathfrak{so}(3) \simeq \mathbb{R}^3$*, we have for any* $u, v \in \mathbb{R}^3$ *and* $F \in \mathrm{SO}(3)$ *that*

$$\mathrm{ad}_u v = \hat{u}v = u \times v, \qquad \mathrm{ad}^*_u v = -\hat{u}v = v \times u, \tag{1.67}$$

$$\mathrm{Ad}_F u = Fu, \qquad \mathrm{Ad}^*_F u = F^{\mathsf{T}}u. \tag{1.68}$$

For every matrix $R \in \mathrm{SO}(3)$, there is a skew-symmetric matrix $\xi \in \mathfrak{so}(3)$ such that

$$R \;=\; \exp \xi \;=\; \sum_{k=0}^{\infty} \frac{\xi^k}{k!}, \tag{1.69}$$

and $x \in \mathbb{R}^3$ such that R can be expressed in terms of the Cayley transformation as

$$R = (\mathbb{I}_3 - S(x))(\mathbb{I}_3 + S(x))^{-1}. \tag{1.70}$$

Furthermore, the exponential map on $\mathrm{SO}(3)$ can be computed explicitly using Rodrigues' formula

$$\exp S(v) = \mathbb{I}_3 + \frac{\sin \|v\|}{\|v\|} S(v) + \frac{1 - \cos \|v\|}{\|v\|^2} S(v)^2 \qquad \forall v \in \mathbb{R}^3. \tag{1.71}$$

Example 1.7 (**Special Euclidean Group $\mathbf{SE}(n)$**)**.** *The Special Euclidean group in 3 dimensions, $\mathrm{SE}(3)$, is a semidirect product of $\mathbb{R}^3$ and $\mathrm{SO}(3)$ and is diffeomorphic to $\mathbb{R}^3 \times \mathrm{SO}(3)$. Elements of $\mathrm{SE}(3)$ can be written as $(x, R) \in \mathbb{R}^3 \times \mathrm{SO}(3)$, and the Lie algebra $\mathfrak{se}(3)$ of $\mathrm{SE}(3)$ is composed of elements $(y, A) \in \mathbb{R}^3 \times \mathfrak{so}(3)$.*

The pose of a rigid body can be described by an element (x, R) of $\mathrm{SE}(3)$, consisting of position $x \in \mathbb{R}^3$ and orientation $R \in \mathrm{SO}(3)$.

Example 1.8 (**Circle Group $\mathbf{U}(1)$**)**.** *The **Circle Group** $\mathrm{U}(1)$, also known as first unitary group, is the one-dimensional Lie group of complex numbers of unit modulus with the standard multiplication operation. It can be parametrized via $e^{i\theta}$ for $\theta \in [0, 2\pi)$, and is isomorphic to the special orthogonal group $\mathrm{SO}(2)$ of rotations in the plane.*

*A **circle action** on a manifold M is defined as a one-parameter family of smooth diffeomorphisms $\Phi_\theta : M \to M$ that satisfies the three properties*

$$\Phi_{\theta+2\pi} = \Phi_\theta \quad \text{(periodicity)}, \qquad \Phi_0 = \mathrm{Id}_M \quad \text{(identity)}, \qquad \Phi_{\theta_1+\theta_2} = \Phi_{\theta_1} \circ \Phi_{\theta_2} \quad \text{(additivity)},$$

*for any $\theta, \theta_1, \theta_2 \in \mathrm{U}(1) \cong \mathbb{R} \bmod 2\pi$. The **infinitesimal generator** of a circle action Φ_θ on M is the vector field on M defined by*

$$m \;\mapsto\; \frac{d}{d\theta}\Big|_{\theta=0} \Phi_\theta(m). \tag{1.72}$$

*Given a vector field X on M, a **$\mathbf{U}(1)$ symmetry** for X is a circle action Φ_θ such that*

$$\Phi_\theta^* X = X \qquad \forall \theta \in \mathrm{U}(1). \tag{1.73}$$

2 Symplectic Numerical Integration

2.1 Lagrangian and Hamiltonian Mechanics

We begin this chapter with a standard description of Lagrangian and Hamiltonian mechanics, inspired by [Marsden and Ratiu, 1999; Leimkuhler and Reich, 2004; Hairer et al., 2006; Holm et al., 2009].

Given a n-dimensional manifold $\mathcal{Q}$, a **Lagrangian** is a function $L : T\mathcal{Q} \to \mathbb{R}$. The corresponding **action integral** $\mathfrak{S}$ is defined to be the functional

$$\mathfrak{S}(q) = \int_0^T L(q, \dot{q}) dt, \tag{2.1}$$

over the space of smooth curves $q : [0, T] \to \mathcal{Q}$. **Hamilton's Variational Principle** states that $\delta\mathfrak{S} = 0$ where the variation $\delta\mathfrak{S}$ is induced by an infinitesimal variation δq of the trajectory q that vanishes at the endpoints. Given local coordinates $(q^1, \ldots, q^n)$ on the manifold $\mathcal{Q}$, Hamilton's Variational Principle can be shown to be equivalent to the **Euler–Lagrange equations**

$$\boxed{\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}^k}\right) = \frac{\partial L}{\partial q^k} \qquad \text{for} \quad k = 1, \ldots, n.} \tag{2.2}$$

The **Legendre transform** $\mathbb{F}L : T\mathcal{Q} \to T^*\mathcal{Q}$ of L is defined fiberwise by

$$\mathbb{F}L : (q^i, \dot{q}^i) \mapsto \left(q^i, \frac{\partial L}{\partial \dot{q}^i}\right). \tag{2.3}$$

We say that a Lagrangian L is **regular** or **nondegenerate** if the Hessian matrix $\frac{\partial^2 L}{\partial \dot{q}^2}$ is invertible for every q and $\dot{q}$, and **hyperregular** if the Legendre transform $\mathbb{F}L$ is a diffeomorphism. Note that hyperregularity implies regularity.

Given a hyperregular Lagrangian L on a manifold $\mathcal{Q}$, we can define the associated **energy function** $E : T\mathcal{Q} \to \mathbb{R}$ via

$$E(q,\dot{q}) \;=\; \langle \mathbb{F}L(q,\dot{q}),\dot{q}\rangle - L(q,\dot{q}) \;=\; \langle \frac{\partial L}{\partial \dot{q}}(q,\dot{q}),\dot{q}\rangle - L(q,\dot{q}). \tag{2.4}$$

Note that

$$\frac{dE}{dt} \;=\; \sum_{j=1}^{n}\left[\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right)\dot{q}_j + \frac{\partial L}{\partial \dot{q}_j}\ddot{q}_j - \frac{\partial L}{\partial q_j}\dot{q}_j - \frac{\partial L}{\partial \dot{q}_j}\ddot{q}_j\right] \;=\; \sum_{j=1}^{n}\left[\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j}\right]\dot{q}_j, \tag{2.5}$$

so in particular, along solutions to the Euler–Lagrange equations (2.2),

$$\frac{dE}{dt} = 0, \tag{2.6}$$

and therefore, the energy function E is conserved along trajectories of the corresponding Euler–Lagrange equations.

By defining the **conjugate momentum** $p \in T^*\mathcal{Q}$ of q via the Legendre transform

$$p_k = \frac{\partial L}{\partial \dot{q}^k}, \qquad \text{for} \quad k = 1,\dots,n, \tag{2.7}$$

we can then obtain a **Hamiltonian** $H : T^*\mathcal{Q} \to \mathbb{R}$ corresponding to L via

$$H = E \circ (\mathbb{F}L)^{-1}, \tag{2.8}$$

or in coordinates

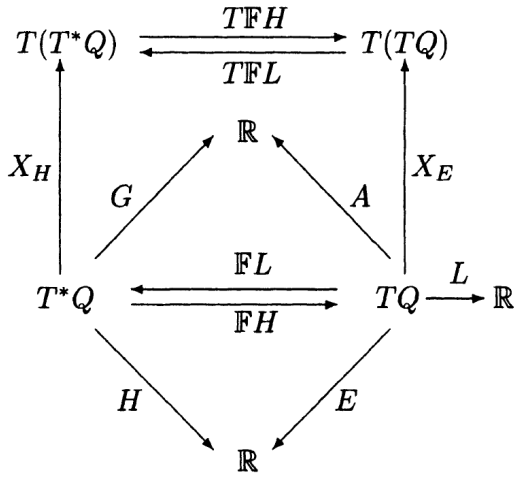
$$H(q,p) \;=\; \langle \mathbb{F}L(q,\dot{q}),\dot{q}\rangle - L(q,\dot{q}) \;=\; \sum_{j=1}^{n} p_j\dot{q}^j - L(q,\dot{q})\Big|_{p_i=\frac{\partial L}{\partial \dot{q}^i}}. \tag{2.9}$$

Therefore, hyperregular Lagrangians on $T\mathcal{Q}$ induce Hamiltonian systems on $T^*\mathcal{Q}$. Now, a Hamiltonian H is called **hyperregular** if $\mathbb{F}H : T^*\mathcal{Q} \to T\mathcal{Q}$ defined by

$$\mathbb{F}H(\alpha)\cdot\beta \;=\; \frac{d}{ds}\Big|_{s=0} H(\alpha + s\beta), \tag{2.10}$$

is a diffeomorphism. As for the Lagrangian, hyperregularity of the Hamiltonian H implies invertibility of the Hessian matrix $\frac{\partial^2 H}{\partial p^2}$ and thus the Hamiltonian H is **nondegenerate**.

Theorem 7.4.3 in [Marsden and Ratiu, 1999] states that hyperregular Lagrangians and hyperregular Hamiltonians correspond in a bijective manner, and if we denote the Hamiltonian and Lagrangian vector fields a by X_H and X_E respectively, the following diagram from [Marsden and Ratiu, 1999] commutes:



We can also define a **Hamiltonian Variational Principle** on the Hamiltonian side in momentum phase space

$$\delta \int_0^T \sum_{j=1}^{n} \left[p_j \dot{q}^j - H(q,p) \right] dt = 0, \tag{2.11}$$

where the variation is induced by an infinitesimal variation δq of the trajectory q that vanishes at the endpoints. This is equivalent to **Hamilton's equations**, given by

$$\boxed{\dot{p}_k = -\frac{\partial H}{\partial q^k}(p,q), \qquad \dot{q}^k = \frac{\partial H}{\partial p_k}(p,q) \qquad \text{for } k = 1, \ldots, n,} \tag{2.12}$$

which can also be shown to be equivalent to the Euler–Lagrange equations (2.2), provided the Lagrangian is hyperregular. Hamilton's equations can be written as

$$\dot{z} = \mathbb{J}^{-1} \nabla H(z), \tag{2.13}$$

where $z = (q,p)$ and

$$\mathbb{J} = \begin{bmatrix} 0 & \mathbb{I}_n \\ -\mathbb{I}_n & 0 \end{bmatrix}. \tag{2.14}$$

We say that a differential equation $\dot{z} = f(z)$ is **locally Hamiltonian** if for every point z_0 in some open set $U \subset \mathbb{R}^{2n}$, there exists a neighbourhood where $f(z) = \mathbb{J}^{-1} \nabla H(z)$ for some function H.

The above discussion can be extended to the case where the Lagrangian is a function $L : T\mathcal{Q} \times \mathbb{R} \to \mathbb{R}$ which depends explicitly on time t. In this case, Hamilton's Variational Principle can also be shown to be equivalent to the Euler–Lagrange equations (2.2). We can then define the **extended Legendre transform** $\bar{\mathbb{F}}L : T\mathcal{Q} \times \mathbb{R} \to T^*\mathcal{Q} \times \mathbb{R}$ via

$$\bar{\mathbb{F}}L : (q, \dot{q}, t) \mapsto \left(q, \frac{\partial L}{\partial \dot{q}}(q, \dot{q}, t), t\right). \tag{2.15}$$

and the same notions of regularity and hyperregularity of Lagrangians as was done earlier in the autonomous case.

Given a hyperregular Lagrangian, we can define the associated time-dependent energy function $E : T\mathcal{Q} \times \mathbb{R} \to \mathbb{R}$ via

$$E(q, \dot{q}, t) = \langle \frac{\partial L}{\partial \dot{q}}(q, \dot{q}, t), \dot{q} \rangle - L(q, \dot{q}, t), \tag{2.16}$$

and now,

$$\frac{dE}{dt} = -\frac{\partial L}{\partial t} \tag{2.17}$$

along solutions of the Euler–Lagrange equations (2.2).

Using the conjugate momentum $p \in T^*\mathcal{Q}$ of q defined via the Legendre transform

$$p_k = \frac{\partial L}{\partial \dot{q}^k}, \qquad \text{for} \quad k = 1, \ldots, n, \tag{2.18}$$

we can then obtain a corresponding time-dependent Hamiltonian $H : T^*\mathcal{Q} \times \mathbb{R} \to \mathbb{R}$ corresponding to L via

$$H = E \circ \left(\bar{\mathbb{F}}L\right)^{-1}, \tag{2.19}$$

or in coordinates

$$H(q, p, t) \;=\; \langle \bar{\mathbb{F}}L(q, \dot{q}, t), \dot{q} \rangle - L(q, \dot{q}, t) \;=\; \sum_{j=1}^{n} p_j \dot{q}^j - L(q, \dot{q}, t)\Big|_{p_i = \frac{\partial L}{\partial \dot{q}^i}}. \tag{2.20}$$

The corresponding Hamiltonian variational principle on the Hamiltonian side in momentum phase space can be shown to be equivalent to Hamilton's equations (2.12), which are also equivalent to the Euler–Lagrange equations (2.2), provided the time-dependent Lagrangian is hyperregular.

2.2 Hamiltonian Systems and Symplecticity

Hamiltonian systems possess a long list of structural invariants and constants of motion, the most important of which are the conservation of the Hamiltonian energy and the conservation of the symplectic 2-form.

Recall from Definition 1.6 that a 2-form α can be written as

$$\alpha_m(v_1, v_2) = \sum_{i_1,i_2} \alpha_{i_1 i_2}(m) v_1^{i_1} v_2^{i_2}, \tag{2.21}$$

where $\alpha_{i_1,i_2}(m) = \alpha_m(\partial_{i_1}, \partial_{i_2})$, for any point $m \in M$ and vectors $v_i = \sum_j v_i^j \partial_j \in T_m M$.

Definition 2.1. *A 2-form α is called* ***non-degenerate*** *if its associated matrix α_{ij} is invertible. A* ***symplectic form*** *ω is a closed non-degenerate 2-form, and a map is* ***symplectic*** *if it preserves the symplectic form.*

We will now give a geometric interpretation for symplecticity in $\mathbb{R}^{2n}$. Given coordinates $(q_1, \ldots, q_n, p_1, \ldots, p_n)$ on $\mathbb{R}^{2n}$, the canonical symplectic form ω on $\mathbb{R}^{2n}$ is given by

$$\omega = \sum_{k=1}^{n} \mathbf{d}q_k \wedge \mathbf{d}p_k. \tag{2.22}$$

Now, any two-dimensional parallelogram lying in $\mathbb{R}^{2n}$ can be parametrized as

$$P = \{t\xi + s\eta \mid s, t \in [0,1]\}, \tag{2.23}$$

where $\xi = \binom{\xi^p}{\xi^q}$ and $\eta = \binom{\eta^p}{\eta^q}$ are linearly independent vectors in the $2n$-dimensional (p,q) space with components $\xi^p, \xi^q, \eta^p, \eta^q \in \mathbb{R}^n$. Now, ω is the sum of the oriented areas of the projections of the parallelogram P onto the coordinate planes (p_i, q_i):

$$\omega(\xi,\eta) \;=\; \sum_{i=1}^{n} \begin{vmatrix} \xi_i^p & \eta_i^p \\ \xi_i^q & \eta_i^q \end{vmatrix} \;=\; \sum_{i=1}^{n} (\xi_i^p \eta_i^q - \xi_i^q \eta_i^p). \tag{2.24}$$

This can be rewritten as $\omega(\xi,\eta) = \xi^\top \mathbb{J} \eta$ where $\mathbb{J}$ is given by equation (2.14). It follows from Definition 2.1 that a linear mapping $A : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ is **symplectic** if

$$A^\top \mathbb{J} A = \mathbb{J}, \qquad \text{or equivalently} \qquad \omega(A\xi, A\eta) = \omega(\xi,\eta) \quad \forall \xi, \eta \in \mathbb{R}^{2n}. \tag{2.25}$$

Similarly, a differentiable map $\varphi : U \to \mathbb{R}^{2n}$ where U is an open subset of $\mathbb{R}^{2n}$ is **symplectic** if the Jacobian matrix $\varphi'(p,q)$ is everywhere symplectic, that is

$$\varphi'(p,q)^\top \mathbb{J} \varphi'(p,q) = \mathbb{J}, \tag{2.26}$$

or equivalently

$$\omega(\varphi'(p,q)\xi, \varphi'(p,q)\eta) = \omega(\xi,\eta) \qquad \forall \xi, \eta \in \mathbb{R}^{2n}. \tag{2.27}$$

As a consequence, for $n = 1$, symplectic mappings are area preserving, and in higher dimensions, symplecticity means that the sum of the oriented areas of the projections of P onto the coordinate planes (p_i, q_i) is preserved.

We will now describe more precisely the relation between Hamiltonian systems and symplecticity of flows, with results presented in greater details in the more comprehensive books [Leimkuhler and Reich, 2004; Hairer et al., 2006; Holm et al., 2009].

The flow $\varphi_t : U \to \mathbb{R}^{2n}$ of a Hamiltonian system is the mapping that advances the solution by time t, that is

$$\varphi_t(p_0, q_0) = \left(p(t, p_0, q_0), q(t, p_0, q_0)\right), \tag{2.28}$$

where $(p(t, p_0, q_0), q(t, p_0, q_0))$ is the solution of the Hamiltonian system with initial values $p(0) = p_0$ and $q(0) = q_0$. We first state an important theorem concerning the symplecticity of the flow of a Hamiltonian system.

Theorem 2.1 ([Poincaré, 1899])**.** *Let $H(p,q)$ be a twice continuously differentiable function on some open set $U \subset \mathbb{R}^{2n}$. Then for each fixed t, the flow φ_t is a symplectic transformation wherever it is defined.*

Furthermore, symplecticity of the flow is a characteristic property of Hamiltonian systems:

Theorem 2.2 ([Hairer et al., 2006])**.** *Let $f : U \to \mathbb{R}^{2n}$ be continuously differentiable on some open set $U \subset \mathbb{R}^{2n}$. Then $\dot{y} = f(y)$ is locally Hamiltonian if and only if its flow $\varphi_t(y)$ is symplectic for all $y \in U$ and for all t sufficiently small.*

In other words, any solution to a Hamiltonian system is a symplectic flow and any symplectic flow corresponds to an appropriate Hamiltonian system.

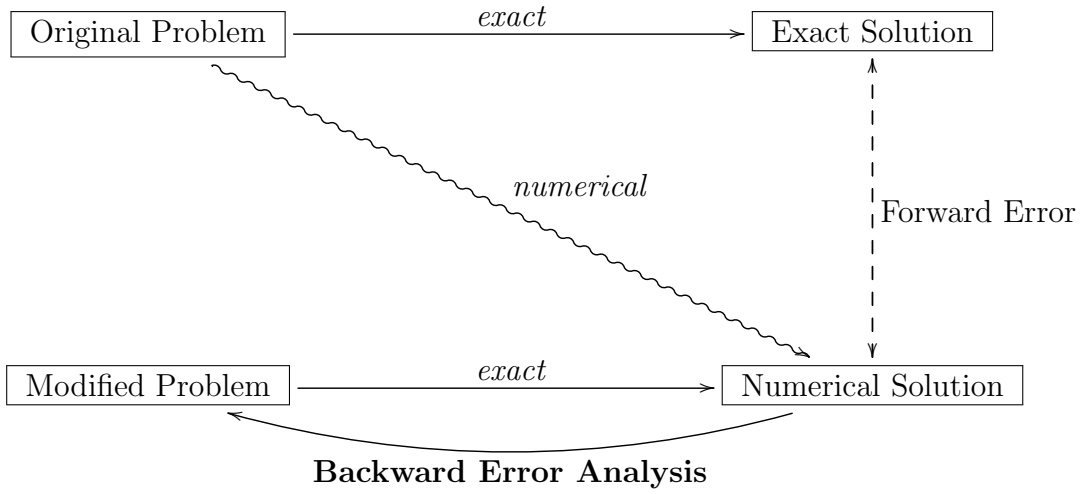
2.3 Symplectic Numerical Integration

We begin with defining what it means for an integrator to be symplectic.

Definition 2.2. *A numerical one-step method is said to be* ***symplectic*** *if the one-step map $y_0 \mapsto y_1$ is symplectic whenever the method is applied to a smooth Hamiltonian system.*

Symplectic integrators form a class of geometric numerical integrators of interest since, when applied to Hamiltonian systems, they yield discrete approximations of the flow that preserve the symplectic 2-form. The preservation of the symplectic 2-form results in the preservation of many qualitative aspects of the underlying dynamical system. We refer the reader to [Iserles and Quispel, 2018] for a brief recent overview of geometric numerical integration, and to [Hairer et al., 2006; Blanes and Casas, 2017] for a more comprehensive presentation of structure-preserving integration techniques.

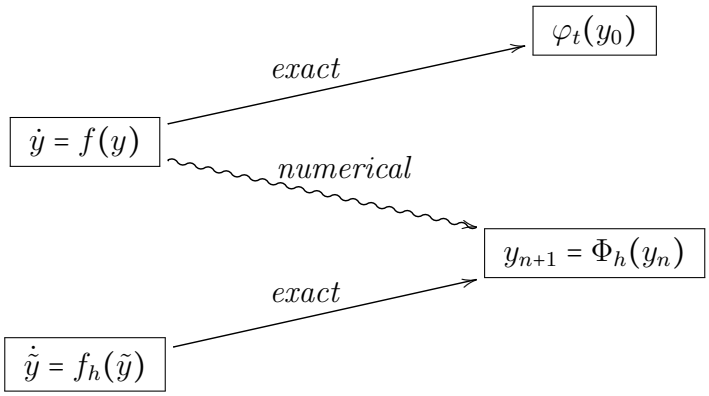
We now look more carefully at one of these preservation properties, namely that when applied to conservative Hamiltonian systems, symplectic integrators show excellent long-time near-energy preservation. This result was first established in 1994 in [Benettin and Giorgilli, 1994] and is presented with great clarity in [Hairer et al., 2006] through Backward Error Analysis. **Backward Error Analysis** (**BEA**) is a way to investigate the error in a numerical solution, first used in [Wilkinson, 1960], where the idea is to find a nearby problem which is solved exactly by the numerical method instead of finding the error of the method.



Consider an ordinary differential equation

$$\dot{y} = f(y), \qquad y(0) = y_0, \tag{2.29}$$

with associated exact time-t flow $\varphi_t(y_0)$, and a numerical method $\Phi_h(y)$ which produces the approximations $y_1, \ldots, y_n$.



Instead of studying the local error $y_1 - \varphi_h(y_0)$ and the global error $y_n - \varphi_{nh}(y_0)$ as in forward error analysis, the idea of backward error analysis is to search for a modified differential equation $\dot{\tilde{y}} = f_h(\tilde{y})$ of the form

$$\dot{\tilde{y}} = f(\tilde{y}) + hf_2(\tilde{y}) + h^2 f_3(\tilde{y}) + \ldots \tag{2.30}$$

such that $y_n = \tilde{y}(nh)$, and to study the difference between the vector fields $f(y)$ and $f_h(y)$. The global error is then $y_n - y(nh) = \tilde{y}(nh) - y(nh)$.

It can be shown that when applied to a Hamiltonian system, a symplectic method solves exactly a nearby Hamiltonian problem:

Theorem 2.3 ([Benettin and Giorgilli, 1994; Hairer et al., 2006])**.** *If a symplectic method $\Phi_h(y)$ is applied to a Hamiltonian system with a smooth Hamiltonian $H : \mathbb{R}^{2n} \to \mathbb{R}$, then the modified equation*

$$\dot{\tilde{y}} = f(\tilde{y}) + hf_2(\tilde{y}) + h^2 f_3(\tilde{y}) + \ldots \tag{2.31}$$

is also Hamiltonian. More precisely, there exist smooth functions $H_j : \mathbb{R}^{2n} \to \mathbb{R}$ for $j = 2, 3, \ldots$ such that

$$f_j(y) = \mathbb{J}^{-1} \nabla H_j(y). \tag{2.32}$$

Furthermore, it can be shown that the symplectic method exhibits excellent near-preservation of the modified nearby Hamiltonian $\tilde{H}$ over exponentially long time intervals:

Theorem 2.4 ([Benettin and Giorgilli, 1994; Hairer et al., 2006])**.** *Consider the numerical solution y_n obtained by applying a symplectic integrator of order ρ with time-step h to a Hamiltonian system with analytic Hamiltonian H on some open set $U \subset \mathbb{R}^{2n}$. If the numerical solution stays in a compact set $K \subset D$, there exists a constant h_0 such that*

$$\tilde{H}(y_n) = \tilde{H}(y_0) + \mathcal{O}(e^{-h_0/2h}), \qquad \text{and} \qquad H(y_n) = H(y_0) + \mathcal{O}(h^\rho), \tag{2.33}$$

over exponentially-long time intervals $nh \leq e^{h_0/2h}$.

This result shows that the numerical solution of a Hamiltonian system obtained using a constant time-step symplectic integrator is exponentially-near to the exact solution of a nearby Hamiltonian system for exponentially-long time. It explains in particular why symplectic integrators exhibit good energy conservation with essentially no accumulation of errors in time, when applied to Hamiltonian systems, and why symplectic methods are best suited to integrate Hamiltonian systems. We refer the reader to [Hairer et al., 2006] for a comprehensive survey of the different techniques to construct symplectic integrators and of their properties.

Formulas for the modified Hamiltonian for symplectic integrators can be derived using B-series [Hairer et al., 2006]. This can be very useful in the context of learning Hamiltonian dynamics from data, for instance. Numerous neural network architectures match a discretization of the Hamiltonian flow to the data to learn the Hamiltonian system. However, the discretization of the Hamiltonian flow is not exact and induces error. While the learnt discrete flow maps can be used very effectively for predictions, the original continuous dynamics cannot be recovered exactly in a naive way from the learnt discrete maps. When we match the updates of a symplectic integrator to the data, we are effectively learning an inverse modified Hamiltonian. To recover the original continuous Hamiltonian, we need to apply corrections to obtain the modified Hamiltonian associated with the symplectic integrator for the learnt Hamiltonian. Recent papers [Zhu et al., 2020; Offen and Ober-Blöbaum, 2022] have started exploring this direction.

We present here two very popular symplectic integrators, the Symplectic Euler and the Störmer–Verlet integrators, and give the formulas for the modified nearby Hamiltonians which are solved exactly when these integrators are applied to a separable Hamiltonian of the form

$$H(q,p) = T(p) + U(q). \tag{2.34}$$

In this section, we will use the Poisson bracket $\{F,G\} = F_q G_p - F_p G_q$ and the matrix commutator $[A,B] = AB - BA$.

The Hamilton equations associated to the separable Hamiltonian (2.34) can be split in two parts as

$$\begin{pmatrix} \dot{q} \\ \dot{p} \end{pmatrix} = \begin{pmatrix} -U_q(q) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ T_p(p) \end{pmatrix}. \tag{2.35}$$

We will denote the symplectic flows of $\begin{pmatrix} \dot{q} \\ \dot{p} \end{pmatrix} = \begin{pmatrix} -U_q(q) \\ 0 \end{pmatrix}$ and $\begin{pmatrix} \dot{q} \\ \dot{p} \end{pmatrix} = \begin{pmatrix} 0 \\ T_p(p) \end{pmatrix}$ by φ_t^U and φ_t^T.

The **Symplectic Euler** integrators

$$\boxed{\begin{array}{lll} \textbf{SE1}: & p_{n+1} = p_n - hU_q(q_n) & \\ & q_{n+1} = q_n + hT_p(p_{n+1}) & \end{array} \qquad \begin{array}{lll} \textbf{SE2}: & q_{n+1} = q_n + hT_p(p_n) & \\ & p_{n+1} = p_n - hU_q(q_{n+1}) & \end{array}} \tag{2.36}$$

can be thought of as the compositions of symplectic flows

$$\Phi_h^{[\mathrm{SE1}]} = \varphi_h^T \circ \varphi_h^U \qquad \text{and} \qquad \Phi_h^{[\mathrm{SE2}]} = \varphi_h^U \circ \varphi_h^T. \tag{2.37}$$

The Symplectic Euler methods are symplectic integrators of order 1 [Hairer et al., 2006]. We can use the Baker–Campbell-Hausdorff (BCH) formula for this first order splitting [Hairer et al., 2006, Section III.4],

$$\begin{aligned} &\exp(hA)\exp(hB) \\ &\quad = \exp\left(h(A+B) + \frac{h^2}{2}[A,B] + \frac{h^3}{12}\Big(\big[B,[B,A]\big] + \big[A,[A,B]\big]\Big) + \frac{h^4}{24}\Big[\big[[B,A],A\big],B\Big] + \mathcal{O}\left(h^5\right)\right), \end{aligned}$$

to obtain the modified Hamiltonians for the separable Hamiltonian (2.34), as done in [Yoshida, 1993; Hairer et al., 2006; Blanes and Casas, 2017]:

$$\tilde{H}^{[\mathrm{SE1}]} = H + \frac{h}{2}\{T,U\} + \frac{h^2}{12}\Big[\big\{U,\{U,T\}\big\} + \big\{T,\{T,U\}\big\}\Big] + \frac{h^3}{24}\Big\{\big\{\{T,U\},U\big\},T\Big\} + \mathcal{O}(h^4),$$
$$\tilde{H}^{[\mathrm{SE2}]} = H + \frac{h}{2}\{U,T\} + \frac{h^2}{12}\Big[\big\{T,\{T,U\}\big\} + \big\{U,\{U,T\}\big\}\Big] + \frac{h^3}{24}\Big\{\big\{\{U,T\},T\big\},U\Big\} + \mathcal{O}(h^4),$$

so when $H : \mathbb{R}^{2n} \to \mathbb{R}$,

$$\tilde{H}^{[\mathrm{SE1}]} = H - \frac{h}{2}T_p^\top U_q + \frac{h^2}{12}\Big[U_q^\top T_{pp} U_q + T_p^\top U_{qq} T_p\Big] + \frac{h^3}{12}T_p^\top U_{qq}^\top \big(T_{pp} + T_{pp}^\top\big) U_q + \mathcal{O}(h^4),$$
$$\tilde{H}^{[\mathrm{SE2}]} = H + \frac{h}{2}T_p^\top U_q + \frac{h^2}{12}\Big[U_q^\top T_{pp} U_q + T_p^\top U_{qq} T_p\Big] - \frac{h^3}{12}T_p^\top U_{qq}^\top \big(T_{pp} + T_{pp}^\top\big) U_q + \mathcal{O}(h^4).$$

The **Störmer–Verlet** integrators

$$\begin{array}{lll} & p_{n+\frac{1}{2}} = p_n - \frac{h}{2}U_q(q_n) & \qquad\qquad q_{n+\frac{1}{2}} = q_n + \frac{h}{2}T_p(p_n) \\ \mathbf{SV1}: & q_{n+1} = q_n + hT_p\left(p_{n+\frac{1}{2}}\right) & \mathbf{SV2}: \quad p_{n+1} = p_n - hU_q\left(q_{n+\frac{1}{2}}\right) \\ & p_{n+1} = p_{n+\frac{1}{2}} - \frac{h}{2}U_q(q_{n+1}) & \qquad\qquad q_{n+1} = q_{n+\frac{1}{2}} + \frac{h}{2}T_p(p_{n+1}) \end{array} \tag{2.38}$$

can be thought of as the symmetric compositions

$$\Phi_h^{[\mathrm{SV1}]} = \varphi_{h/2}^U \circ \varphi_h^T \circ \varphi_{h/2}^U \qquad \text{and} \qquad \Phi_h^{[\mathrm{SV2}]} = \varphi_{h/2}^T \circ \varphi_h^U \circ \varphi_{h/2}^T, \tag{2.39}$$

also known as Strang splittings [Strang, 1968]. The Störmer–Verlet methods are symmetric symplectic methods of order 2. [Hairer et al., 2003] gives a very detailed description of these methods, their different interpretations and numerical properties. Using the symmetric Baker–Campbell-Hausdorff (BCH) formula for the Strang splitting [Hairer et al., 2006, Section III.4] as in [Yoshida, 1993; Hairer et al., 2003, 2006; Blanes and Casas, 2017]

$$\exp\left(\frac{h}{2}A\right)\exp\left(hB\right)\exp\left(\frac{h}{2}A\right) = \exp\left(h(A+B) + \frac{h^3}{24}\Big(2\big[B,[B,A]\big] - \big[A,[A,B]\big]\Big) + \mathcal{O}\left(h^5\right)\right),$$

the modified Hamiltonians are obtained to be

$$\tilde{H}^{[\mathrm{SV1}]} = H + \frac{h^2}{24}\Big[2\Big\{T,\{T,U\}\Big\} - \Big\{U,\{U,T\}\Big\}\Big] + \mathcal{O}(h^4), \tag{2.40}$$
$$\tilde{H}^{[\mathrm{SV2}]} = H + \frac{h^2}{24}\Big[2\Big\{U,\{U,T\}\Big\} - \Big\{T,\{T,U\}\Big\}\Big] + \mathcal{O}(h^4), \tag{2.41}$$

so when $H : \mathbb{R}^{2n} \to \mathbb{R}$,

$$\tilde{H}^{[\mathrm{SV1}]} = H + \frac{h^2}{24}\Big[2T_p^\top U_{qq} T_p - U_q^\top T_{pp} U_q\Big] + \mathcal{O}(h^4), \tag{2.42}$$
$$\tilde{H}^{[\mathrm{SV2}]} = H + \frac{h^2}{24}\Big[2U_q^\top T_{pp} U_q - T_p^\top U_{qq} T_p\Big] + \mathcal{O}(h^4). \tag{2.43}$$

2.4 Variational Integrators

2.4.1 General Description

Variational integrators are constructed by discretizing Hamilton's variational principle, instead of discretizing the Euler–Lagrange equations or Hamilton's equations directly. As a result, they are symplectic, and thus benefit from the nice properties of symplectic integrators presented Section 2.3. In particular, they preserve many invariants and momentum maps, and have excellent long-time near-energy preservation.

Type I. Traditionally, variational integrators have been designed based on the Type I generating function known as the **discrete Lagrangian**, $L_d : Q \times Q \to \mathbb{R}$. The **exact discrete Lagrangian** $L_d^E : Q \times Q \to \mathbb{R}$ of the true flow of the Euler–Lagrange equations can be represented both in a variational form and in a boundary-value form. The latter is given by

$$L_d^E(q_0, q_1; h) = \int_0^h L(q(t), \dot{q}(t))dt, \tag{2.44}$$

where $q(0) = q_0$, $q(h) = q_1$, and the trajectory $q(t)$ satisfies the Euler–Lagrange equations over the time interval $[0, h]$.

A Lagrangian variational integrator is defined by constructing an approximation L_d to the exact discrete Lagrangian L_d^E, and then applying the **discrete Euler–Lagrange equations**,

$$\boxed{p_k = -D_1 L_d(q_k, q_{k+1}), \qquad p_{k+1} = D_2 L_d(q_k, q_{k+1}),} \tag{2.45}$$

which implicitly define the **discrete Hamiltonian map** $\tilde{F}_{L_d} : (q_k, p_k) \mapsto (q_{k+1}, p_{k+1})$, where D_i denotes a partial derivative with respect to the i-th argument.

Numerical methods constructed in this way are called Lagrangian variational integrators as they can be derived from a **discrete Hamilton's principle**, which involves extremizing a discrete action sum

$$\mathfrak{S}_d\left(\{q_k\}_{k=0}^N\right) \equiv \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}), \tag{2.46}$$

subject to fixed boundary conditions on q_0, q_N.

We can define the **discrete Legendre transforms**, $\mathbb{F}^{\pm}L_d : \mathcal{Q} \times \mathcal{Q} \to T^*\mathcal{Q}$ via

$$\mathbb{F}^{+}L_d : (q_k, q_{k+1}) \mapsto (q_{k+1}, p_{k+1}) = (q_{k+1}, D_2 L_d(q_k, q_{k+1})), \tag{2.47}$$

$$\mathbb{F}^{-}L_d : (q_k, q_{k+1}) \mapsto (q_k, p_k) = (q_k, -D_1 L_d(q_k, q_{k+1})), \tag{2.48}$$

and the discrete Hamiltonian map can be expressed as $\tilde{F}_{L_d} \equiv (\mathbb{F}^{+}L_d) \circ (\mathbb{F}^{-}L_d)^{-1}$.

The error analysis is greatly simplified by Theorem 2.3.1 of [Marsden and West, 2001], which states that if a discrete Lagrangian, $L_d : Q \times Q \to \mathbb{R}$, approximates the exact discrete Lagrangian $L_d^E : Q \times Q \to \mathbb{R}$ to order r, that is,

$$L_d(q_0, q_1; h) = L_d^E(q_0, q_1; h) + \mathcal{O}(h^{r+1}), \tag{2.49}$$

then the discrete Hamiltonian map $\tilde{F}_{L_d} : (q_k, p_k) \mapsto (q_{k+1}, p_{k+1})$, viewed as a one-step method, has order of accuracy r. Many other properties of the integrator, such as momentum conservation properties of the method, can be determined by analyzing the associated discrete Lagrangian, as opposed to analyzing the integrator directly.

More recently, variational integrators have been designed based on the framework of Type II/III generating functions, commonly referred to as **discrete Hamiltonians** (see [Lall and West, 2006; Leok and Zhang, 2011; Schmitt and Leok, 2017]). Hamiltonian variational integrators are derived by discretizing Hamilton's phase space principle.

Type II. The boundary-value formulation of the exact Type II generating function of the time-h flow of Hamilton's equations is given by the **exact discrete right Hamiltonian**,

$$H_d^{+,E}(q_0, p_1; h) = p_1^{\top} q_1 - \int_0^h \left[p(t)^{\top} \dot{q}(t) - H(q(t), p(t)) \right] dt, \tag{2.50}$$

where (q, p) satisfies Hamilton's equations on the time interval $[0, h]$ with boundary conditions $q(0) = q_0$, $p(h) = p_1$.

A Type II Hamiltonian variational integrator is constructed by approximating the exact discrete right Hamiltonian $H_d^{+,E}$ via an approximate discrete Hamiltonian H_d^{+}, and applying the **discrete right Hamilton's equations**,

$$\boxed{p_0 = D_1 H_d^{+}(q_0, p_1), \qquad q_1 = D_2 H_d^{+}(q_0, p_1),} \tag{2.51}$$

which implicitly defines the integrator, $\tilde{F}_{H_d^{+}} : (q_0, p_0) \mapsto (q_1, p_1)$.

Theorem 2.3.1 of [Marsden and West, 2001], which greatly simplified the variational error analysis for Lagrangian variational integrators, has an analogue for Hamiltonian variational integrators. Theorem 2.2 in [Schmitt and Leok, 2017] states that if a discrete right Hamiltonian H_d^+ approximates the exact discrete right Hamiltonian $H_d^{+,E}$ to order r, that is,

$$H_d^+(q_0, p_1; h) = H_d^{+,E}(q_0, p_1; h) + \mathcal{O}(h^{r+1}), \tag{2.52}$$

then the **discrete right Hamilton's map** $\tilde{F}_{H_d^+} : (q_k, p_k) \mapsto (q_{k+1}, p_{k+1})$, viewed as a one-step method, is order r accurate.

Type III. The boundary-value formulation of the exact Type III generating function of the time-h flow of Hamilton's equations is the **exact discrete left Hamiltonian**,

$$H_d^{-,E}(q_1, p_0; h) = -p_0^\top q_0 - \int_0^h \left[p(t)^\top \dot{q}(t) - H(q(t), p(t))\right] dt, \tag{2.53}$$

where (q, p) satisfies Hamilton's equations on the time interval $[0, h]$ with boundary conditions $q(h) = q_1$, $p(0) = p_0$.

A Type III Hamiltonian variational integrator is constructed by approximating the exact discrete left Hamiltonian $H_d^{-,E}$ via an approximate discrete Hamiltonian H_d^-, and applying the **discrete left Hamilton's equations**,

$$\boxed{p_1 = -D_1 H_d^-(q_1, p_0), \qquad q_0 = -D_2 H_d^-(q_1, p_0),} \tag{2.54}$$

which implicitly defines the integrator, $\tilde{F}_{H_d^-} : (q_0, p_0) \mapsto (q_1, p_1)$. As mentioned in [Schmitt and Leok, 2017], the proof of Theorem 2.2 in [Schmitt and Leok, 2017] can be easily adjusted to derive an equivalent error analysis theorem for the discrete left Hamiltonian case, which states that if a discrete left Hamiltonian H_d^- approximates the exact discrete left Hamiltonian $H_d^{-,E}$ to order r, i.e.,

$$H_d^-(q_1, p_0; h) = H_d^{-,E}(q_1, p_0; h) + \mathcal{O}(h^{r+1}), \tag{2.55}$$

then it follows that the **discrete left Hamilton's map** $\tilde{F}_{H_d^-} : (q_k, p_k) \mapsto (q_{k+1}, p_{k+1})$, viewed as a one-step method, is order r accurate.

Typical examples of Hamiltonian and Lagrangian variational integrators include Galerkin variational integrators [Marsden and West, 2001; Leok and Zhang, 2011], Taylor variational integrators [Schmitt et al., 2018], and prolongation-collocation variational integrators [Leok and Shingel, 2012a].

The Type I and Type II/III approaches will produce equivalent integrators in many cases. In particular, this equivalence was established in [Schmitt et al., 2018] for Taylor variational integrators provided the Lagrangian is hyperregular, and in [Leok and Zhang, 2011] for generalized Galerkin variational integrators constructed using the same choices of basis functions and numerical quadrature formula provided the Hamiltonian is hyperregular.

Note however that Hamiltonian and Lagrangian variational integrators are not always equivalent. In particular, it was shown in [Schmitt and Leok, 2017] that even when the Hamiltonian and Lagrangian integrators are analytically equivalent, they might still have different numerical properties because of numerical conditioning issues. Even more to the point, Lagrangian variational integrators cannot always be constructed when the underlying Hamiltonian is degenerate, and in that situation, Hamiltonian variational integrators are the more natural choice. Depending on the form of the Hamiltonian and the approximation method used to design the corresponding approximate discrete Hamiltonian, one of the Type II or Type III approaches might be more convenient than the other, in the sense that it might allow for an explicit algorithm or might allow for higher-order methods given some constraints on the type of methods permitted. In Section 2.7.1, we will examine a transformation commonly used to construct variable time-step symplectic integrators, which results in a degenerate Hamiltonian in most cases of interest, such as the optimization application considered in Section 3.3. We will apply Hamiltonian variational integrators to the resulting transformed Hamiltonian system. For the optimization application presented in Section 3.3, we will prefer Type II Hamiltonian Taylor variational integrators to their Type III analogues, and this choice will be justified carefully based on the order and explicitness of the resulting methods.

2.4.2 Taylor Variational Integrators

A discrete approximate Lagrangian or Hamiltonian is obtained by approximating the flow map and the trajectory associated with the boundary values using a Taylor method, and by approximating the integral using a quadrature rule. The Taylor variational integrator is then generated implicitly by the discrete Euler–Lagrange equations associated with the discrete Lagrangian or by the discrete Hamilton's equations associated with the discrete Hamiltonian.

More explicitly, we first construct ρ-order and $(\rho+1)$-order Taylor methods $\Psi_h^{(\rho)}$ and $\Psi_h^{(\rho+1)}$ approximating the exact time-h flow map $\Phi_h : TQ \to TQ$ corresponding to the Euler–Lagrange equation in the Type I case or the exact time-h flow map $\Phi_h : T^*Q \to T^*Q$ corresponding to Hamilton's equation in the Type II/III cases.

Let $\pi_{T^*Q} : (q,p) \mapsto p$ and $\pi_Q : (q,p) \mapsto q$. Given a quadrature rule of order s with weights and nodes (b_i, c_i) for $i = 1, \ldots, m$, the Taylor variational integrators are then constructed as follows:

Type I Lagrangian Taylor Variational Integrator (LTVI)

(i) Approximate $\dot{q}(0) = v_0$ by the solution $\tilde{v}_0$ of the problem $q_1 = \pi_Q \circ \Psi_h^{(\rho+1)}(q_0, \tilde{v}_0)$.

(ii) Generate approximations $(q_{c_i}, v_{c_i}) \approx (q(c_i h), \dot{q}(c_i h))$ via $(q_{c_i}, v_{c_i}) = \Psi_{c_i h}^{(\rho)}(q_0, \tilde{v}_0)$.

(iii) Apply the quadrature rule to obtain the associated discrete Lagrangian

$$L_d(q_0, q_1; h) = h \sum_{i=1}^{m} b_i L(q_{c_i}, v_{c_i}). \tag{2.56}$$

(iv) The variational integrator is then defined by the implicit discrete Euler–Lagrange equations

$$p_0 = -D_1 L_d(q_0, q_1), \qquad p_1 = D_2 L_d(q_0, q_1). \tag{2.57}$$

Type II Hamiltonian Taylor Variational Integrator (HTVI)

(i) Approximate $p(0) = p_0$ by the solution $\tilde{p}_0$ of the problem $p_1 = \pi_{T^*Q} \circ \Psi_h^{(\rho)}(q_0, \tilde{p}_0)$.

(ii) Generate approximations $(q_{c_i}, p_{c_i}) \approx (q(c_i h), p(c_i h))$ via $(q_{c_i}, p_{c_i}) = \Psi_{c_i h}^{(\rho)}(q_0, \tilde{p}_0)$.

(iii) Approximate q_1 via $\tilde{q}_1 = \pi_Q \circ \Psi_h^{(\rho+1)}(q_0, \tilde{p}_0)$.

(iv) Use the continuous Legendre transform to obtain $\dot{q}_{c_i} = \frac{\partial H}{\partial p_{c_i}}$.

(v) Apply the quadrature rule to obtain the associated discrete right Hamiltonian

$$H_d^+(q_0, p_1; h) = p_1^\top \tilde{q}_1 - h \sum_{i=1}^{m} b_i \left[p_{c_i}^\top \dot{q}_{c_i} - H(q_{c_i}, p_{c_i}) \right]. \tag{2.58}$$

(vi) The variational integrator is then defined by the implicit discrete right Hamilton's equations

$$q_1 = D_2 H_d^+(q_0, p_1), \qquad p_0 = D_1 H_d^+(q_0, p_1). \tag{2.59}$$

Type III Hamiltonian Taylor Variational Integrator (HTVI)

(i) Approximate $q(0) = q_0$ by the solution $\tilde{q}_0$ of the problem $q_1 = \pi_Q \circ \Psi_h^{(\rho+1)}(\tilde{q}_0, p_0)$.

(ii) Generate approximations $(q_{c_i}, p_{c_i}) \approx (q(c_i h), p(c_i h))$ via $(q_{c_i}, p_{c_i}) = \Psi_{c_i h}^{(\rho)}(\tilde{q}_0, p_0)$.

(iii) Use the continuous Legendre transform to obtain $\dot{q}_{c_i} = \frac{\partial H}{\partial p_{c_i}}$.

(iv) Apply the quadrature rule to obtain the associated discrete left Hamiltonian

$$H_d^-(q_1, p_0; h) = -p_0^\top \tilde{q}_0 - h \sum_{i=1}^{m} b_i \left[p_{c_i}^\top \dot{q}_{c_i} - H(q_{c_i}, p_{c_i}) \right]. \tag{2.60}$$

(v) The variational integrator is then defined by the implicit discrete left Hamilton's equations

$$p_1 = -D_1 H_d^-(q_1, p_0), \qquad q_0 = -D_2 H_d^-(q_1, p_0). \tag{2.61}$$

The following error analysis result concerning the order of accuracy of Lagrangian Taylor variational integrators can be derived:

Theorem 2.5 ([Schmitt et al., 2018])**.** *Suppose the Lagrangian L is Lipschitz continuous in both variables and sufficiently regular for $\Psi_h^{(\rho+1)}$ to be well-defined. Then the discrete Lagrangian $L_d(q_0, q_1)$, obtained using the above construction, approximates $L_d^E(q_0, q_1)$ with at least order of accuracy $\min(\rho+1, s)$. By Theorem 2.3.1 in [Marsden and West, 2001], the associated discrete Hamiltonian map has the same order of accuracy.*

Similar error analysis results can be derived for Type II and Type III Hamiltonian Taylor variational integrators:

Theorem 2.6 ([Duruisseaux et al., 2021])**.** *Suppose the Hamiltonian H and its partial derivative $\frac{\partial H}{\partial p}$ are Lipschitz continuous in both variables, and H is sufficiently regular for $\Psi_h^{(\rho+1)}$ to be well-defined. Then, the discrete Hamiltonian $H_d^{\pm}$ obtained using the above construction, approximates $H_d^{\pm,E}$ with at least order of accuracy $\min(\rho+1, s)$. By Theorem 2.2 in [Schmitt and Leok, 2017] (or its analogue for the left Hamiltonian case), the associated discrete Hamiltonian map has the same order of accuracy.*

Proof. See Appendix A.1 □

Taylor variational integrators were inspired by a resurgence of interest in high-order Taylor methods for celestial mechanics that has been fueled by the continued progress in automatic differentiation software (see for instance [Barrio, 2005; Neidinger, 2005; Pearlmutter, 2007; Neidinger, 2013; Bettencourt et al., 2019; Biscani and Izzo, 2021]). For high-order Taylor methods, the key to an efficient implementation relies upon efficient automatic differentiation software to compute higher-order gradients.

Implicit modified Taylor methods have been proposed to deal with stiff ordinary differential equations [Kirlinger and Corliss, 1991], while Taylor variational integrators provide a class of Taylor-based integrators to deal with conservative Hamiltonian systems, and can be viewed as a predictor-corrector method that applies a symplectic correction to the Taylor method.

2.5 Forced Variational Integrators

External forcing and control can be incorporated into the variational integrators framework [Marsden and West, 2001; Ober-Blöbaum et al., 2011].

Let $u(t)$ be a control parameter in some control manifold $\mathcal{U}$, and consider a **Lagrangian control force** $f_L : T\mathcal{Q} \times \mathcal{U} \to T^*\mathcal{Q}$. Hamilton's variational principle

$$\delta \int_0^T L(q(t), \dot{q}(t))dt = 0, \tag{2.62}$$

can be modified into the **Lagrange–d'Alembert Variational Principle**

$$\delta \int_0^T L(q(t), \dot{q}(t))dt \; + \; \int_0^T f_L(q(t), \dot{q}(t), u(t)) \cdot \delta q(t)dt \; = \; 0, \tag{2.63}$$

where the variation is induced by an infinitesimal variation δq that vanishes at the endpoints. This variational principle is equivalent to the **forced Euler–Lagrange equations**

$$\boxed{\frac{\partial L}{\partial q}(q, \dot{q}) - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}(q, \dot{q})\right) + f_L(q, \dot{q}, u) \; = \; 0.} \tag{2.64}$$

Using a discrete Lagrangian L_d to approximate the exact discrete Lagrangian,

$$L_d(q_k, q_{k+1}) \; \approx \; \int_{t_k}^{t_{k+1}} L(q(t), \dot{q}(t))dt, \tag{2.65}$$

and **discrete Lagrangian control forces** $f_d^{\pm} : \mathcal{Q} \times \mathcal{Q} \times \mathcal{U} \to T^*\mathcal{Q}$ to approximate the virtual work of the Lagrangian control force f_L,

$$f_d^-(q_k, q_{k+1}, u_k) \cdot \delta q_k + f_d^+(q_k, q_{k+1}, u_k) \cdot \delta q_{k+1} \; \approx \; \int_{t_k}^{t_{k+1}} f_L(q(t), \dot{q}(t), u(t)) \cdot \delta q(t)dt, \tag{2.66}$$

one can obtain a **forced variational integrator** from the **forced discrete Euler–Lagrange equations**

$$\boxed{D_2L_d(q_{k-1}, q_k) + D_1L_d(q_k, q_{k+1}) + f_d^+(q_{k-1}, q_k, u_{k-1}) + f_d^-(q_k, q_{k+1}, u_k) \; = \; 0,} \tag{2.67}$$

which can also be written in Hamiltonian form as

$$\boxed{\begin{aligned} p_k \; &= \; -D_1L_d(q_k, q_{k+1}) - f_d^-(q_k, q_{k+1}, u_k), \\ p_{k+1} \; &= \; D_2L_d(q_k, q_{k+1}) + f_d^+(q_k, q_{k+1}, u_k). \end{aligned}} \tag{2.68}$$

2.6 Constrained Variational Integrators

We now investigate how holonomic constraints can be incorporated into the design of discrete variational integrators to constrain the numerical discretization of a continuous Hamiltonian or Lagrangian system to a certain constraint manifold. Enforcing holonomic constraints in geometric numerical integrators has been studied extensively in the past (see for instance [Marsden and Ratiu, 1999; Marsden and West, 2001; Hairer et al., 2006; Holm et al., 2009]), and some work has been done from the variational perspective for the Type I Lagrangian formulation in [Marsden and West, 2001] via augmented Lagrangians in some appropriate extended spaces.

We will first show in Section 2.6.1 the equivalence between constrained variational principles and constrained Euler–Lagrange equations, both in continuous and discrete time, before deriving analogous results for both the Type II and Type III Hamiltonian formulations of variational integrators in Section 2.6.2. In Section 2.6.3, we will exploit existing error analysis theorems for unconstrained variational integrators from [Marsden and West, 2001] and [Schmitt and Leok, 2017] to obtain variational error analysis results for the discrete maps defined implicitly by the discrete constrained Euler–Lagrange and discrete constrained Hamilton's equations.

2.6.1 Constrained Variational Lagrangian Mechanics

Suppose we are given a configuration manifold $\mathcal{M}$, and a holonomic constraint function $\mathcal{C} : \mathcal{M} \to \mathbb{R}^d$. Assuming that $0 \in \mathbb{R}^d$ is a regular point of $\mathcal{C}$, we can constrain the dynamics to the constraint submanifold $\mathcal{Q} = \mathcal{C}^{-1}(0)$, which is truly a submanifold of $\mathcal{M}$ (see [Abraham et al., 1988; Marsden and West, 2001]).

We will now present a variational formulation of Lagrangian mechanics with holonomic constraints $\mathcal{C}(q)$ using Lagrange multipliers $\lambda : [0, T] \to \Lambda$.

Continuous Constrained Variational Lagrangian Mechanics

We first present an equivalence between the continuous constrained variational principle and the continuous constrained Euler–Lagrange equations:

Theorem 2.7 ([Marsden and West, 2001; Duruisseaux and Leok, 2022a])**.** *Consider the* ***constrained action functional*** $\mathfrak{S} : C^2([0,T], \mathcal{Q} \times \Lambda) \to \mathbb{R}$ *given by*

$$\mathfrak{S}(q(\cdot), \lambda(\cdot)) = \int_0^T \left[L(q(t), \dot{q}(t)) - \langle \lambda(t), \mathcal{C}(q(t)) \rangle \right] dt. \tag{2.69}$$

The condition that $\mathfrak{S}(q(\cdot), \lambda(\cdot))$ *is stationary with respect to the boundary conditions* $\delta q(0) = 0$ *and* $\delta q(T) = 0$ *is equivalent to* $(q(\cdot), \lambda(\cdot))$ *satisfying the* ***constrained Euler–Lagrange equations***

$$\frac{\partial L}{\partial q} - \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \langle \lambda, \nabla \mathcal{C}(q) \rangle, \qquad \mathcal{C}(q) = 0. \tag{2.70}$$

Proof. See Appendix A.2.1. □

Remark 2.1. *The constrained equations* (2.70) *can be thought of as the Euler–Lagrange equations coming from the augmented Lagrangian* $\bar{L}\left(q, \lambda, \dot{q}, \dot{\lambda}\right) = L(q, \dot{q}) - \langle \lambda, \mathcal{C}(q) \rangle$.

Now, consider the function $\mathcal{S}(q_0, q_T)$ given by the extremal value of the constrained action functional $\mathfrak{S}$ over the family of curves $(q(\cdot), \lambda(\cdot))$ satisfying the boundary conditions $q(0) = q_0$ and $q(T) = q_T$:

$$\mathcal{S}(q_0, q_T) = \operatorname*{ext}_{\substack{(q,\lambda)\in C^2([0,T], \mathcal{Q}\times\Lambda) \\ q(0)=q_0, \quad q(T)=q_T}} \mathfrak{S}(q(\cdot), \lambda(\cdot)). \tag{2.71}$$

The following theorem shows that $\mathcal{S}(q_0, q_T)$ is a generating function for the flow of the continuous constrained Euler–Lagrange equations:

Theorem 2.8 ([Marsden and West, 2001; Duruisseaux and Leok, 2022a])**.** *The exact time-T flow map of Hamilton's equations* $(q_0, p_0) \mapsto (q_T, p_T)$ *is implicitly given by the following relations:*

$$D_1 \mathcal{S}(q_0, q_T) = -\frac{\partial L}{\partial \dot{q}}(q_0, \dot{q}(0)), \qquad D_2 \mathcal{S}(q_0, q_T) = \frac{\partial L}{\partial \dot{q}}(q_T, \dot{q}(T)). \tag{2.72}$$

In particular, $\mathcal{S}(q_0, q_T)$ *is a Type I generating function that generates the exact flow of the constrained Euler–Lagrange equations* (2.70)*.*

Proof. See Appendix A.2.4. □

Discrete Constrained Variational Lagrangian Mechanics

We now introduce a discrete variational formulation of Lagrangian mechanics which includes holonomic constraints.

Suppose we are given a partition $0 = t_0 < t_1 < \ldots < t_N = T$ of the interval $[0, T]$, and a discrete curve in $\mathcal{Q} \times \Lambda$ denoted by $\{(q_k, \lambda_k)\}_{k=0}^N$ such that $q_k \approx q(t_k)$ and $\lambda_k \approx \lambda(t_k)$. We will formulate discrete constrained variational Lagrangian mechanics in terms of the following discrete analogues of the constrained action functional $\mathfrak{S}$ given by equation (2.69):

$$\mathfrak{S}_d^+\left(\{(q_k, \lambda_k)\}_{k=0}^N\right) = \sum_{k=0}^{N-1} \left[L_d(q_k, q_{k+1}) - \langle \lambda_{k+1}, \mathcal{C}(q_{k+1})\rangle\right], \tag{2.73}$$

$$\mathfrak{S}_d^-\left(\{(q_k, \lambda_k)\}_{k=0}^N\right) = \sum_{k=0}^{N-1} \left[L_d(q_k, q_{k+1}) - \langle \lambda_k, \mathcal{C}(q_k)\rangle\right], \tag{2.74}$$

where

$$L_d(q_k, q_{k+1}) \approx \operatorname*{ext}_{\substack{(q,\lambda)\in C^2([t_k,t_{k+1}],\mathcal{Q}\times\Lambda) \\ q(t_k)=q_k,\quad q(t_{k+1})=q_{k+1}}} \int_{t_k}^{t_{k+1}} L(q(t), \dot{q}(t))dt. \tag{2.75}$$

We can now derive discrete analogues to Theorem 2.7 relating discrete Type I variational principles to discrete Euler–Lagrange equations:

Theorem 2.9 ([Marsden and West, 2001; Duruisseaux and Leok, 2022a])**.** *The Type I discrete Hamilton's variational principles*

$$\delta\mathfrak{S}_d^\pm\left(\{(q_k, \lambda_k)\}_{k=0}^N\right) = 0, \tag{2.76}$$

are equivalent to the ***discrete constrained Euler–Lagrange equations***

$$\boxed{D_1 L_d(q_k, q_{k+1}) + D_2 L_d(q_{k-1}, q_k) = \langle \lambda_k, \nabla \mathcal{C}(q_k)\rangle, \qquad \mathcal{C}(q_k) = 0,} \tag{2.77}$$

where $L_d(q_k, q_{k+1})$ is defined via equation (2.75)*.*

Proof. See Appendix A.2.7. □

Remark 2.2. *These discrete constrained Euler–Lagrange equations can be thought of as the discrete Euler–Lagrange equations coming from the augmented discrete Lagrangians*

$$\bar{L}_d^+(q_k, \lambda_k, q_{k+1}, \lambda_{k+1}) = L_d(q_k, q_{k+1}) - \langle \lambda_{k+1}, C(q_{k+1}) \rangle, \tag{2.78}$$

$$\bar{L}_d^-(q_k, \lambda_k, q_{k+1}, \lambda_{k+1}) = L_d(q_k, q_{k+1}) - \langle \lambda_k, C(q_k) \rangle. \tag{2.79}$$

2.6.2 Constrained Variational Hamiltonian Mechanics

We now derive analogous results to those of Section 2.6.1 from the Hamiltonian perspective. As in the Lagrangian case, we will assume we have a configuration manifold $\mathcal{M}$, a holonomic constraint function $C : \mathcal{M} \to \mathbb{R}^d$, and that the dynamics are constrained to the submanifold $\mathcal{Q} = C^{-1}(0)$.

Continuous Constrained Variational Hamiltonian Mechanics

The following theorem presents the equivalence between a continuous constrained variational principle and continuous constrained Hamilton's equations in the Type II case, generalizing Lemma 2.1 from [Leok and Zhang, 2011] to include holonomic constraints:

Theorem 2.10 ([Duruisseaux and Leok, 2022a])**.** *Consider the* ***Type II constrained action functional*** $\mathfrak{S} : C^2([0,T], T^*\mathcal{Q} \times \Lambda) \to \mathbb{R}$ *given by*

$$\mathfrak{S}(q(\cdot), p(\cdot), \lambda(\cdot)) = p(T)q(T) - \int_0^T \left[p(t)\dot{q}(t) - H(q(t), p(t)) - \langle \lambda(t), C(q(t)) \rangle \right] dt. \tag{2.80}$$

The condition that the action functional $\mathfrak{S}(q(\cdot), p(\cdot), \lambda(\cdot))$ *is stationary with respect to the boundary conditions* $\delta q(0) = 0$ *and* $\delta p(T) = 0$ *is equivalent to* $(q(\cdot), p(\cdot), \lambda(\cdot))$ *satisfying* ***Hamilton's constrained equations***

$$\dot{q} = \frac{\partial H}{\partial p}(q,p), \qquad \dot{p} = -\frac{\partial H}{\partial q}(q,p) - \langle \lambda, \nabla C(q) \rangle, \qquad C(q) = 0. \tag{2.81}$$

Proof. See Appendix A.2.2. □

As in the Type II case, we can derive a theorem relating a continuous constrained variational principle and continuous constrained Hamilton's equations in the Type III Hamiltonian case:

Theorem 2.11 ([Duruisseaux and Leok, 2022a])**.** *Consider the* ***Type III constrained action functional*** $\mathfrak{S} : C^2([0,T], T^*\mathcal{Q} \times \Lambda) \to \mathbb{R}$ *given by*

$$\mathfrak{S}(q(\cdot), p(\cdot), \lambda(\cdot)) = -p(0)q(0) - \int_0^T \left[p(t)\dot{q}(t) - H(q(t), p(t)) - \langle \lambda(t), C(q(t))\rangle\right] dt. \quad (2.82)$$

The condition that the action functional $\mathfrak{S}(q(\cdot), p(\cdot), \lambda(\cdot))$ *is stationary with respect to the boundary conditions* $\delta q(T) = 0$ *and* $\delta p(0) = 0$ *is equivalent to* $(q(\cdot), p(\cdot), \lambda(\cdot))$ *satisfying* ***Hamilton's constrained equations***

$$\dot{q} = \frac{\partial H}{\partial p}(q,p), \qquad \dot{p} = -\frac{\partial H}{\partial q}(q,p) - \langle \lambda, \nabla C(q)\rangle, \qquad C(q) = 0. \quad (2.83)$$

Proof. See Appendix A.2.3. □

Remark 2.3. *Hamilton's constrained equations are the same in the Type II and Type III formulations of Hamiltonian mechanics, and they can be thought of as the Hamilton's equations generated by the augmented Hamiltonian*

$$\bar{H}(q, \lambda, p, \mathfrak{p}) = H(q,p) + \langle \lambda, C(q)\rangle, \quad (2.84)$$

where $\mathfrak{p}$ *is the conjugate momentum for the variable* λ*. Furthermore, they are equivalent to the constrained Euler–Lagrange equations* (2.70)*, provided that the Lagrangian* L *is hyperregular.*

Remark 2.4. *It is sometimes beneficial to augment the continuous Hamilton's constrained equations with the equation* $\langle \frac{\partial H}{\partial p}(q,p), \nabla C(q)\rangle = 0$*, (and analogously for the discrete case) to ensure that the momentum* p *lies in the cotangent space to the manifold, as explained and illustrated in [Hairer et al., 2006, Chapter VII].*

We now generalize Theorem 2.2 from [Leok and Zhang, 2011] for the Type II case and its Type III analogue to include holonomic constraints $C(q)$ using Lagrange multipliers $\lambda : [0,T] \to \Lambda$.

In the Type II case, consider the function $\mathcal{S}(q_0, p_T)$ given by the extremal value of the constrained action functional $\mathfrak{S}$ over the family of curves $(q(\cdot), p(\cdot), \lambda(\cdot))$ satisfying the boundary conditions $q(0) = q_0$ and $p(T) = p_T$:

$$\mathcal{S}(q_0, p_T) \;=\; \underset{\substack{(q,p,\lambda)\in C^2([0,T],\, T^*\mathcal{Q}\times\Lambda) \\ q(0)=q_0,\quad p(T)=p_T}}{\operatorname{ext}} \mathfrak{S}(q(\cdot), p(\cdot), \lambda(\cdot)). \tag{2.85}$$

The following theorem shows that $\mathcal{S}(q_0, p_T)$ is a generating function for the flow of the continuous constrained Hamilton's equations:

Theorem 2.12 ([Duruisseaux and Leok, 2022a])**.** *The exact time-T flow map of Hamilton's equations $(q_0, p_0) \mapsto (q_T, p_T)$ is implicitly given by the following relations:*

$$q_T = D_2\mathcal{S}(q_0, p_T), \qquad p_0 = D_1\mathcal{S}(q_0, p_T). \tag{2.86}$$

In particular, $\mathcal{S}(q_0, p_T)$ is a Type II generating function that generates the exact flow of the constrained Hamilton's equations (2.81)*.*

Proof. See Appendix A.2.5. □

In the Type III case, consider the function $\mathcal{S}(q_T, p_0)$ given by the extremal value of the constrained action functional $\mathfrak{S}$ over the family of curves $(q(\cdot), p(\cdot), \lambda(\cdot))$ satisfying the boundary conditions $q(T) = q_T$ and $p(0) = p_0$:

$$\mathcal{S}(q_T, p_0) \;=\; \underset{\substack{(q,p,\lambda)\in C^2([0,T],\, T^*\mathcal{Q}\times\Lambda) \\ q(T)=q_T,\quad p(0)=p_0}}{\operatorname{ext}} \mathfrak{S}(q(\cdot), p(\cdot), \lambda(\cdot)). \tag{2.87}$$

The following theorem shows that $\mathcal{S}(q_T, p_0)$ is a generating function for the flow of the continuous constrained Hamilton's equations:

Theorem 2.13 ([Duruisseaux and Leok, 2022a])**.** *The exact time-T flow map of the Hamilton's equations $(q_0, p_0) \mapsto (q_T, p_T)$ is implicitly given by the following relations:*

$$q_0 = -D_2\mathcal{S}(q_T, p_0), \qquad p_T = -D_1\mathcal{S}(q_T, p_0). \tag{2.88}$$

In particular, $\mathcal{S}(q_T, p_0)$ is a Type III generating function that generates the exact flow of the constrained Hamilton's equations (2.83)*.*

Proof. See Appendix A.2.6. □

Discrete Constrained Variational Hamiltonian Mechanics

Let us now extend the results of [Leok and Zhang, 2011] to introduce a discrete formulation of variational Hamiltonian mechanics which includes holonomic constraints. Suppose we are given a partition $0 = t_0 < t_1 < \ldots < t_N = T$ of the interval $[0, T]$, and a discrete curve $\{(q_k, p_k, \lambda_k)\}_{k=0}^N$ in $T^*\mathcal{Q} \times \Lambda$ such that $q_k \approx q(t_k)$, $p_k \approx p(t_k)$ and $\lambda_k \approx \lambda(t_k)$. We formulate discrete constrained variational Hamiltonian mechanics in terms of the following discrete analogues of the constrained action functional $\mathfrak{S}$ given by equation (2.80):

$$\mathfrak{S}_d^+\left(\{(q_k, p_k, \lambda_k)\}_{k=0}^N\right) = p_N q_N - \sum_{k=0}^{N-1} \left[p_{k+1} q_{k+1} - H_d^+(q_k, p_{k+1}) - \langle \lambda_k, \mathcal{C}(q_k)\rangle\right], \tag{2.89}$$

$$\mathfrak{S}_d^-\left(\{(q_k, p_k, \lambda_k)\}_{k=0}^N\right) = -p_0 q_0 - \sum_{k=0}^{N-1} \left[-p_k q_k - H_d^-(q_{k+1}, p_k) - \langle \lambda_{k+1}, \mathcal{C}(q_{k+1})\rangle\right], \tag{2.90}$$

where

$$H_d^+(q_k, p_{k+1}) \approx \operatorname*{ext}_{\substack{(q,p,\lambda)\in C^2([t_k,t_{k+1}],T^*\mathcal{Q}\times\Lambda)\\ q(t_k)=q_k,\quad p(t_{k+1})=p_{k+1}}} p(t_{k+1})q(t_{k+1}) - \int_{t_k}^{t_{k+1}} \left[p(t)\dot{q}(t) - H(q(t), p(t))\right] dt \tag{2.91}$$

$$H_d^-(q_{k+1}, p_k) \approx \operatorname*{ext}_{\substack{(q,p,\lambda)\in C^2([t_k,t_{k+1}],T^*\mathcal{Q}\times\Lambda)\\ q(t_{k+1})=q_{k+1},\quad p(t_k)=p_k}} -p(t_k)q(t_k) - \int_{t_k}^{t_{k+1}} \left[p(t)\dot{q}(t) - H(q(t), p(t))\right] dt. \tag{2.92}$$

We can now derive a discrete analogue of Theorem 2.10 which relates a Type II discrete variational principle to discrete constrained Hamilton's equations, generalizing Lemma 3.1 from [Leok and Zhang, 2011]:

Theorem 2.14 ([Duruisseaux and Leok, 2022a])**.** *The Type II discrete Hamilton's phase space variational principle*

$$\delta\mathfrak{S}_d^+\left(\{(q_k, p_k, \lambda_k)\}_{k=0}^N\right) = 0 \tag{2.93}$$

is equivalent to the ***discrete constrained right Hamilton's equations***

$$\boxed{q_{k+1} = D_2 H_d^+(q_k, p_{k+1}), \quad p_k = D_1 H_d^+(q_k, p_{k+1}) + \langle \lambda_k, \nabla \mathcal{C}(q_k)\rangle, \quad \mathcal{C}(q_k) = 0,} \tag{2.94}$$

where $H_d^+(q_k, p_{k+1})$ is defined via equation (2.91)*.*

Proof. See Appendix A.2.8. □

Similarly, we can obtain a discrete analogue of Theorem 2.11 which relates a Type III discrete variational principle to discrete constrained Hamilton's equations:

Theorem 2.15 ([Duruisseaux and Leok, 2022a])**.** *The Type III discrete Hamilton's phase space variational principle*

$$\delta\mathfrak{S}_d^-\left(\{(q_k,p_k,\lambda_k)\}_{k=0}^N\right)=0 \tag{2.95}$$

is equivalent to the ***discrete constrained left Hamilton's equations***

$$\boxed{q_k=-D_2H_d^-(q_{k+1},p_k),\quad p_{k+1}=-D_1H_d^-(q_{k+1},p_k)-\langle\lambda_{k+1},\nabla\mathcal{C}(q_{k+1})\rangle,\quad \mathcal{C}(q_k)=0,} \tag{2.96}$$

where $H_d^-(q_{k+1},p_k)$ *is defined via equation* (2.92)*.*

Proof. See Appendix A.2.9. □

Remark 2.5. *These discrete constrained Hamilton's equations can be thought of as the discrete Hamilton's equations generated by the augmented discrete Hamiltonians*

$$\bar{H}_d^+\left((q_k,\lambda_k),(p_{k+1},p_{k+1})\right) \;=\; H_d^+(q_k,p_{k+1})+\langle\lambda_k,\mathcal{C}(q_k)\rangle, \tag{2.97}$$

and

$$\bar{H}_d^-\left((q_{k+1},\lambda_{k+1}),(p_k,p_k)\right) \;=\; H_d^-(q_k,p_{k+1})+\langle\lambda_{k+1},\mathcal{C}(q_{k+1})\rangle. \tag{2.98}$$

This augmented Hamiltonian perspective together with the augmented Lagrangian perspective described in Remark 2.2 imply that the constrained $\bar{H}_d^+$ *variational integrator is equivalent to the constrained* $\bar{L}_d^+$ *variational integrator whenever the* H_d^+ *variational integrator is equivalent to the* L_d^+ *variational integrator (and similarly for the constrained variational integrators generated by the* $\bar{H}_d^-$ *and* $\bar{L}_d^-$ *variational integrators). Examples where this happens are presented in [Schmitt et al., 2018] for Taylor variational integrators provided the Lagrangian is hyperregular, and in [Leok and Zhang, 2011] for generalized Galerkin variational integrators using the same choices of basis functions and numerical quadrature formula provided the Hamiltonian is hyperregular.*

2.6.3 Error Analysis for Constrained Variational Integrators

Recall Theorem 2.3.1 from [Marsden and West, 2001] which states that if a discrete Lagrangian, $L_d : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}$, approximates the exact discrete Lagrangian $L_d^E : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}$ to order r, i.e.,

$$L_d(q_0, q_h) = L_d^E(q_0, q_h) + \mathcal{O}(h^{r+1}), \tag{2.99}$$

then the discrete Hamiltonian map $\tilde{F}_{L_d} : (q_k, p_k) \mapsto (q_{k+1}, p_{k+1})$, viewed as a one-step method defined implicitly from the discrete Euler–Lagrange equations

$$D_1 L_d(q_k, q_{k+1}) + D_2 L_d(q_{k-1}, q_k) = 0, \tag{2.100}$$

has order of accuracy r.

Recall as well Theorem 2.2 from [Schmitt and Leok, 2017] which states that if a discrete right Hamiltonian H_d^+ approximates the exact discrete right Hamiltonian $H_d^{+,E}$ to order r, i.e.,

$$H_d^+(q_0, p_h) = H_d^{+,E}(q_0, p_h) + \mathcal{O}(h^{r+1}), \tag{2.101}$$

then the discrete right Hamiltonian map $\tilde{F}_{H_d^+} : (q_k, p_k) \mapsto (q_{k+1}, p_{k+1})$, viewed as a one-step method defined implicitly by the discrete right Hamilton's equations

$$p_k = D_1 H_d^+(q_k, p_{k+1}), \qquad q_{k+1} = D_2 H_d^+(q_k, p_{k+1}), \tag{2.102}$$

is order r accurate.

Similarly, if a discrete left Hamiltonian H_d^- approximates the exact discrete left Hamiltonian $H_d^{-,E}$ to order r, i.e.,

$$H_d^-(q_1, p_0) = H_d^{-,E}(q_1, p_0) + \mathcal{O}(h^{r+1}), \tag{2.103}$$

then the discrete left Hamiltonian map $\tilde{F}_{H_d^-} : (q_k, p_k) \mapsto (q_{k+1}, p_{k+1})$, viewed as a one-step method defined implicitly by the discrete left Hamilton's equations

$$p_{k+1} = -D_1 H_d^-(q_{k+1}, p_k), \qquad q_k = -D_2 H_d^-(q_{k+1}, p_k), \tag{2.104}$$

is order r accurate.

We will now exploit these error analysis results to derive analogous results for the constrained versions of Hamiltonian and Lagrangian variational integrators discussed in Sections 2.6.1 and 2.6.2.

For the Lagrangian case, we can think of the Lagrange multipliers λ as extra position coordinates and define an augmented Lagrangian $\bar{L}$ via

$$\bar{L}\left((q,\lambda),(\dot{q},\dot{\lambda})\right) \;=\; L(q,\dot{q}) - \langle \lambda, \mathcal{C}(q)\rangle. \tag{2.105}$$

A corresponding augmented discrete Lagrangian is given by

$$\bar{L}_d\left((q_k,\lambda_k),(q_{k+1},\lambda_{k+1})\right) \;=\; L_d(q_k,q_{k+1}) - \langle \lambda_k, \mathcal{C}(q_k)\rangle, \tag{2.106}$$

and the discrete Euler–Lagrange equations (2.100)

$$D_1\bar{L}_d\left((q_k,\lambda_k),(q_{k+1},\lambda_{k+1})\right) \;+\; D_2\bar{L}_d\left((q_{k-1},\lambda_{k-1}),(q_k,\lambda_k)\right) \;=\; 0, \tag{2.107}$$

yield the discrete constrained Euler–Lagrange equations

$$D_1L_d(q_k,q_{k+1}) + D_2L_d(q_{k-1},q_k) = \langle \lambda_k, \nabla\mathcal{C}(q_k)\rangle, \qquad \mathcal{C}(q_k) = 0, \tag{2.108}$$

derived in Section 2.6.1.

As a consequence, we can apply Theorem 2.3.1 of [Marsden and West, 2001] to the augmented Lagrangian (2.105) and obtain the following result:

Theorem 2.16 ([Duruisseaux and Leok, 2022a])**.** *Suppose that for an exact discrete Lagrangian L_d^E and a discrete Lagrangian L_d,*

$$L_d(q_0,q_h) - \langle \lambda_0, \mathcal{C}(q_0)\rangle \;=\; L_d^E(q_0,q_h) - \int_0^h \langle \lambda(t), \mathcal{C}(q(t))\rangle dt + \mathcal{O}(h^{r+1}). \tag{2.109}$$

Then, the discrete map

$$(q_k,p_k,\lambda_k) \mapsto (q_{k+1},p_{k+1},\lambda_{k+1}), \tag{2.110}$$

viewed as a one-step method defined implicitly by the discrete constrained Euler–Lagrange equations, has order of accuracy r.

For the Hamiltonian case, we can think of the Lagrange multipliers λ as extra position coordinates and define conjugate momenta ρ, which are constants of motion since the time-derivative of the Lagrange multiplier λ does not appear anywhere, and are constrained to be zero.

The augmented Hamiltonian $\bar{H}$, given by

$$\bar{H}\left((q,\lambda),(p,\rho)\right) \;=\; H(q,p) + \langle \lambda, \mathcal{C}(q)\rangle, \tag{2.111}$$

yields the augmented left Hamiltonian

$$\bar{H}_d^-\left((q_{k+1},\lambda_{k+1}),(p_k,\rho_k)\right) \;=\; H_d^-(q_{k+1},p_k) + \langle \lambda_{k+1}, \mathcal{C}(q_{k+1})\rangle, \tag{2.112}$$

and the augmented right discrete Hamiltonian

$$\bar{H}_d^+\left((q_k,\lambda_k),(p_{k+1},\rho_{k+1})\right) \;=\; H_d^+(q_k,p_{k+1}) + \langle \lambda_k, \mathcal{C}(q_k)\rangle. \tag{2.113}$$

The corresponding discrete left Hamilton's equations

$$\begin{cases} (p_{k+1},\rho_{k+1}) = -D_1\bar{H}_d^-\left((q_{k+1},\lambda_{k+1}),(p_k,\rho_k)\right) \\ (q_k,\lambda_k) = -D_2\bar{H}_d^-\left((q_{k+1},\lambda_{k+1}),(p_k,\rho_k)\right) \end{cases} \tag{2.114}$$

and discrete right Hamilton's equations

$$\begin{cases} (p_k,\rho_k) = D_1\bar{H}_d^+\left((q_k,\lambda_k),(p_{k+1},\rho_{k+1})\right) \\ (q_{k+1},\lambda_{k+1}) = D_2\bar{H}_d^+\left((q_k,\lambda_k),(p_{k+1},\rho_{k+1})\right) \end{cases} \tag{2.115}$$

yield the discrete constrained left Hamilton's equations

$$q_k = -D_2H_d^-(q_{k+1},p_k), \quad p_{k+1} = -D_1H_d^-(q_{k+1},p_k) - \langle \lambda_{k+1}, \nabla\mathcal{C}(q_{k+1})\rangle, \quad \mathcal{C}(q_k) = 0, \tag{2.116}$$

and the discrete constrained right Hamilton's equations

$$q_{k+1} = D_2H_d^+(q_k,p_{k+1}), \quad p_k = D_1H_d^+(q_k,p_{k+1}) + \langle \lambda_k, \nabla\mathcal{C}(q_k)\rangle, \quad \mathcal{C}(q_k) = 0, \tag{2.117}$$

derived in Section 2.6.2.

As a consequence, we can apply Theorem 2.2 in [Schmitt and Leok, 2017] for Type II unconstrained variational integrators to the augmented Hamiltonian and obtain the following result:

Theorem 2.17 ([Duruisseaux and Leok, 2022a])**.** *Suppose that given an exact discrete right Hamiltonian $H_d^{+,E}$ and a discrete right Hamiltonian H_d^+, we have*

$$H_d^+(q_0,p_h)+\langle\lambda_0,\mathcal{C}(q_0)\rangle \;=\; H_d^{+,E}(q_0,p_h)+\int_0^h\langle\lambda(t),\mathcal{C}(q(t))\rangle dt+\mathcal{O}(h^{r+1}). \tag{2.118}$$

Then, the discrete map

$$(q_k,p_k,\lambda_k)\mapsto(q_{k+1},p_{k+1},\lambda_{k+1}), \tag{2.119}$$

viewed as a one-step method defined implicitly by the discrete constrained right Hamilton's equations, has order of accuracy r.

Similarly, we can apply the analogue of Theorem 2.2 in [Schmitt and Leok, 2017] for Type III unconstrained variational integrators to the augmented Hamiltonian and obtain the following result:

Theorem 2.18 ([Duruisseaux and Leok, 2022a])**.** *Suppose that given an exact discrete left Hamiltonian $H_d^{-,E}$ and a discrete left Hamiltonian H_d^-, we have*

$$H_d^-(q_h,p_0)+\langle\lambda_h,\mathcal{C}(q_h)\rangle \;=\; H_d^{-,E}(q_h,p_0)+\int_0^h\langle\lambda(t),\mathcal{C}(q(t))\rangle dt+\mathcal{O}(h^{r+1}). \tag{2.120}$$

Then, the discrete map

$$(q_k,p_k,\lambda_k)\mapsto(q_{k+1},p_{k+1},\lambda_{k+1}), \tag{2.121}$$

viewed as a one-step method defined implicitly by the discrete constrained left Hamilton's equations, has order of accuracy r.

2.7 Symplectic Integrators with Variable Time-steps

We will now discuss how prescribed variable time-steps can be incorporated in symplectic integrators.

The use of prescribed variable time-steps is motivated by the observation that the global error estimates for a numerical method depend in part on the maximum local truncation error, and this in turn is related to both the time-step and the magnitude of the $(r+1)$-derivatives of the solution of a r-order numerical method. For a fixed number of time-steps, the maximum local truncation error is minimized if the local truncation error is equidistributed over the time intervals. In turn, this can be achieved if, for example, the time-step is chosen to be an appropriate function of the reciprocal of the relevant derivative of the solution. This derivative can be estimated *a posteriori* by comparing methods with different orders of accuracy, or methods with the same order of accuracy but different error constants. Alternatively, in the Kepler 2-body problem, for example, Kepler's second law states that the line joining the planet and the Sun sweeps out equal areas during equal intervals of time, so the angular velocity of the planet is proportional to the reciprocal of the radius squared, which gives an *a priori* bound. In essence, variable time-steps are chosen to control the error incurred at each time-step, which in turn affects the global accuracy of the numerical trajectory.

As discussed earlier in Section 2.3, symplectic integrators form a class of geometric numerical integrators of interest since, when applied to Hamiltonian systems, they yield discrete approximations of the flow that preserve the symplectic 2-form. The preservation of symplecticity results in the preservation of many qualitative aspects of the underlying dynamical system. In particular, when applied to conservative Hamiltonian systems, symplectic integrators show excellent long-time near-energy preservation.

However, when symplectic integrators were first used in combination with variable time-steps, the near-energy preservation was lost and the integrators performed poorly (see [Gladman et al., 1991; Calvo and Sanz-Serna, 1993]). Backward error analysis provided justification both for the excellent long-time near-energy preservation of symplectic

integrators and for the poor performance experienced when using variable time-steps (see Chapter IX of [Hairer et al., 2006]). We saw in Section 2.3 using backward error analysis that symplectic integrators can be associated with a modified Hamiltonian in the form of a powers series in terms of the time-step. The use of a variable time-step results in a different modified Hamiltonian at every iteration where the time-step is changed, which is the source of the poor energy conservation.

There has been a great effort to circumvent this problem, and there have been many successes. However, there has yet to be a unified general framework for constructing adaptive symplectic integrators. We contribute to this effort in this section by demonstrating how variational integrators can be used to systematically construct symplectic integrators that allow for the use of prescribed variable time-steps, both on the Hamiltonian and Lagrangian sides.

In this section, we will first review a mechanism for variable time-stepping, the Poincaré transformation, based on an autonomous monitor function on the Hamiltonian side, and then extend it to time-dependent monitor functions and provide a novel variational derivation for the Poincaré transformation. We will then combine this approach with Hamiltonian variational integrators and derive corresponding error analysis results. We will then construct a framework for variable time-stepping on the Lagrangian side which mimics the Poincaré transformation.

The resulting integrators will prove very useful later on in this dissertation when we design efficient geometric integrators with variable time-steps based on a precise time-rescaling for accelerated optimization.

In this section, we will consider a time reparameterization $t \mapsto \tau$, given explicitly by a monitor function

$$\frac{dt}{d\tau} = g(q, t, p), \tag{2.122}$$

and we will refer to t and τ as the physical and fictive times, respectively. We will use dots on top of variables to denote derivatives with respect to physical time t, and apostrophes to denote derivatives with respect to fictive time τ.

2.7.1 Hamiltonian Integrators with Prescribed Variable Time-steps

We now demonstrate how to construct symplectic integrators that allow for the use of a prescribed variable time-steps, on the Hamiltonian side.

The Poincaré Transformation

The Poincaré transformation is one way to incorporate variable time-steps in geometric integrators without losing the nice conservation properties associated with these integrators. This transformation for time-adaptive symplectic integrators on the Hamiltonian side was first introduced in [Zare and Szebehely, 1975] in the case where the monitor function $g(q,p)$ is autonomous.

Given an autonomous Hamiltonian $H(q,p)$, and a desired transformation of time $t \mapsto \tau$ described by the **monitor function** $g(q,p)$ via

$$\frac{dt}{d\tau} = g(q,p), \tag{2.123}$$

a new Hamiltonian system is constructed using the **Poincaré transformation**,

$$\boxed{\bar{H}(\bar{q},\bar{p}) = g(q,p)\,(H(q,p) + \mathfrak{p})\,,} \tag{2.124}$$

in the extended phase space defined by

$$\bar{q} = \begin{bmatrix} q \\ \mathfrak{q} \end{bmatrix} \in \bar{\mathcal{Q}} \qquad \text{and} \qquad \bar{p} = \begin{bmatrix} p \\ \mathfrak{p} \end{bmatrix} \tag{2.125}$$

where $\mathfrak{p} = -H(q(0),p(0))$ is the conjugate momentum for $\mathfrak{q} = t$, so that $\bar{H}(\bar{q},\bar{p}) = 0$ along all integral curves through $(\bar{q}(0),\bar{p}(0))$. The time t shall be referred to as the physical time, while τ will be referred to as the fictive time.

We can then use a symplectic integrator with constant time-step in fictive time τ on the Poincaré transformed Hamiltonian system, which will have the effect of integrating the original Hamiltonian system with the desired variable time-step in physical time t via the relation $\frac{dt}{d\tau} = g(q,p)$. Indeed, the Poincaré transformed Hamiltonian is chosen in such a way that the corresponding component dynamics satisfy Hamilton's equations in the original space.

Note that in general, along an integral curve through $(\bar{q}(0), \bar{p}(0))$,

$$\frac{\partial^2 \bar{H}}{\partial \bar{p}^2} = \begin{bmatrix} \frac{\partial H}{\partial p} \nabla_p g(q,p)^\top + g(q,p) \frac{\partial^2 H}{\partial p^2} + \nabla_p g(q,p) \frac{\partial H}{\partial p}^\top & \nabla_p g(q,p) \\ \nabla_p g(q,p)^\top & 0 \end{bmatrix}, \tag{2.126}$$

which can become singular for many choices of initial Hamiltonian H and time-rescaling monitor function g.

This Poincaré framework can also be extended to the case where the original Hamiltonian H and the chosen monitor function g depend explicitly on time t, based on ideas from [Hairer, 1997]. Given a time-dependent Hamiltonian $H(q,t,p)$, consider a desired transformation of time $t \mapsto \tau$, given by the monitor function

$$\frac{dt}{d\tau} = g(q,t,p). \tag{2.127}$$

Then, consider the new Hamiltonian system given by the Poincaré transformation

$$\boxed{\bar{H}(\bar{q},\bar{p}) = g(q,\mathfrak{q},p)\,(H(q,\mathfrak{q},p) + \mathfrak{p})} \tag{2.128}$$

in the extended phase space defined by

$$\bar{q} = \begin{bmatrix} q \\ \mathfrak{q} \end{bmatrix} \in \bar{\mathcal{Q}} \qquad \text{and} \qquad \bar{p} = \begin{bmatrix} p \\ \mathfrak{p} \end{bmatrix} \tag{2.129}$$

where $\mathfrak{p}$ is the conjugate momentum for $\mathfrak{q} = t$ and satisfies $\mathfrak{p}(0) = -H(q(0), 0, p(0))$. The corresponding equations of motion in the extended phase space are then given by

$$\bar{q}' = \frac{\partial \bar{H}}{\partial \bar{p}}, \qquad \bar{p}' = -\frac{\partial \bar{H}}{\partial \bar{q}}. \tag{2.130}$$

As before, we can then use a symplectic integrator with constant time-step in fictive time τ on the Poincaré transformed Hamiltonian system, which will have the effect of integrating the original Hamiltonian system with the desired variable time-step in physical time t via the relation $\frac{dt}{d\tau} = g(q,t,p)$.

More precisely, suppose that the curve $(\bar{Q}(\tau), \bar{P}(\tau))$ is a solution to these extended Hamilton's equations of motion, and let $(q(t), p(t))$ solve Hamilton's equations for the original Hamiltonian H. Then

$$\bar{H}(\bar{Q}(\tau), \bar{P}(\tau)) = \bar{H}(\bar{Q}(0), \bar{P}(0)) = 0. \tag{2.131}$$

Therefore, the components $(Q(\tau), P(\tau))$ in the original phase space of the augmented solutions $(\bar{Q}(\tau), \bar{P}(\tau))$ satisfy

$$H(Q(\tau), \tau, P(\tau)) = -\mathfrak{p}(\tau), \qquad H(Q(0), 0, P(0)) = -\mathfrak{p}(0) = H(q(0), 0, p(0)). \tag{2.132}$$

Then, $(Q(\tau), P(\tau))$ and $(q(t), p(t))$ both satisfy Hamilton's equations for the original Hamiltonian H with the same initial values, so they must be the same.

As before, the matrix

$$\frac{\partial^2 \bar{H}}{\partial \bar{p}^2} = \begin{bmatrix} \frac{\partial H}{\partial p} \nabla_p g(\bar{q}, p)^\top + g(\bar{q}, p) \frac{\partial^2 H}{\partial p^2} + \nabla_p g(\bar{q}, p) \frac{\partial H}{\partial p}^\top & \nabla_p g(\bar{q}, p) \\ \nabla_p g(\bar{q}, p)^\top & 0 \end{bmatrix}, \tag{2.133}$$

will be singular in many cases.

Most of the prior literature on variable time-step symplectic integrators cited in this dissertation focuses exclusively on monitor functions g that only depend on position, in which case the Hessian matrix $\frac{\partial^2 \bar{H}}{\partial \bar{p}^2}$ is singular, and as a result the associated Legendre transformation, $\mathbb{F}\bar{H} : T^*Q \to TQ$ is noninvertible, and thus the resulting transformed Poincaré Hamiltonian is degenerate and there is no corresponding Lagrangian formulation. Therefore, the Type II and Type III Hamiltonian variational integrator frameworks are the most natural ways to derive variable time-step symplectic integrators using the Poincaé transformation.

The Poincaré Transformation and Hamiltonian Variational Integrators

Variational integrators provide a systematic method for constructing symplectic integrators of arbitrary order based on the discretization of Hamilton's principle [Marsden and West, 2001; Hall and Leok, 2015], or equivalently, by the approximation of generating functions. However, there has not been a systematic attempt to incorporate time-adaptivity into the setting of variational integrators. This is due to the fact that the Poincaré transformed Hamiltonian that is used is in general degenerate, so there is no corresponding Lagrangian analogue, which prevents the use of traditional variational integrators that are based on a Lagrangian formulation of mechanics and involve the construction of a discrete Lagrangian that approximates a Type I generating function given by Jacobi's solution of the Hamilton–Jacobi equation.

Instead, we propose the use of Hamiltonian variational integrators [Leok and Zhang, 2011], which are based on Type II and Type III generating functions that have no difficulty with this degeneracy. We develop an analogue of the methods derived using the Poincaré transformation framework but directly in terms of generating functions of symplectic maps. These prior results are based on symplectic (partitioned) Runge–Kutta methods, which are related to Type I generating functions [Suris, 1990], but we desire an explicit characterization of the flow maps of time-adaptive Hamiltonian systems so that we can employ the Hamiltonian variational integrator framework instead.

The exact Type II generating function for the Poincaré transformed Hamiltonian is given by

$$\bar{H}_d^{+,E}(\bar{q}_0,\bar{p}_1;h) \;=\; \bar{p}_1^\top \bar{q}_1 - \int_0^h \left(\bar{p}(\tau)^\top \dot{\bar{q}}(\tau) - \bar{H}(\bar{q}(\tau),\bar{p}(\tau))\right) d\tau, \tag{2.134}$$

where the curve $(\bar{q}(\tau),\bar{p}(\tau))$ satisfy the Hamilton's equations associated with the Poincaré transformed Hamiltonian $\bar{H}$, with boundary conditions $\bar{q}(0) = \bar{q}_0$ and $\bar{p}(h) = \bar{p}_1$. This exact discrete right Hamiltonian implicitly defines a symplectic map with respect to the symplectic form $\bar{\omega}(\bar{p}_k,\bar{q}_k)$ on $T^*\bar{Q}$ via the discrete Legendre transforms given by

$$\bar{p}_0 = \frac{\partial \bar{H}_d^{+,E}}{\partial \bar{q}_0}, \qquad \bar{q}_1 = \frac{\partial \bar{H}_d^{+,E}}{\partial \bar{p}_1}. \tag{2.135}$$

Similarly, the exact Type III generating function for the Poincaré transformed Hamiltonian is given by

$$\bar{H}_d^{-,E}(\bar{q}_0, \bar{p}_1; h) \;=\; -\bar{p}_0^\top \bar{q}_0 - \int_0^h \left(\bar{p}(\tau)^\top \dot{\bar{q}}(\tau) - \bar{H}(\bar{q}(\tau), \bar{p}(\tau))\right) d\tau, \tag{2.136}$$

where the curve $(\bar{q}(\tau), \bar{p}(\tau))$ satisfy the Hamilton's equations associated with the Poincaré transformed Hamiltonian $\bar{H}$ with boundary conditions $\bar{q}(h) = \bar{q}_1$ and $\bar{p}(0) = \bar{p}_0$. This exact discrete left Hamiltonian implicitly defines a symplectic map with respect to the symplectic form $\bar{\omega}(\bar{p}_k, \bar{q}_k)$ on $T^*\bar{Q}$ via the discrete Legendre transforms given by

$$\bar{p}_1 = -\frac{\partial \bar{H}_d^{-,E}}{\partial \bar{q}_1}, \qquad\qquad \bar{q}_0 = -\frac{\partial \bar{H}_d^{-,E}}{\partial \bar{p}_0}. \tag{2.137}$$

Our approach is to construct Hamiltonian variational integrators using a discrete Hamiltonian $\bar{H}_d^{\pm}$ that approximates the corresponding exact discrete Hamiltonian $\bar{H}_d^{\pm,E}$ to order r. The resulting integrator will be symplectic with constant time-step in fictive time τ and more importantly with the desired variable time-step in physical time t via $\frac{dt}{d\tau} = g(q, t, p)$. It is important to note that this method will be symplectic in two different ways. It will be symplectic both with respect to the symplectic form $\mathbf{d}\bar{p} \wedge \mathbf{d}\bar{q}$ and with respect to the symplectic form $\mathbf{d}p \wedge \mathbf{d}q$. Since the derivative of $\mathfrak{p}$ is 0, $\mathfrak{p}$ is constant and $\mathbf{d}\mathfrak{p}_k \wedge \mathbf{d}\mathfrak{q}_k = 0$, so the symplectic form in generalized coordinates is given by

$$\bar{\omega}(\bar{p}_k, \bar{q}_k) \;=\; \mathbf{d}\bar{p}_k \wedge \mathbf{d}\bar{q}_k \;=\; \sum_{i=1}^{n+1} \mathbf{d}\bar{p}_{k,i} \wedge \mathbf{d}\bar{q}_{k,i} \;=\; \sum_{i=1}^{n} \mathbf{d}p_{k,i} \wedge \mathbf{d}q_{k,i} \;=\; \omega(p_k, q_k). \tag{2.138}$$

A symplectic integrator with variable time-stepping was proposed independently in [Hairer, 1997] and [Reich, 1999], which applied a symplectic integrator to the Poincaré transformed Hamiltonian. In [Hairer, 1997], it is noted that one of the first applications of the Poincaré transformation framework was by Levi–Civita, who applied it to the three-body problem. A more in-depth discussion of such time transformations can be found in [Struckmeier, 2005]. Further work using this type of transformation has been published, such as [Blanes and Budd, 2004; Blanes and Iserles, 2012], which focused on developing symplectic, explicit, splitting methods with variable time-steps.

The novelty of our approach consists in discretizing the Type II or Type III generating function for the flow of Hamilton's equations, where the Hamiltonian is given by the Poincaré transformation. Thus, we are constructing variational integrators, and in particular Hamiltonian variational integrators [Lall and West, 2006; Leok and Zhang, 2011]. The use of Type II or Type III integrators is justified by the degeneracy of the Hamiltonian, which implies that there is no corresponding Type I Lagrangian formulation. This approach works seamlessly with existing methods and theorems of Hamiltonian variational integrators, but now the system under consideration is the transformed Hamiltonian system resulting from the Poincaré transformation. The methods presented in [Hairer, 1997; Reich, 1999] include the possibility of applying a given variational integrator to the transformed differential equations. Our approach gives a framework for constructing variational integrators at the level of the generating function by using the Poincaré transformed discrete right Hamiltonian.

Remark 2.6. *Other approaches to variable time-step variational integrators can be found in [Kane et al., 1999; Modin and Führer, 2006; Nair, 2012]. In particular, the variational integrator in [Kane et al., 1999] is inspired by the result of [Ge and Marsden, 1988], which states that constant time-step symplectic integrators of autonomous Hamiltonian systems cannot exactly conserve the energy unless it agrees with the exact flow map up to a time reparametrization. As a result, they sought a variable time-step energy-conserving symplectic integrator in an expanded nonautonomous system. However, symplecticity is with respect to the space-time symplectic form* $\mathbf{d}p \wedge \mathbf{d}q + \mathbf{d}H \wedge \mathbf{d}t$. *The time-step is determined by enforcing discrete energy conservation, which arises as a consequence of the fact that energy is the Noether quantity associated with time translational symmetry. An extended Hamiltonian is used, similar in spirit to the Poincaré transformation. An approach that builds off this idea and space-time symplecticity was presented in [Nair, 2012], and a less constrained choice of time-step was allowed. In [Modin and Führer, 2006], adaptive variational integrators are constructed using a transformation of the Lagrangian, which is motivated by the Poincaré transformation, but it is not equivalent. The lack of equivalence is not surprising, since the Poincaré transformed Hamiltonian is degenerate for their choice of monitor functions. As a consequence, the phase space path is not preserved.*

Variational Error Analysis

The standard error analysis for Hamiltonian variational integrators assumes a nondegenerate Hamiltonian, i.e., $\det\left(\frac{\partial^2 \bar{H}}{\partial \bar{p}^2}\right) \neq 0$ (see [Schmitt and Leok, 2017]), which might not be the case for the Poincaré transformed Hamiltonian. The nondegeneracy of the Hamiltonian ensures that we can apply the usual implicit function theorem to the discrete Hamilton's equations, and the proof of the standard error analysis theorem relies upon the following result:

Lemma 2.1 ([Schmitt and Leok, 2017])**.** *Let $f_1, g_1, e_1, f_2, g_2, e_2$ be r-times continuously differentiable functions such that*

$$f_1(x,h) = g_1(x,h) + h^{r+1}e_1(x,h), \qquad f_2(x,h) = g_2(x,h) + h^{r+1}e_2(x,h). \tag{2.139}$$

Then, there exist functions e_{12} and $\bar{e}_1$ bounded on compact sets such that

$$f_2(f_1(x,h),h) = g_2(g_1(x,h),h) + h^{r+1}e_{12}(g_1(x,h),h), \tag{2.140}$$

$$f_1^{-1}(y) = g_1^{-1}(y) + h^{r+1}\bar{e}_1(y). \tag{2.141}$$

Given a discrete Hamiltonian $H_d^{\pm}$, we introduce the **discrete fiber derivatives** (or **discrete Legendre transforms**), $\mathbb{F}^{\pm}H_d^{\pm}$:

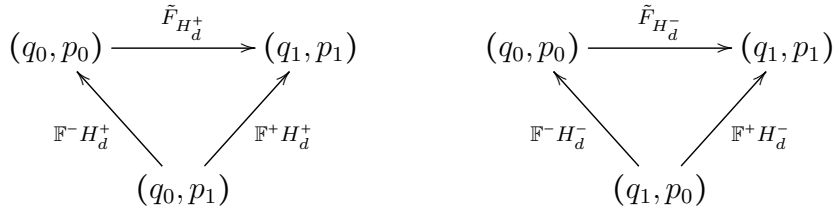
$$\mathbb{F}^{+}H_d^{+}(q_0,p_1) \;:\; (q_0,p_1) \;\mapsto\; (D_2H_d^{+}(q_0,p_1),p_1), \tag{2.142}$$

$$\mathbb{F}^{+}H_d^{-}(q_1,p_0) \;:\; (q_1,p_0) \;\mapsto\; (q_1,-D_1H_d^{-}(q_1,p_0)), \tag{2.143}$$

$$\mathbb{F}^{-}H_d^{+}(q_0,p_1) \;:\; (q_0,p_1) \;\mapsto\; (q_0,D_1H_d^{+}(q_0,p_1)), \tag{2.144}$$

$$\mathbb{F}^{-}H_d^{-}(q_1,p_0) \;:\; (q_1,p_0) \;\mapsto\; (-D_2H_d^{-}(q_1,p_0),p_0). \tag{2.145}$$

We observe that the following diagrams commute:



As such, the discrete left and right Hamiltonian maps can be expressed in terms of the discrete fiber derivatives,

$$\tilde{F}_{H_d^\pm}(q_0, p_0) \;=\; \mathbb{F}^+ H_d^\pm \;\circ\; (\mathbb{F}^- H_d^\pm)^{-1}(q_0, p_0) \;=\; (q_1, p_1), \tag{2.146}$$

and this observation together with Lemma 2.1 ensures that the order of accuracy of the integrator is at least of the order to which the discrete Hamiltonian $H_d^\pm$ approximates the exact discrete Hamiltonian $H_d^{\pm,E}$.

However, the Poincaré transformed Hamiltonian might be degenerate so we cannot apply the usual implicit function theorem, and we need to establish the invertibility of the discrete Legendre transform $\mathbb{F}^- H_d^\pm$ in a different way.

The strongest general result we have been able to establish involves the case where the original Hamiltonian is autonomous, i.e., $H = H(q,p)$, and nondegenerate, and the monitor function is autonomous as well. These assumptions hold for an interesting and useful class of problems, and we will show that the exact discrete left and right Hamiltonians can be reduced to a particular form and that the extended variables $\mathfrak{p}_1$ and $\mathfrak{q}_1$ can be solved for explicitly. As a result, the implicit function theorem is not needed with respect to these variables.

Hamilton's equations for the Poincaré transformed Hamiltonian

$$\bar{H}(\bar{q}, \bar{p}) = g(q,p)\,(H(q,p) + \mathfrak{p})\,, \tag{2.147}$$

are given by

$$\bar{q}' \;=\; \begin{bmatrix} \nabla_p g(q,p)(H(q,p)+\mathfrak{p}) + \frac{\partial H}{\partial p} g(q,p) \\ g(q,p) \end{bmatrix}, \tag{2.148}$$

$$\bar{p}' \;=\; -\begin{bmatrix} \nabla_q g(q,p)(H(q,p)+\mathfrak{p}) + \frac{\partial H}{\partial q} g(q,p) \\ 0 \end{bmatrix}. \tag{2.149}$$

Using these equations, the corresponding exact discrete Hamiltonians are of the form

$$\bar{H}_d^{+,E}(\bar{q}_0, \bar{p}_1; h) = p_1^\top q_1 + \mathfrak{p}_1 \mathfrak{q}_1 - \int_0^h \left[p(\tau)^\top q'(\tau) - g(q(\tau), p(\tau)) H(q(\tau), p(\tau))\right] d\tau, \tag{2.150}$$

$$\bar{H}_d^{-,E}(\bar{q}_1, \bar{p}_0; h) = -p_0^\top q_0 - \mathfrak{p}_0 \mathfrak{q}_0 - \int_0^h \left[p(\tau)^\top q'(\tau) - g(q(\tau), p(\tau)) H(q(\tau), p(\tau))\right] d\tau. \tag{2.151}$$

As a result, only one part of these exact discrete left and right Hamiltonians requires approximations of the extended variable $\mathfrak{q}$ and $\mathfrak{p}$. Furthermore, the derivative of $\mathfrak{p}$ is 0 so $\mathfrak{p}_1 = \mathfrak{p}_0$.

Now, let $\bar{H}_d^{\pm}$ be approximations to the exact discrete left and right Hamiltonians of the form

$$\bar{H}_d^+(\bar{q}_0, \bar{p}_1; h) = p_1^\top \hat{q}_1(q_0, p_1; h) + \mathfrak{p}_1 \hat{\mathfrak{q}}_1(\mathfrak{q}_0, q_0, p_1; h) - I_1(q_0, p_1; h), \tag{2.152}$$

$$\bar{H}_d^-(\bar{q}_1, \bar{p}_0; h) = -p_0^\top \hat{q}_0(q_1, p_0; h) - \mathfrak{p}_0 \hat{\mathfrak{q}}_0(\mathfrak{q}_1, q_1, p_0; h) - I_2(q_1, p_0; h), \tag{2.153}$$

where $\hat{\cdot}$ denotes an approximation and where $I_1(q_0, p_1; h)$ and $I_2(q_1, p_0; h)$ both approximate the integral

$$\int_0^h \left[p(\tau)^\top q'(\tau) - g(q(\tau), p(\tau)) H(q(\tau), p(\tau)) \right] d\tau. \tag{2.154}$$

Then, the discrete right Legendre transforms give the following relations for $\mathfrak{p}_1$ and $\mathfrak{q}_1$:

$$\begin{bmatrix} p_0 \\ \mathfrak{p}_0 \end{bmatrix} = \begin{bmatrix} \frac{\partial \hat{q}_1}{\partial q_0}^\top p_1 + \mathfrak{p}_1 \frac{\partial \hat{\mathfrak{q}}_1}{\partial q_0} - \frac{\partial I_1}{\partial q_0} \\ \frac{\partial \hat{\mathfrak{q}}_1}{\partial \mathfrak{q}_0} \mathfrak{p}_1 \end{bmatrix}, \qquad \begin{bmatrix} q_1 \\ \mathfrak{q}_1 \end{bmatrix} = \begin{bmatrix} \hat{q}_1 + \frac{\partial \hat{q}_1}{\partial p_1}^\top p_1 + \frac{\partial \hat{\mathfrak{q}}_1}{\partial p_1}^\top \mathfrak{p}_1 - \frac{\partial I_1}{\partial p_1} \\ \hat{\mathfrak{q}}_1 \end{bmatrix}. \tag{2.155}$$

Now, the analytic solution satisfies $\mathfrak{p}_1 = \mathfrak{p}_0$, so there is no need to approximate $\mathfrak{p}_1$. Therefore, $\frac{\partial \hat{\mathfrak{q}}_1}{\partial \mathfrak{q}_0} = 1$. The resulting two systems can both be solved by first setting $\mathfrak{p}_1 = \mathfrak{p}_0$, then implicitly solving for p_1 in terms of $(\mathfrak{q}_0, q_0, \mathfrak{p}_1, p_1)$, explicitly solving for q_1 and finally explicitly solving for $\mathfrak{q}_1$. Since p_1 is not determined by $\mathfrak{q}_1$, the implicit function theorem is simply needed for finding p_1. Thus, we need $\det\left(\frac{\partial^2 \bar{H}}{\partial p^2}\right) \neq 0$, and from equation (2.126), this is the same as $\det\left(\frac{\partial H}{\partial p} \nabla_p g(q,p)^\top + g(q,p) \frac{\partial^2 H}{\partial p^2} + \nabla_p g(q,p) \frac{\partial H}{\partial p}^\top\right) \neq 0$. Note that this holds for nondegenerate Hamiltonians H and p-independent monitor functions.

Similarly, the discrete left Legendre transforms give relations for $\mathfrak{p}_1$ and $\mathfrak{q}_1$:

$$\begin{bmatrix} p_1 \\ \mathfrak{p}_1 \end{bmatrix} = \begin{bmatrix} \frac{\partial \hat{q}_0}{\partial q_1}^\top p_0 + \mathfrak{p}_0 \frac{\partial \hat{\mathfrak{q}}_0}{\partial q_1} + \frac{\partial I_2}{\partial q_1} \\ \frac{\partial \hat{\mathfrak{q}}_0}{\partial \mathfrak{q}_1} \mathfrak{p}_0 \end{bmatrix}, \qquad \begin{bmatrix} q_0 \\ \mathfrak{q}_0 \end{bmatrix} = \begin{bmatrix} \hat{q}_0 + \frac{\partial \hat{q}_0}{\partial p_0}^\top p_0 + \frac{\partial \hat{\mathfrak{q}}_0}{\partial p_0}^\top \mathfrak{p}_0 + \frac{\partial I_2}{\partial p_0} \\ \hat{\mathfrak{q}}_0 \end{bmatrix}, \tag{2.156}$$

which can be solved provided $\det\left(\frac{\partial H}{\partial p} \nabla_p g(q,p)^\top + g(q,p) \frac{\partial^2 H}{\partial p^2} + \nabla_p g(q,p) \frac{\partial H}{\partial p}^\top\right) \neq 0$.

The results that we have established are summarized in the following theorem:

Theorem 2.19 ([Duruisseaux et al., 2021])**.** *Suppose H is a nondegenerate Hamiltonian and $g \in C^1([0,h])$ is a monitor function such that*

$$\det\left(\frac{\partial H}{\partial p}\nabla_p g(q,p)^\top + g(q,p)\frac{\partial^2 H}{\partial p^2} + \nabla_p g(q,p)\frac{\partial H}{\partial p}^\top\right) \neq 0. \tag{2.157}$$

If the discrete Hamiltonian $\bar{H}_d^{\pm}$ approximates the exact discrete Hamiltonian $\bar{H}_d^{\pm,E}$ to a given order r, that is,

$$\bar{H}_d^{\pm}(\bar{q}_0,\bar{p}_1;h) = \bar{H}_d^{\pm,E}(\bar{q}_0,\bar{p}_1;h) + \mathcal{O}(h^{r+1}), \tag{2.158}$$

then the discrete Hamiltonian map

$$\tilde{F}_{\bar{H}_d^{\pm}} : (\bar{q}_k,\bar{p}_k) \mapsto (\bar{q}_{k+1},\bar{p}_{k+1}), \tag{2.159}$$

viewed as one-step method, is order r accurate.

Remark 2.7. *It should be noted that the assumptions that the original Hamiltonian is nondegenerate and autonomous fail to hold in the application of time-adaptive variational integrators to the discretization of the Bregman Hamiltonian associated with accelerated optimization which we will consider in Section 3.3, as it is time-dependent. This is unavoidable, as it models a system with dissipation, which cannot be described with an autonomous Hamiltonian, as the Hamiltonian would otherwise be an integral of motion, as it is the Noether quantity associated with time translational symmetry.*

In the cases when the original Hamiltonian is degenerate or nonautonomous, we need to analyze the solvability of the discrete Hamiltonian equations on a case-by-case basis, but as we demonstrate, this can be done in the case of the Bregman Hamiltonian with the given choices of monitor function $g(t)$ and discrete Hamiltonians that we consider.

Variational Derivation of the Poincaré Hamiltonian

All the literature to date on the Poincaré transformation have constructed the Poincaré transformed system by reverse-engineering. We now depart from this traditional strategy and describe a new way to think about the Poincaré transformed Hamiltonian by deriving it from a variational principle. This simple derivation, originally introduced in [Duruisseaux and Leok, 2023a], gives additional insight into the transformation mechanism and will provide natural candidates later on for time-adaptivity on the Lagrangian side and for more general frameworks.

As before, we work in the extended space $(q, \mathfrak{q}, p, \mathfrak{p})$ where $\mathfrak{q} = t$ and $\mathfrak{p}$ is the corresponding conjugate momentum, and consider a time transformation $t \to \tau$ given by $\frac{dt}{d\tau} = g(q, t, p)$. We define an extended action functional $\mathfrak{S} : C^2([0,T], T^*\bar{\mathcal{Q}}) \to \mathbb{R}$ by

$$\begin{aligned}\mathfrak{S}(\bar{q}(\cdot), \bar{p}(\cdot)) &= \bar{p}(T)\bar{q}(T) - \int_0^T \left[\bar{p}(t)\dot{\bar{q}}(t) - H(q(t), t, p(t)) - \mathfrak{p}(t)\right] dt \\ &= \bar{p}(T)\bar{q}(T) - \int_{\tau(t=0)}^{\tau(t=T)} \left[\bar{p}(\tau)\frac{d\tau}{dt}\bar{q}'(\tau) - H(q(\tau), \mathfrak{q}(\tau), p(\tau)) - \mathfrak{p}(\tau)\right] \frac{dt}{d\tau} d\tau \\ &= \bar{p}(T)\bar{q}(T) - \int_{\tau(t=0)}^{\tau(t=T)} \left\{\bar{p}(\tau)\bar{q}'(\tau) - \frac{dt}{d\tau}\left[H(q(\tau), \mathfrak{q}(\tau), p(\tau)) + \mathfrak{p}(\tau)\right]\right\} d\tau,\end{aligned} \tag{2.160}$$

where we have performed a change of variables in the integral. Then,

$$\begin{aligned}\mathfrak{S}(\bar{q}(\cdot), \bar{p}(\cdot)) &= \bar{p}(T)\bar{q}(T) \\ &\quad - \int_{\tau(t=0)}^{\tau(t=T)} \left\{\bar{p}(\tau)\bar{q}'(\tau) - g(q(\tau), \mathfrak{q}(\tau), p(\tau))\left[H(q(\tau), \mathfrak{q}(\tau), p(\tau)) + \mathfrak{p}(\tau)\right]\right\} d\tau.\end{aligned} \tag{2.161}$$

Computing the variation of $\mathfrak{S}$ yields

$$\begin{aligned}\delta\mathfrak{S} &= \bar{q}(T)\delta\bar{p}(T) + \bar{p}(T)\delta\bar{q}(T) \\ &\quad - \int_{\tau(t=0)}^{\tau(t=T)} \left[q'\delta p + p\delta q' - \left(g\frac{\partial H}{\partial q} + \frac{\partial g}{\partial q}(H + \mathfrak{p})\right)\delta q - \left(g\frac{\partial H}{\partial p} + \frac{\partial g}{\partial p}(H + \mathfrak{p})\right)\delta p\right] d\tau \\ &\quad - \int_{\tau(t=0)}^{\tau(t=T)} \left[\mathfrak{q}'\delta\mathfrak{p} + \mathfrak{p}\delta\mathfrak{q}' - \left(g\frac{\partial H}{\partial \mathfrak{q}} + \frac{\partial g}{\partial \mathfrak{q}}(H + \mathfrak{q})\right)\delta\mathfrak{q} - g\delta\mathfrak{p}\right] d\tau,\end{aligned} \tag{2.162}$$

and using integration by parts and the boundary conditions $\delta\bar{q}(0) = \delta\bar{p}(T) = 0$ gives

$$\begin{aligned}\delta\mathfrak{S} &= \int_{\tau(t=0)}^{\tau(t=T)} \left[\mathfrak{p}' + g\frac{\partial H}{\partial \mathfrak{q}} + \frac{\partial g}{\partial \mathfrak{q}}(H + \mathfrak{p})\right]\delta\mathfrak{q} d\tau - \int_{\tau(t=0)}^{\tau(t=T)} \left[\mathfrak{q}' - g\right]\delta\mathfrak{p} d\tau \\ &\quad + \int_{\tau(t=0)}^{\tau(t=T)} \left[p' + g\frac{\partial H}{\partial q} + \frac{\partial g}{\partial q}(H + \mathfrak{p})\right]\delta q d\tau + \int_{\tau(t=0)}^{\tau(t=T)} \left[g\frac{\partial H}{\partial p} + \frac{\partial g}{\partial p}(H + \mathfrak{p}) - q'\right]\delta p d\tau.\end{aligned} \tag{2.163}$$

Thus, the condition that $\mathfrak{S}(\bar{q}(\cdot), \bar{p}(\cdot))$ is stationary with respect to the boundary conditions $\delta\bar{q}(0) = 0$ and $\delta\bar{p}(T) = 0$ is equivalent to $(\bar{q}(\cdot), \bar{p}(\cdot))$ satisfying Hamilton's canonical equations corresponding to the Poincaré transformed Hamiltonian,

$$\mathfrak{q}' = g(q, \mathfrak{q}, p), \tag{2.164}$$

$$q' = g(q, \mathfrak{q}, p)\frac{\partial H}{\partial p}(q, \mathfrak{q}, p) + \frac{\partial g}{\partial p}(q, \mathfrak{q}, p)\left[H(q, \mathfrak{q}, p) + \mathfrak{p}\right], \tag{2.165}$$

$$p' = -g(q, \mathfrak{q}, p)\frac{\partial H}{\partial q}(q, \mathfrak{q}, p) - \frac{\partial g}{\partial q}(q, \mathfrak{q}, p)\left[H(q, \mathfrak{q}, p) + \mathfrak{p}\right], \tag{2.166}$$

$$\mathfrak{p}' = -g(q, \mathfrak{q}, p)\frac{\partial H}{\partial \mathfrak{q}}(q, \mathfrak{q}, p) - \frac{\partial g}{\partial \mathfrak{q}}(q, \mathfrak{q}, p)\left[H(q, \mathfrak{q}, p) + \mathfrak{p}\right]. \tag{2.167}$$

An alternative way to reach the same conclusion is by interpreting equation (2.161) as the usual Type II action functional for the modified Hamiltonian, which coincides with the Poincaré transformed Hamiltonian:

$$g(q(\tau), \mathfrak{q}(\tau), p(\tau))\left[H(q(\tau), \mathfrak{q}(\tau), p(\tau)) + \mathfrak{p}(\tau)\right]. \tag{2.168}$$

Numerical Tests on Kepler's Planar 2-Body Problem

We now demonstrate the approach using Hamiltonian Taylor variational integrators, presented in Section 2.4.2, on Kepler's planar 2-body problem. For a lucid exposition, we assume at first that $g(q, p) = g(q)$ and $H(q, p) = \frac{1}{2}p^{\mathsf{T}}M^{-1}p + V(q)$.

Consider the discrete right Hamiltonian given by approximating $\bar{q}_1$ with a first-order Taylor method about $\bar{q}_0$, approximating $\bar{p}_0$ with a zeroth-order Taylor expansion about $\bar{p}_0$, and using the rectangular quadrature rule about the initial point:

$$\bar{H}_d^+ = p_1^{\mathsf{T}}\left(q_0 + \frac{1}{2}hg(q_0)M^{-1}p_1\right) + \mathfrak{p}_1(\mathfrak{q}_0 + hg(q_0)) + hg(q_0)V(q_0). \tag{2.169}$$

The corresponding variational integrator is given by

$$\bar{p}_1 = \begin{bmatrix} p_0 - hg(q_0)\nabla V(q_0) - h\nabla g(q_0)\left(\frac{1}{2}p_1^{\mathsf{T}}M^{-1}p_1 + V(q_0) + \mathfrak{p}_0\right) \\ \mathfrak{p}_0 \end{bmatrix}, \tag{2.170}$$

$$\bar{q}_1 = \begin{bmatrix} q_0 + hg(q_0)M^{-1}p_1 \\ \mathfrak{q}_0 + hg(q_0) \end{bmatrix}. \tag{2.171}$$

This integrator is merely the symplectic Euler-B method applied to the transformed Hamiltonian system

$$\bar{q}_1 = \bar{q}_0 + h\frac{\partial \bar{H}(\bar{q}_0, \bar{p}_1)}{\partial \bar{p}}, \qquad \bar{p}_1 = \bar{p}_0 - h\frac{\partial \bar{H}(\bar{q}_0, \bar{p}_1)}{\partial \bar{q}}. \tag{2.172}$$

This is precisely the adaptive symplectic integrator first proposed in [Hairer, 1997] and presented in [Leimkuhler and Reich, 2004, page 254]. Most existing symplectic integrators can be interpreted as variational integrators, but there are also new methods that are most naturally derived as variational integrators. We will also consider a fourth-order Hamiltonian Taylor variational integrator (HTVI4), which is distinct from any existing known symplectic method.

One of the most important aspects of implementing a variable time-step symplectic integrator of this form is a well-chosen monitor function, $g(q)$. We need g to be positive-definite, so that we never stall or march backward in time. Noting that the above integrator is first-order, so a natural choice is to use the second-order truncation error given by

$$-\frac{(\mathfrak{q}_1 - \mathfrak{q}_0)^2}{2} M^{-1}\nabla V(q_0). \tag{2.173}$$

Let δ be some desired level of accuracy. Then, using $\mathfrak{q}_1 - \mathfrak{q}_0 = hg(q_0)$, one choice for g is obtained implicitly via the equation

$$g(q_0) \;=\; \frac{\delta}{\left\| \frac{(\mathfrak{q}_1 - \mathfrak{q}_0)^2}{2} g(q_0) M^{-1}\nabla V(q_0) \right\|} \;=\; \frac{2\delta}{\left\| h^2 g(q_0)^3 M^{-1}\nabla V(q_0) \right\|}. \tag{2.174}$$

Explicitly,

$$g(q_0) = \left(\frac{2\delta}{\left\| h^2 M^{-1}\nabla V(q_0) \right\|} \right)^{1/4}. \tag{2.175}$$

Experimentally, the fourth root in equation (2.175) did not affect results very much, but made computations messier, which is why we have chosen the simpler yet very similar monitor function

$$g(q_0) = \frac{2\delta}{\left\| h^2 M^{-1}\nabla V(q_0) \right\|}, \tag{2.176}$$

which achieves an error which is comparable to the chosen value of δ.

Alternative choices for the monitor function $g(q)$, proposed in [Hairer, 1997], include the p-independent arclength parameterization

$$g(q) = \left[2(H_0 - V(q)) + \nabla V(q)^\top M^{-1} \nabla V(q)\right]^{-1/2}, \tag{2.177}$$

and a choice particular to Kepler's 2-Body problem,

$$g(q) = q^\top q, \tag{2.178}$$

which is motivated by Kepler's second law, which states that a line segment joining the two bodies sweeps out equal areas during equal intervals of time.

We have tested the algorithm given by equation (2.170) on Kepler's planar 2-body problem, with an eccentricity of 0.9, using the three choices of monitor function g given by (2.176), (2.177), and (2.178). Of these three choices, (2.178) is specific to Kepler's 2-body problem, while (2.176) and (2.177) are more general choices. Unlike (2.177), which is independent of the order of the method, (2.176) is based on the truncation error and thus the corresponding cost of computing this function will increase as the order of the method increases.

Simulations using Kepler's 2-body problem with an eccentricity of 0.9 over a time interval of $[0, 1000]$ were run using the three different choices of g and the usual symplectic Euler-B. Results indicate that symplectic Euler-B takes the most steps and computational time to achieve a level of accuracy around 10^{-5}. To achieve this level of accuracy, the choice of the truncation error monitor function, (2.176), resulted in the least number of steps, and the second lowest computational time. The lowest computational time belonged to (2.178), but it used significantly more steps than (2.176). The lower computational cost can be attributed to the cheaper evaluation cost of the monitor function and its derivative. Finally, the monitor function (2.177) required the most steps and computational time of the adaptive algorithms, but it is still a good choice in general given its broad applicability. Figures 2.1 and 2.2 present the energy and angular momentum errors for the fixed time-step method versus adaptive time-step method, and the time-steps for the different monitor functions, respectively.

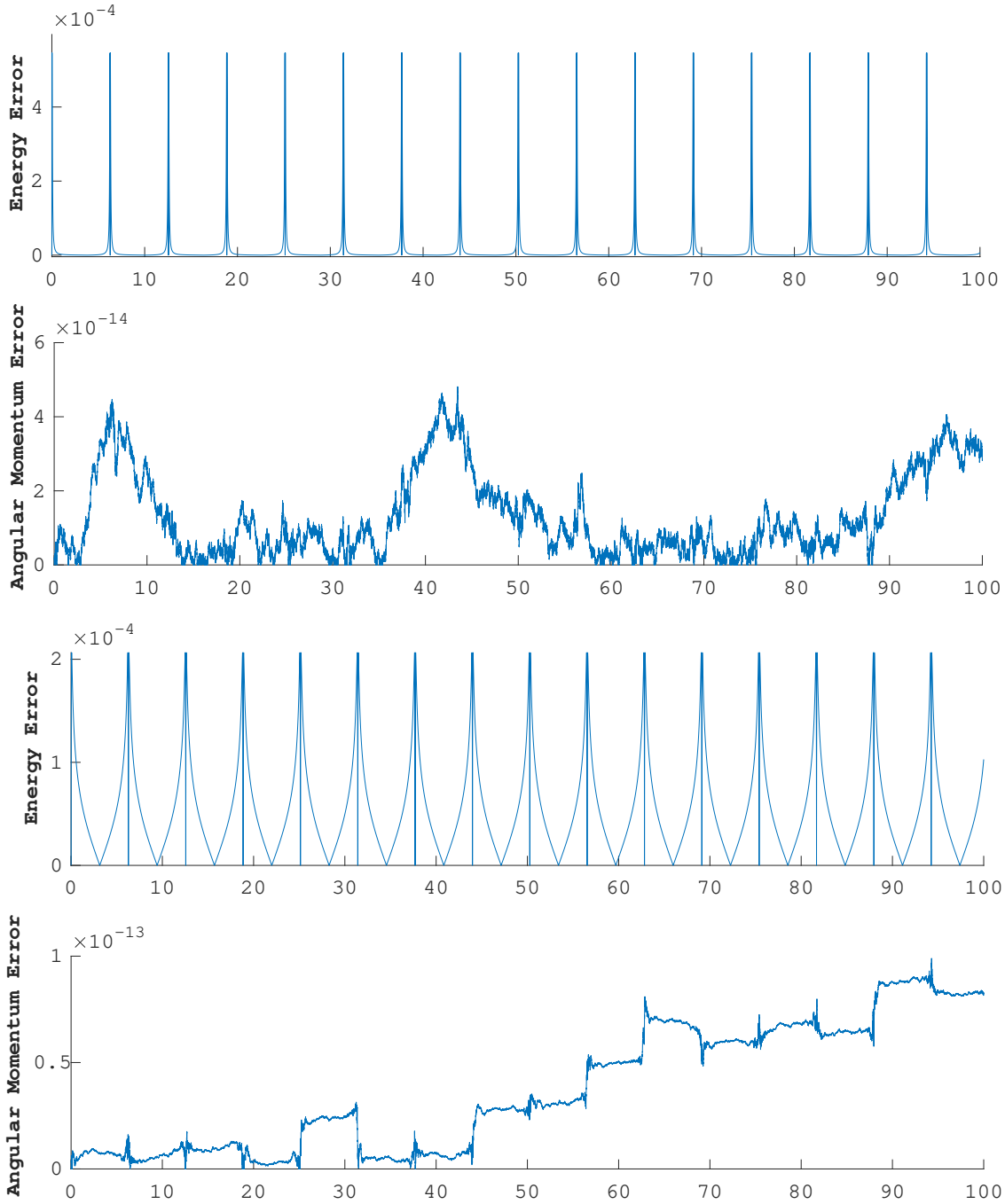


Figure 2.1: Symplectic Euler-B (top two graphs) and the adaptive algorithm with monitor function given by equation (2.176) (bottom two graphs) were applied to Kepler's planar 2-body problem over a time interval of $[0, 100]$ with an eccentricity of 0.9.

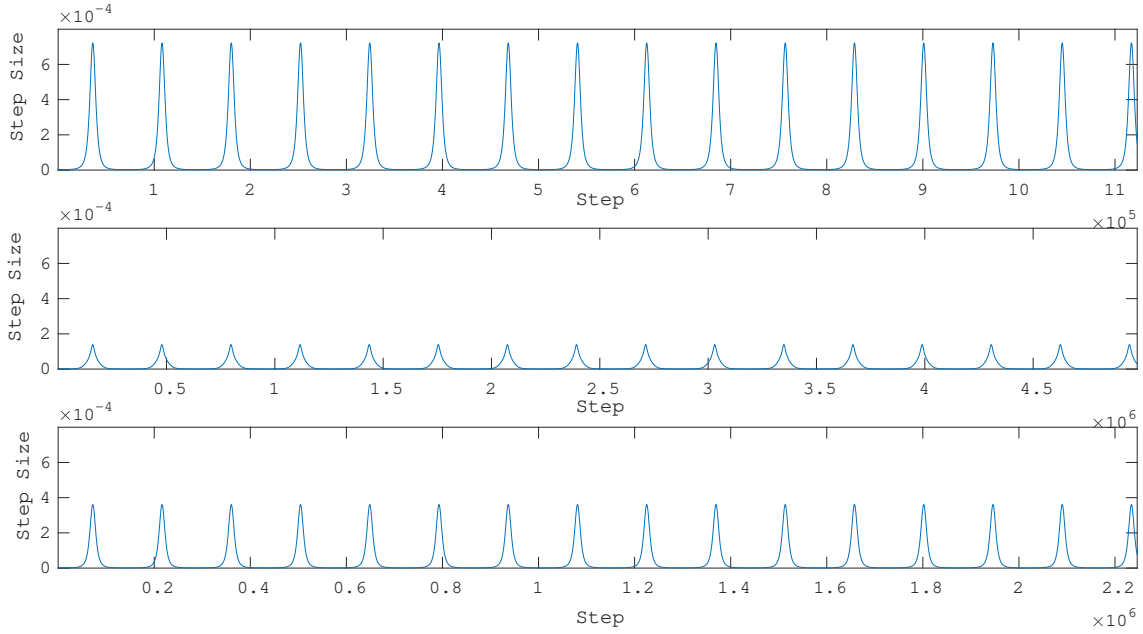


Figure 2.2: Time-steps taken for the various choices of monitor functions. The top, middle and bottom plots correspond to the monitor functions (2.176), (2.177), and (2.178), respectively. All of the monitor functions appear to increase and decrease the time-step at the same points along the trajectory, but clearly (2.176) allowed for larger steps.

Next, we consider a Type II fourth-order Hamiltonian Taylor variational integrator (HTVI4) constructed using the strategy from Section 2.4.2 and the automatic differentiation package from [Neidinger, 2010; Patterson et al., 2013]. We now drop the assumption of p-independent monitor functions and consider the following monitor functions:

$$g(q) = (q^\top q)^\gamma \quad \text{for } \gamma = 0.5 \text{ and } 1, \qquad \text{(Gamma)} \tag{2.179}$$

$$g(q) = \left[2(H_0 - V(q)) + \nabla V(q)^\top M^{-1} \nabla V(q)\right]^{-\frac{1}{2}}, \qquad \text{(Arclength)} \tag{2.180}$$

$$g(q,p) = \|\mathfrak{p} - L(q, M^{-1}p)\|_2^{-1}. \qquad \text{(Energy)} \tag{2.181}$$

The monitor function (2.181) was originally intended to be $\|\mathfrak{p}+H(q,p)\|_2^{-1}$, but experimental results suggested that (2.181) is the better choice. We will discuss the shortcomings of using the inverse energy error in the next paragraph. Note that $\|L(q, M^{-1}p)\|_2^{-1}$ also performs well, but the addition of $\mathfrak{p} = -H(q_0, p_0)$ showed noticeable improvement. It was noted in [Hairer, 1997] that the inverse Lagrangian has been considered as a possible choice for g in the Poincaré transformation, but not in the framework of symplectic integration.

While the choice of (2.179) was generally the most efficient, (2.181) was very close in terms of efficiency and offers a more general monitor function. This also implies that efficiency is not limited to only q-independent or p-independent monitor functions. However, various attempts to construct separable transformed Hamiltonians (see [Blanes and Budd, 2004; Blanes and Iserles, 2012]) required the use of q-independent or p-independent monitor functions, so this is where such monitor functions are most useful.

In the case of monitor functions involving the gradient, higher-order derivatives will be required for higher-order Taylor variational integrators, but there are efficiencies to be had when leveraging the higher-order derivatives already being calculated for the underlying Taylor method and Hessian-vector multiplication that can be done efficiently without needing to explicitly construct the full Hessian [Christianson, 1992]. The calculation of higher-order derivatives come with a higher computational cost, and in the case of Kepler's 2-body problem there is a clear computational advantage in using the gradient-free gamma monitor function (2.179), as shown in Tables 2.1 and 2.2. However, the gamma monitor function (2.179) is more specific to Kepler's 2-body, while the energy and arclength monitor functions are applicable to a wider range of problems. Monitor functions that are both general and efficient would be highly desirable.

Figure 2.3 displays the time-steps taken for the different choices of monitor functions for this fourth-order Hamiltonian Taylor variational integrator.

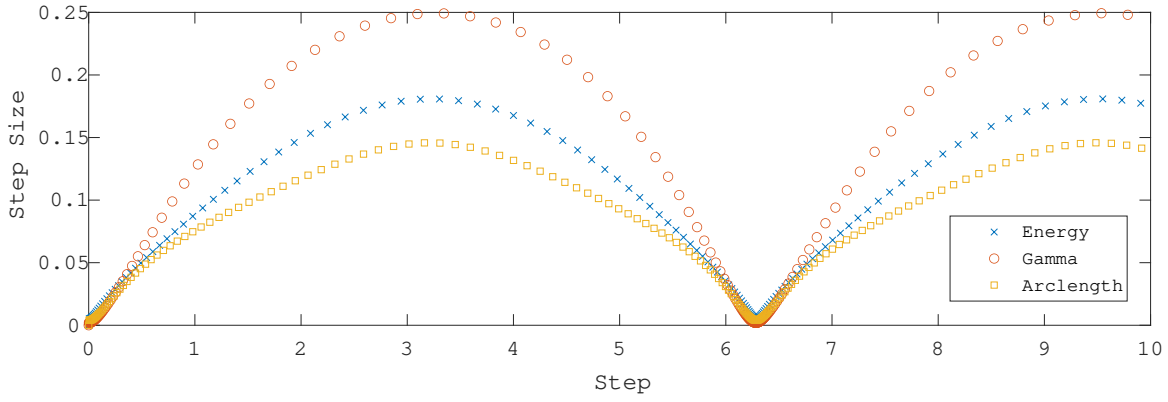


Figure 2.3: Time-steps taken for the different choices of monitor functions g. The energy (2.181) and gamma (2.179) monitor functions performed better, in terms of fewest steps, lowest computational cost, and lowest global error, than the arclength monitor function (2.180). Note that (2.181) did not take the largest nor the smallest steps.

The truncation error monitor function (2.176) performed quite well for first-order methods, and this motivated the choice of using Taylor variational integrators, since derivatives would be readily available. However, its success cannot as easily be applied to higher-order methods. This is due to the fact that for higher-order truncation errors, one obtains an implicit differential-algebraic definition of the monitor function. This deviates from the first-order case, where the monitor function can be solved for explicitly.

Another seemingly natural choice for the monitor function is the inverse of the energy error. However, Taylor variational integrators are constructed using Taylor expansions about the initial point, and consequently the monitor function is mostly evaluated at or near the initial point. If the initial point is at a particularly tricky part of the dynamics and requires a small first step, then the energy error at the first step will not reflect this, since initially the energy error is zero. In contrast, the inverse Lagrangian will be small at an initial point that requires a small first step. The inverse energy error may work well for methods that primarily evaluate the energy error at the end point rather than the initial point. It is also often advantageous to bound the time-step below or above. As noted in [Leimkuhler and Reich, 2004, page 248], this can be done by defining the new monitor function as

$$\hat{g} = b\frac{g+a}{g+b}, \qquad \text{where} \quad a = \frac{\Delta t_{\min}}{\Delta \tau} \quad \text{and} \quad b = \frac{\Delta t_{\max}}{\Delta \tau}. \tag{2.182}$$

Tables 2.1 and 2.2 display a comparison of bounds, computational time, steps, and error. We note that for numerical methods such as the Taylor variational integrator, bounding $g(q,p)$ bounds the time-step, but not directly. Also, compared to non-adaptive variational integrators, such as the non-adaptive Taylor variational integrator and the Störmer–Verlet (SV) method, the adaptive methods showed a significant gain in efficiency for Kepler's 2-body planar problem with high eccentricity, while low eccentricity models do not need nor do they benefit from adaptivity. A Hamiltonian dynamical system with regions of high curvature in the vector field and its norm will in general benefit from an adaptive scheme such as the one outlined here.

Table 2.1: Kepler Planar 2-Body Problem, Eccentricity = 0.9

Method	Monitor $g(q,p)$	h	min Step	max Step	min g	max g	Energy Error	Global Error	Steps	Time
HTVI4	Gamma	0.1	0.0020	0.2493	0.01	8	1.43E-05	7.09E-06	181	26.9
HTVI4	Energy	0.1	0.0051	0.1809	1E-04	2	1.93E-06	4.76E-06	146	28.3
HTVI4	Arclength	0.1	0.0040	0.1458	3E-03	0.3	1.10E-04	3.69E-05	185	70.2
HTVI4	-	0.0025	0.0025	0.0025	-	-	2.50E-06	2.89E-05	4000	120
SV	-	5E-05	5E-05	5E-05	-	-	3.12E-06	4.68E-05	2E05	1.9

Table 2.2: Kepler Planar 2-Body Problem, Eccentricity = 0.99

Method	Monitor $g(q,p)$	h	min Step	max Step	min g	max g	Energy Error	Global Error	Steps	Time
HTVI4	Gamma	0.1	6E-05	0.2648	5E-04	8	4.88E-05	5.60E-06	372	49.3
HTVI4	Energy	0.03	1.5E-04	0.1462	1E-06	5	9.13E-06	4.63E-06	383	58.4
HTVI4	Arclength	0.1	5E-05	0.1379	8E-04	10	1.31E-05	1.49E-05	691	146.0
HTVI4	-	5E-04	5E-04	5E-04	-	-	1.38E-01	7.83E-01	2E04	525.2
SV	-	5E-07	5E-07	5E-07	-	-	3.34E-06	2.68E-05	2E07	189.2

2.7.2 Lagrangian Integrators with Prescribed Variable Time-steps

We now demonstrate how to construct symplectic integrators that allow for the use of a prescribed variable time-steps, on the Lagrangian side. This framework will be particularly useful in the Riemannian manifold setting and on Lie groups, where Lagrangian variational integrators are well-defined while the current formulations of Hamiltonian variational integrators do not make intrinsic sense. Indeed, the current Hamiltonian variational approach involves Type II and III generating functions $H_d^+(q_k, p_{k+1})$, $H_d^-(p_k, q_{k+1})$, which depend on the position at one boundary point, and the momentum at the other boundary point. However, this does not make intrinsic sense on a manifold, since one needs the base point in order to specify the corresponding cotangent space. On the other hand, Lagrangian variational integrators involve a Type I generating function $L_d(q_k, q_{k+1})$ which only depends on the position at the boundary points and is thus well-defined on manifolds, and many Lagrangian variational integrators have been derived on Riemannian manifolds, especially in the Lie group [Leok, 2004; Lee et al.; Hussein et al., 2006; Lee et al., 2007a,b; Lee, 2008; Bou-Rabee and Marsden, 2009; Nordkvist and Sanyal, 2010; Hall and Leok, 2015] and homogeneous space [Lee et al., 2009] settings.

The well-definedness of Lagrangian variational integrators on manifolds gives an incentive to construct a mechanism to incorporate variable time-stepping in Lagrangian variational integrators. On the Hamiltonian side, the Poincaré transformation was at the heart of the construction of time-adaptive geometric integrators via Hamiltonian variational integrators. This provided a systematic method for constructing symplectic integrators of arbitrarily high-order based on the discretization of Hamilton's principle [Marsden and West, 2001; Hall and Leok, 2015], or equivalently, by the approximation of the generating function of the symplectic flow map. However, the Poincaré transformed Hamiltonian is degenerate and therefore does not have a corresponding Type I Lagrangian formulation. As a result, we cannot exploit the usual correspondence between Hamiltonian and Lagrangian dynamics and need to come up with a different strategy to allow time-adaptivity from the Lagrangian perspective.

Time-adaptivity from a Variational Principle on the Lagrangian side

We will now derive a mechanism for time-adaptivity on the Lagrangian side by mimicking the variational derivation of the Poincaré Hamiltonian. We will work in the extended space $\bar{q} = (q, \mathfrak{q}, \lambda)^\intercal \in \bar{\mathcal{Q}}$ where $\mathfrak{q} = t$ and λ is a Lagrange multiplier used to enforce the time rescaling

$$\frac{dt}{d\tau} = g(t). \tag{2.183}$$

Consider the action functional $\mathfrak{S} : C^2([0,T], T\bar{\mathcal{Q}}) \to \mathbb{R}$ given by

$$\begin{aligned}
\mathfrak{S}(\bar{q}(\cdot), \dot{\bar{q}}(\cdot)) &= \int_0^T \left[L(q(t), \dot{q}(t), \mathfrak{q}(t)) - \lambda(t) \left(\frac{d\mathfrak{q}}{d\tau} - g(\mathfrak{q}(t)) \right) \right] dt \\
&= \int_{\tau(t=0)}^{\tau(t=T)} \left[\frac{dt}{d\tau} L\left(q(\tau), \frac{d\tau}{dt} q'(\tau), \mathfrak{q}(\tau) \right) - \lambda(\tau) \frac{dt}{d\tau} \left(\frac{d\mathfrak{q}}{d\tau} - g(\mathfrak{q}(\tau)) \right) \right] d\tau \\
&= \int_{\tau(t=0)}^{\tau(t=T)} \left[\mathfrak{q}'(\tau) L\left(q(\tau), \frac{d\tau}{dt} q'(\tau), \mathfrak{q}(\tau) \right) - \lambda(\tau) \mathfrak{q}'(\tau) \left[\mathfrak{q}'(\tau) - g(\mathfrak{q}(\tau)) \right] \right] d\tau,
\end{aligned} \tag{2.184}$$

where, as before, we have performed a change of variables in the integral. This is the usual Type I action functional for the extended autonomous Lagrangian,

$$\bar{L}(\bar{q}(\tau), \bar{q}'(\tau)) = \mathfrak{q}'(\tau) L\left(q(\tau), \frac{d\tau}{dt} q'(\tau), \mathfrak{q}(\tau) \right) - \lambda(\tau) \mathfrak{q}'(\tau) \left[\mathfrak{q}'(\tau) - g(\mathfrak{q}(\tau)) \right]. \tag{2.185}$$

Theorem 2.20 ([Duruisseaux and Leok, 2023a]). *If the curve* $(\bar{q}(\tau), \bar{q}'(\tau))$ *satisfies the Euler–Lagrange equations corresponding to the Lagrangian* $\bar{L}$*, then its components satisfy* $\frac{dt}{d\tau} = g(t)$ *and the original Euler–Lagrange equations*

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}}(q, \dot{q}, t) = \frac{\partial L}{\partial q}(q, \dot{q}, t). \tag{2.186}$$

Proof. Substituting the expression for $\bar{L}$ into the Euler–Lagrange equations, $\frac{d}{d\tau}\frac{\partial \bar{L}}{\partial \lambda'} = \frac{\partial \bar{L}}{\partial \lambda}$, and $\frac{d}{d\tau}\frac{\partial \bar{L}}{\partial q'} = \frac{\partial \bar{L}}{\partial q}$, gives

$$\mathfrak{q}'\left[\mathfrak{q}' - g(\mathfrak{q})\right] = 0,$$

and

$$\frac{d\mathfrak{q}}{d\tau}\frac{d}{d\mathfrak{q}}\left[\mathfrak{q}'\frac{\partial L\left(q, \frac{d\tau}{d\mathfrak{q}}q', \mathfrak{q}\right)}{\partial q'}\right] = \mathfrak{q}'\frac{\partial L\left(q, \frac{d\tau}{d\mathfrak{q}}q', \mathfrak{q}\right)}{\partial q}.$$

Now, $\mathfrak{q}' = g(\mathfrak{q}) > 0$ so $\mathfrak{q}' = g(\mathfrak{q})$, and the chain rule gives

$$\frac{d}{d\mathfrak{q}}\frac{\partial L}{\partial \dot{q}}\left(q, \frac{d\tau}{d\mathfrak{q}}q', \mathfrak{q}\right) = \frac{\partial L}{\partial q}\left(q, \frac{d\tau}{d\mathfrak{q}}q', \mathfrak{q}\right).$$

Using the equation $\dot{q} = \frac{d\tau}{d\mathfrak{q}}q'$ and replacing $\mathfrak{q}$ by t recovers the original Euler–Lagrange equations. □

We now introduce a discrete variational formulation of these continuous Lagrangian mechanics. Suppose we are given a partition $0 = \tau_0 < \tau_1 < \ldots < \tau_N = \mathcal{T}$ of the interval $[0, \mathcal{T}]$, and a discrete curve in $\mathcal{Q} \times \mathbb{R} \times \mathbb{R}$ denoted by $\{(q_k, \mathfrak{q}_k, \lambda_k)\}_{k=0}^N$ such that $q_k \approx q(\tau_k)$, $\mathfrak{q}_k \approx \mathfrak{q}(\tau_k)$, and $\lambda_k \approx \lambda(\tau_k)$. Consider the discrete action functional,

$$\bar{\mathfrak{S}}_d\left(\{(q_k, \mathfrak{q}_k, \lambda_k)\}_{k=0}^N\right) = \sum_{k=0}^{N-1}\left[L_d(q_k, \mathfrak{q}_k, q_{k+1}, \mathfrak{q}_{k+1}) - \lambda_k\frac{\mathfrak{q}_{k+1} - \mathfrak{q}_k}{\tau_{k+1} - \tau_k} + \lambda_k g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1} - \mathfrak{q}_k}{\tau_{k+1} - \tau_k}, \tag{2.187}$$

where

$$L_d(q_k, \mathfrak{q}_k, q_{k+1}, \mathfrak{q}_{k+1}) \approx \underset{\substack{(q,\mathfrak{q})\in C^2([\tau_k,\tau_{k+1}],\mathcal{Q}\times\mathbb{R}) \\ (q,\mathfrak{q})(\tau_k)=(q_k,\mathfrak{q}_k),\ (q,\mathfrak{q})(\tau_{k+1})=(q_{k+1},\mathfrak{q}_{k+1})}}{\operatorname{ext}} \int_{\tau_k}^{\tau_{k+1}} L\left(q, \frac{q'}{g(\mathfrak{q})}, \mathfrak{q}\right)d\tau. \tag{2.188}$$

$\bar{\mathfrak{S}}_d$ is a discrete analogue of the action functional $\bar{\mathfrak{S}} : C^2([0, T], \mathcal{Q} \times \mathbb{R} \times \mathbb{R}) \to \mathbb{R}$ given by

$$\bar{\mathfrak{S}}(q(\cdot), \mathfrak{q}(\cdot), \lambda(\cdot)) = \int_0^{\mathcal{T}} \bar{L}\left(q(\tau), \mathfrak{q}(\tau), \lambda(\tau), q'(\tau), \mathfrak{q}'(\tau), \lambda'(\tau)\right)d\tau \tag{2.189}$$

$$= \int_0^{\mathcal{T}}\left[L\left(q, \frac{q'}{g(\mathfrak{q})}, \mathfrak{q}\right) - \lambda\mathfrak{q}' + \lambda g(\mathfrak{q})\right]\mathfrak{q}'d\tau. \tag{2.190}$$

We can derive the following result which relates a discrete Type I variational principle to a set of discrete Euler–Lagrange equations:

Theorem 2.21 ([Duruisseaux and Leok, 2023a])**.** *The Type I discrete Hamilton's variational principle,*

$$\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=0, \tag{2.191}$$

is equivalent to the discrete extended Euler–Lagrange equations,

$$\mathfrak{q}_{k+1}=\mathfrak{q}_k+(\tau_{k+1}-\tau_k)g(\mathfrak{q}_k), \tag{2.192}$$

$$\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_1L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})+\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}D_3L_d(q_{k-1},\mathfrak{q}_{k-1},q_k,\mathfrak{q}_k)=0, \tag{2.193}$$

$$\begin{aligned}&\left[D_2L_{d_k}+\lambda_k\frac{1}{\tau_{k+1}-\tau_k}+\lambda_k\nabla g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}-\frac{1}{\tau_{k+1}-\tau_k}\left[L_{d_k}-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\\&+\left[D_4L_{d_{k-1}}-\lambda_{k-1}\frac{1}{\tau_k-\tau_{k-1}}\right]\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\frac{1}{\tau_k-\tau_{k-1}}\left[L_{d_{k-1}}-\lambda_{k-1}\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\lambda_{k-1}g(\mathfrak{q}_{k-1})\right]=0,\end{aligned} \tag{2.194}$$

where L_{d_k} denotes $L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})$.

Proof. See Appendix A.3.1. □

Defining the discrete momenta via the discrete Legendre transformations,

$$p_k=-D_1L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1}), \tag{2.195}$$

$$\mathfrak{p}_k=-D_2L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1}), \tag{2.196}$$

and using a constant time-step h in τ, the discrete Euler–Lagrange equations can be rewritten as

$$p_k=-D_1L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1}), \tag{2.197}$$

$$\mathfrak{p}_k=-D_2L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1}), \tag{2.198}$$

$$\mathfrak{q}_{k+1}=\mathfrak{q}_k+hg(\mathfrak{q}_k), \tag{2.199}$$

$$p_{k+1}=\frac{g(\mathfrak{q}_k)}{g(\mathfrak{q}_{k+1})}D_3L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1}), \tag{2.200}$$

$$\mathfrak{p}_{k+1}=\frac{L_{d_k}-L_{d_{k+1}}}{hg(\mathfrak{q}_{k+1})}+\frac{\lambda_{k+1}}{h}+\lambda_{k+1}\nabla g(\mathfrak{q}_{k+1})+\frac{g(\mathfrak{q}_k)}{g(\mathfrak{q}_{k+1})}\left[D_4L_{d_k}-\frac{\lambda_k}{h}\right]. \tag{2.201}$$

A Second Time-Adaptive Framework obtained by Reverse-Engineering

As mentioned earlier, all the literature to date on the Poincaré transformation have constructed the Poincaré transformed system by reverse-engineering. The Poincaré transformed Hamiltonian is chosen in such a way that the corresponding component dynamics satisfy the Hamilton's equations in the original space. We will follow a similar strategy to derive a second framework for time-adaptivity from the Lagrangian perspective.

Given a time-dependent Lagrangian $L(q(t), \dot{q}(t), t)$ consider a transformation of time $t \to \tau$,

$$\frac{dt}{d\tau} = g(t), \tag{2.202}$$

described by the monitor function $g(t)$. As before, the time t shall be referred to as the physical time, while τ will be referred to as the fictive time, and we will denote derivatives with respect to t and τ by dots and apostrophes, respectively.

We define the autonomous Lagrangian,

$$\bar{L}(\bar{q}(\tau), \bar{q}'(\tau)) \;=\; \mathfrak{q}' L\left(q, \frac{q'}{g(\mathfrak{q})}, \mathfrak{q}\right) - \lambda\left(\mathfrak{q}' - g(\mathfrak{q})\right), \tag{2.203}$$

in the extended space with $\bar{q} = (q, \mathfrak{q}, \lambda)^\top$ where $\mathfrak{q} = t$, and where λ is a multiplier used to impose the constraint that the time evolution is guided by the monitor function $g(t)$. Note that in contrast to the earlier framework, the Lagrange multiplier term lacks an extra multiplicative factor of $\mathfrak{q}'$.

Theorem 2.22 ([Duruisseaux and Leok, 2023a])**.** *If the curve $(\bar{q}(\tau), \bar{q}'(\tau))$ satisfies the Euler–Lagrange equations corresponding to the Lagrangian $\bar{L}$, then its components satisfy*

$$\frac{dt}{d\tau} = g(t) \tag{2.204}$$

and the original Euler–Lagrange equations

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}}(q, \dot{q}, t) = \frac{\partial L}{\partial q}(q, \dot{q}, t). \tag{2.205}$$

Proof. Substituting the expression for $\bar{L}$ in the Euler–Lagrange equations $\frac{d}{d\tau}\frac{\partial \bar{L}}{\partial \lambda'} = \frac{\partial \bar{L}}{\partial \lambda}$ and $\frac{d}{d\tau}\frac{\partial \bar{L}}{\partial q'} = \frac{\partial \bar{L}}{\partial q}$ gives

$$\mathfrak{q}' = g(\mathfrak{q}),$$

and

$$\frac{d\mathfrak{q}}{d\tau}\frac{d}{d\mathfrak{q}}\left[\frac{d\mathfrak{q}}{d\tau}\frac{\partial L\left(q, \frac{q'}{g(\mathfrak{q})}, \mathfrak{q}\right)}{\partial q'}\right] = \frac{d\mathfrak{q}}{d\tau}\frac{\partial L\left(q, \frac{q'}{g(\mathfrak{q})}, \mathfrak{q}\right)}{\partial q}.$$

We can divide by $\frac{d\mathfrak{q}}{d\tau}$ and use the chain rule to get $\mathfrak{q}' = g(\mathfrak{q})$ and

$$\frac{d}{d\mathfrak{q}}\frac{\partial L}{\partial \dot{q}}\left(q, \frac{d\tau}{d\mathfrak{q}}q', \mathfrak{q}\right) = \frac{\partial L}{\partial q}\left(q, \frac{d\tau}{d\mathfrak{q}}q', \mathfrak{q}\right).$$

Using the equations $\dot{q} = \frac{d\tau}{d\mathfrak{q}}q'$ and $\mathfrak{q}' = g(\mathfrak{q})$, and replacing $\mathfrak{q}$ by t recovers the desired equations. □

We now introduce a discrete variational formulation of these continuous Lagrangian mechanics. Suppose we are given a partition $0 = \tau_0 < \tau_1 < \ldots < \tau_N = \mathcal{T}$ of the interval $[0, \mathcal{T}]$, and a discrete curve in $\mathcal{Q} \times \mathbb{R} \times \mathbb{R}$ denoted by $\{(q_k, \mathfrak{q}_k, \lambda_k)\}_{k=0}^N$ such that $q_k \approx q(\tau_k)$, $\mathfrak{q}_k \approx \mathfrak{q}(\tau_k)$, and $\lambda_k \approx \lambda(\tau_k)$.

Consider the discrete action functional,

$$\bar{\mathfrak{S}}_d\left(\{(q_k, \mathfrak{q}_k, \lambda_k)\}_{k=0}^N\right) = \sum_{k=0}^{N-1}\left\{\frac{\mathfrak{q}_{k+1} - \mathfrak{q}_k}{\tau_{k+1} - \tau_k}\left[L_d(q_k, \mathfrak{q}_k, q_{k+1}, \mathfrak{q}_{k+1}) - \lambda_k\right] + \lambda_k g(\mathfrak{q}_k)\right\}, \tag{2.206}$$

where,

$$L_d(q_k, \mathfrak{q}_k, q_{k+1}, \mathfrak{q}_{k+1}) \approx \underset{\substack{(q,\mathfrak{q})\in C^2([\tau_k,\tau_{k+1}],\mathcal{Q}\times\mathbb{R}) \\ (q,\mathfrak{q})(\tau_k)=(q_k,\mathfrak{q}_k),\ (q,\mathfrak{q})(\tau_{k+1})=(q_{k+1},\mathfrak{q}_{k+1})}}{\operatorname{ext}} \int_{\tau_k}^{\tau_{k+1}} L\left(q, \frac{q'}{g(\mathfrak{q})}, \mathfrak{q}\right) d\tau. \tag{2.207}$$

This discrete action functional $\bar{\mathfrak{S}}_d$ can be thought of as a discrete analogue of the action functional $\bar{\mathfrak{S}} : C^2([0,T], \mathcal{Q} \times \mathbb{R} \times \mathbb{R}) \to \mathbb{R}$ given by

$$\bar{\mathfrak{S}}(q(\cdot), \mathfrak{q}(\cdot), \lambda(\cdot)) = \int_0^{\mathcal{T}} \bar{L}\left(q(\tau), \mathfrak{q}(\tau), \lambda(\tau), q'(\tau), \mathfrak{q}'(\tau), \lambda'(\tau)\right) d\tau \tag{2.208}$$

$$= \int_0^{\mathcal{T}} \left\{\mathfrak{q}'\left[L\left(q, \frac{q'}{g(\mathfrak{q})}, \mathfrak{q}\right) - \lambda\right] + \lambda g(\mathfrak{q})\right\} d\tau. \tag{2.209}$$

We can derive the following result which relates a discrete Type I variational principle to a set of discrete Euler–Lagrange equations:

Theorem 2.23 ([Duruisseaux and Leok, 2023a])**.** *The Type I discrete Hamilton's variational principle,*

$$\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=0, \tag{2.210}$$

is equivalent to the discrete extended Euler–Lagrange equations,

$$\mathfrak{q}_{k+1}=\mathfrak{q}_k+(\tau_{k+1}-\tau_k)g(\mathfrak{q}_k), \tag{2.211}$$

$$\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_1L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})+\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}D_3L_d(q_{k-1},\mathfrak{q}_{k-1},q_k,\mathfrak{q}_k)=0, \tag{2.212}$$

$$\begin{aligned}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_2L_{d_k}-\frac{1}{\tau_{k+1}-\tau_k}L_{d_k}+\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}D_4L_{d_{k-1}}+\frac{1}{\tau_k-\tau_{k-1}}L_{d_{k-1}}\\ =\frac{\lambda_{k-1}}{\tau_k-\tau_{k-1}}-\frac{\lambda_k}{\tau_{k+1}-\tau_k}-\lambda_k\nabla g(\mathfrak{q}_k),\end{aligned} \tag{2.213}$$

where L_{d_k} denotes $L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})$.

Proof. See Appendix A.3.2. □

Defining the discrete momenta via the discrete Legendre transformations,

$$p_k=-D_1L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1}), \tag{2.214}$$

$$\mathfrak{p}_k=-D_2L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1}), \tag{2.215}$$

and using a constant time-step h in τ, the discrete Euler–Lagrange equations can be rewritten as

$$p_k=-D_1L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1}), \tag{2.216}$$

$$\mathfrak{p}_k=-D_2L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1}), \tag{2.217}$$

$$\mathfrak{q}_{k+1}=\mathfrak{q}_k+hg(\mathfrak{q}_k), \tag{2.218}$$

$$p_{k+1}=\frac{g(\mathfrak{q}_k)}{g(\mathfrak{q}_{k+1})}D_3L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1}), \tag{2.219}$$

$$\mathfrak{p}_{k+1}=\frac{L_{d_k}-L_{d_{k+1}}+\lambda_{k+1}-\lambda_k+h\lambda_{k+1}\nabla g(\mathfrak{q}_{k+1})+hg(\mathfrak{q}_k)D_4L_{d_k}}{hg(\mathfrak{q}_{k+1})}. \tag{2.220}$$

2.7.3 Remarks on the Frameworks for Time-Adaptivity

Time-adaptivity in geometric integrators comes more naturally on the Hamiltonian side through the Poincaré transformation. Indeed, in the Hamiltonian case, the time-rescaling equation $\mathfrak{q}' = g(q, \mathfrak{q}, p)$ emerged naturally through the change of time variable inside the extended action functional. By contrast, in the Lagrangian case, we need to impose the time-rescaling equation as a constraint via a multiplier, which we then consider as an extra position coordinate. This strategy can be thought of as being a special case of the more general framework for constrained Lagrangian variational integrators (see Section 2.6.1 or [Marsden and West, 2001; Duruisseaux and Leok, 2022a]).

The Poincaré transformation on the Hamiltonian side was presented in [Duruisseaux et al., 2021] and Section 2.7.1 for the general case where the monitor function can depend on position, time and momentum, $g = g(q, t, p)$. For the accelerated optimization application which is our main motivation to develop a time-adaptive framework for geometric integrators, the monitor function only depends on time, $g = g(t)$. For the sake of simplicity and clarity, we have decided to only present the theory for time-adaptive Lagrangian integrators for monitor functions of the form $g = g(t)$ in Section 2.7.2. Note however that this time-adaptivity framework on the Lagrangian side can be extended to the case where the monitor function also depends on position, $g = g(q, t)$. The action integral remains the same with the exception that g is now a function of $(q, \mathfrak{q})$. Unlike the case where $g = g(t)$, the corresponding Euler–Lagrange equation

$$\frac{d}{d\tau}\frac{\partial \bar{L}}{\partial q'} = \frac{\partial L}{\partial q} \tag{2.221}$$

yields an extra term $\lambda(t)\frac{\partial g}{\partial q}(q, t)$ in the original phase-space,

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}}(q, \dot{q}, t) - \frac{\partial L}{\partial q}(q, \dot{q}, t) = \lambda(t)\frac{\partial g}{\partial q}(q, t). \tag{2.222}$$

The discrete Euler–Lagrange equations become more complicated and involve terms with partial derivatives $D_1 g(q_k, \mathfrak{q}_k)$ of g with respect to q. Furthermore, when $g = g(q, t)$, the discrete Euler–Lagrange equations involve λ_k but the time-evolution of the Lagrange multiplier λ is not well-defined, so the discrete Hamiltonian map corresponding to the

discrete Lagrangian L_d is not well-defined, as explained in [Marsden and West, 2001, page 440]. Although there are ways to circumvent this problem, this adds some difficulty and makes the time-adaptive Lagrangian approach with $g = g(q,t)$ less natural and desirable than the corresponding Poincaré transformation on the Hamiltonian side.

It might also be tempting to generalize further and consider the case where the monitor function also depends on $\dot{q}$, that is, $g = g(q,\dot{q},t)$. However, in this case, the time-rescaling equation $\frac{dt}{d\tau} = g(q,\dot{q},t)$ becomes implicit and it becomes less clear how to generalize the variational derivation presented in this section. There are examples where time-adaptivity with these more general monitor functions proved advantageous (see for instance Kepler's problem in Section 2.7.1 and [Duruisseaux et al., 2021]). This motivates further effort towards developing a better framework for time-adaptivity on the Lagrangian side with more general monitor functions.

It might be more natural to consider the time-rescaled Lagrangian and Hamiltonian dynamics as Dirac mechanics [Yoshimura and Marsden, 2006a,b; Leok and Ohsawa, 2011] on the Pontryagin bundle $(q,v,p) \in T\mathcal{Q} \oplus T^*\mathcal{Q}$. Dirac dynamics are described by the Hamilton-Pontryagin variational principle where the momentum p acts as a Lagrange multiplier to impose the kinematic equation $\dot{q} = v$,

$$\delta \int_0^T \left[L(q,v,t) + p(\dot{q} - v) \right] dt = 0. \tag{2.223}$$

This provides a variational description of both Lagrangian and Hamiltonian mechanics, yields the implicit Euler–Lagrange equations,

$$\dot{q} = v, \qquad \dot{p} = \frac{\partial L}{\partial q}, \qquad p = \frac{\partial L}{\partial v}, \tag{2.224}$$

and suggest the introduction of a more general quantity, the generalized energy,

$$E(q,v,p,t) = pv - L(q,v,t), \tag{2.225}$$

as an alternative to the Hamiltonian.

➢ Chapter 2 contains original material from

① "Adaptive Hamiltonian Variational Integrators and Applications to Symplectic Accelerated Optimization" by V. Duruisseaux, J. Schmitt, and M. Leok. *SIAM Journal on Scientific Computing,* Vol.43, No.4, A2949-A2980, 2021

② "Accelerated Optimization on Riemannian Manifolds via Discrete Constrained Variational Integrators" by V. Duruisseaux and M. Leok. *Journal of Nonlinear Science*, Vol.32, No.42, 2022

③ "Time-adaptive Lagrangian Variational Integrators for Accelerated Optimization on Manifolds" by V. Duruisseaux and M. Leok. *Journal of Geometric Mechanics*, Vol.15, Issue 1, pages 224-255, 2023.

The dissertation author was the primary investigator and author of this paper.

Part II

Accelerated Optimization via Geometric Numerical Integration

3 Accelerated Optimization via Geometric Numerical Integrators on Normed Vector Spaces

3.1 Motivation

Efficient optimization has become one of the major concerns in data analysis. Many machine learning algorithms are designed around the minimization of a loss function or the maximization of a likelihood function. Due to the ever-growing scale of the data sets and size of the problems, there has been a lot of focus on first-order optimization algorithms because of their low cost per iteration and the ease with which they can be executed on parallel and distributed processing architectures.

The first gradient descent algorithm was proposed in [Cauchy, 1847] by Cauchy to deal with the very large systems of equations he was facing when trying to simulate orbits of celestial bodies, and many gradient-based optimization methods have been proposed since Cauchy's work in 1847. In 1983, **Nesterov's Accelerated Gradient** method was introduced in [Nesterov, 1983],

$$x_k = y_{k-1} - h\nabla f(y_{k-1}), \qquad y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1}), \tag{3.1}$$

which converges in $\mathcal{O}(1/k^2)$ to the minimum of the convex objective function f, improving on the $\mathcal{O}(1/k)$ convergence rate exhibited by the standard gradient descent methods.

This $\mathcal{O}(1/k^2)$ rate of convergence was shown in [Nesterov, 2004] to be optimal among first-order methods using only information about ∇f at consecutive iterates. This phenomenon in which an algorithm displays this improved rate of convergence is referred to as **acceleration**. Note that several other accelerated algorithms have been derived since Nesterov's algorithm, such as accelerated mirror descent [Nemirovsky and Yudin, 1983], and accelerated cubic-regularized Newton's method [Nesterov, 2008].

More recently, it was shown in [Su et al., 2016] that Nesterov's Accelerated Gradient method limits to the second-order ODE,

$$\ddot{x}(t) + \frac{3}{t}\dot{x}(t) + \nabla f(x(t)) = 0, \tag{3.2}$$

as the step size h goes to 0. The authors also proved that the objective function $f(x(t))$ converges to its optimal value at a rate of $\mathcal{O}(1/t^2)$ along the trajectories of this ODE. It was then shown in [Wibisono et al., 2016] that in continuous time, the convergence rate of $f(x(t))$ can be accelerated to an arbitrary convergence rate, by considering flow maps generated by a family of time-dependent Bregman Lagrangian and Hamiltonian systems on normed vector spaces which is closed under time rescaling.

We will present this result in greater details in Section 3.2 together with the variational framework introduced in [Wibisono et al., 2016] for accelerated optimization on normed vector spaces. We will then introduce a time-adaptive approach to integrating the resulting time-dependent Hamiltonian and Lagrangian dynamical systems and exploit the frameworks for incorporating variable time-stepping in symplectic integrators introduced in Section 2.7 to design efficient symplectic algorithms for accelerated optimization in Sections 3.3 and 3.4. We will also conduct a careful computational study to investigate how time-adaptivity and symplecticity of the numerical schemes affect the performance of the resulting optimization algorithms, and compare their performance with those of other popular optimization algorithms.

3.2 A Variational Formulation of Accelerated Optimization on Normed Vector Spaces

In this section, we review the variational framework introduced in [Wibisono et al., 2016] for accelerated optimization on normed vector spaces. In a general space $\mathcal{X}$, given a convex, continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$ such that $\|\nabla h(x)\| \to \infty$ as $\|x\| \to \infty$, its corresponding **Bregman divergence** is given by

$$D_h(x, y) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle. \tag{3.3}$$

We then define the **Bregman Lagrangian and Hamiltonian**

$$\boxed{L_{\alpha,\beta,\gamma}(x, v, t) = e^{\alpha_t + \gamma_t} \left[D_h(x + e^{-\alpha_t} v, x) - e^{\beta_t} f(x) \right],} \tag{3.4}$$

$$\boxed{H_{\alpha,\beta,\gamma}(x, r, t) = e^{\alpha_t + \gamma_t} \left[D_{h^*}(\nabla h(x) + e^{-\gamma_t} r, \nabla h(x)) + e^{\beta_t} f(x) \right],} \tag{3.5}$$

which are scalar-valued functions of position $x \in \mathcal{X}$, velocity $v \in \mathbb{R}^d$, momentum $r \in \mathbb{R}^d$, and time t, which are parametrized by smooth functions of time, α, β, γ. Here, the function $h^* : \mathcal{X}^* \to \mathbb{R}$ denotes the Legendre transform (or convex dual function) of h, defined by $h^*(w) = \sup_{z \in \mathcal{X}} [\langle w, z \rangle - h(z)]$. The Euler–Lagrange equation associated to the Bregman Lagrangian (3.4) is given by

$$\begin{aligned} \left[\nabla^2 h \left(x_t + e^{-\alpha_t} \dot{x}_t \right) \right]^{-1} \Big(e^{\beta_t} \nabla f(x_t) + (\dot{\gamma}_t e^{-\alpha_t} - 1) \left[\nabla h \left(x_t + e^{-\alpha_t} \dot{x}_t \right) - \nabla h(x_t) \right] \Big) \\ + e^{-2\alpha_t} \ddot{x}_t + \left(e^{-\alpha_t} - \dot{\alpha}_t e^{-2\alpha_t} \right) \dot{x}_t = 0. \end{aligned} \tag{3.6}$$

The parameter functions α, β, γ are said to satisfy the **ideal scaling conditions** if

$$\dot{\beta}_t \le e^{\alpha_t} \qquad \text{and} \qquad \dot{\gamma}_t = e^{\alpha_t}. \tag{3.7}$$

If the ideal scaling conditions are satisfied, then the Euler–Lagrange equation becomes

$$\boxed{\ddot{x}_t + (e^{\alpha_t} - \dot{\alpha}_t) \dot{x}_t + e^{2\alpha_t + \beta_t} \left[\nabla^2 h \left(x_t + e^{-\alpha_t} \dot{x}_t \right) \right]^{-1} \nabla f(x_t) = 0,} \tag{3.8}$$

and Theorem 1.1 in [Wibisono et al., 2016] states that

$$\boxed{f(x(t)) - f(x^*) \;\le\; \mathcal{O}(e^{-\beta_t}),} \tag{3.9}$$

along the solution $x(t)$ of this Bregman Euler–Lagrange equation, where x^* is the desired minimizer of the objective function f.

Another very important property of this family of Bregman Lagrangians is its closure under time rescaling:

Theorem 3.1 ([Wibisono et al., 2016])**.** *Suppose the curve $x(t)$ satisfies the Euler–Lagrange equations corresponding to the Bregman Lagrangian $L_{\alpha,\beta,\gamma}$. Then the reparametrized curve $y(t) = x(\tau(t))$ satisfies the Euler–Lagrange equations corresponding to the Bregman Lagrangian $L_{\tilde{\alpha},\tilde{\beta},\tilde{\gamma}}$ where*

$$\tilde{\alpha}_t = \alpha_{\tau(t)} + \log \dot{\tau}(t), \qquad \tilde{\beta}_t = \beta_{\tau(t)}, \qquad \tilde{\gamma}_t = \gamma_{\tau(t)}, \tag{3.10}$$

and

$$L_{\tilde{\alpha},\tilde{\beta},\tilde{\gamma}}(x, v, t) \;=\; \dot{\tau}(t) L_{\alpha,\beta,\gamma}\left(x, \frac{1}{\dot{\tau}(t)} v, \tau(t)\right). \tag{3.11}$$

Furthermore α, β, γ satisfy the ideal scaling equation (3.7) *if and only if $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$ do.*

A subfamily of Bregman Lagrangians of interest, indexed by a parameter $p > 0$, is given by the choice of parameter functions

$$\boxed{\alpha_t = \log p - \log t, \qquad \beta_t = p \log t + \log C, \qquad \gamma_t = p \log t,} \tag{3.12}$$

where $C > 0$ is a constant. The Bregman Lagrangian and Hamiltonian become

$$\boxed{L_p(x, v, t) = p t^{p-1} \left[D_h\left(x + \frac{t}{p} v, x\right) - C t^p f(x) \right],} \tag{3.13}$$

$$\boxed{H_p(x, r, t) = p t^{p-1} \left[D_{h^*}(\nabla h(x) + t^p r, \nabla h(x)) + C t^p f(x) \right],} \tag{3.14}$$

with corresponding Bregman Euler–Lagrange equation

$$\boxed{\ddot{x}(t) + \frac{p+1}{t} \dot{x}(t) + C p^2 t^{p-2} \left[\nabla^2 h\left(x(t) + \frac{t}{p} \dot{x}(t)\right) \right]^{-1} \nabla f(x(t)) = 0.} \tag{3.15}$$

These parameter functions satisfy the ideal scaling conditions (3.7), and the resulting evolution $x(t)$ satisfies the aforementioned $\mathcal{O}(1/t^p)$ convergence rate,

$$\boxed{f(x(t)) - f(x^*) \;\leq\; \mathcal{O}(1/t^p),} \tag{3.16}$$

where x^* is the desired minimizer of the objective function f.

3.3 Symplectic Accelerated Optimization via Hamiltonian Integrators

3.3.1 Introduction

For simplicity of exposition, we will consider the case where $h(x) = \frac{1}{2}\langle x, x\rangle$. Our new approaches will make use of the adaptive framework developed in Section 2.7.1 via carefully chosen Poincaré transformations. Recalling the discussion of Section 2.7.1, there is usually no equivalent Lagrangian formulation for the future Poincaré transformed systems, so we will work from the Hamiltonian point of view here.

When $h(x) = \frac{1}{2}\langle x, x\rangle$, the Bregman Hamiltonian with parameters α, β, γ given by equation (3.12) for a specific value of $p > 0$ becomes

$$H_p(q, r, t) = \frac{p}{2t^{p+1}}\langle r, r\rangle + Cpt^{2p-1}f(q). \tag{3.17}$$

From Theorem 1.1 in [Wibisono et al., 2016], any solution $q(t)$ to the corresponding Hamilton's equations satisfies the convergence rate

$$f(q(t)) - f(q^*) \;\leq\; \mathcal{O}(1/t^p), \tag{3.18}$$

where q^* is the desired minimizer of the objective function f.

Together with the time-rescaling property of the Bregman family of dynamical systems from Theorem 3.1, this implies that this entire subfamily of Bregman trajectories indexed by the parameter p can be obtained by speeding up or slowing down along the Bregman curve in spacetime corresponding to any specific value of p. We now present two approaches based on the adaptive framework developed in Section 2.7.1 to integrate the Bregman Hamiltonian dynamics, thereby solving the optimization problem, and then compare their performance.

3.3.2 Direct Approach

Our first approach will use the adaptive framework from Section 2.7.1 with monitor function $g(q,r) = 1$ to design a variational integrator for the Bregman Hamiltonian given in equation (3.17) for a given value of $p > 0$. This choice of monitor function will convert the time-dependent Bregman Hamiltonian into an autonomous Hamiltonian in extended phase space. More precisely, given a value of $p > 0$, the time transformation $t \mapsto \tau$ given by

$$\boxed{\frac{dt}{d\tau} = g(q,t,r) = 1} \tag{3.19}$$

generates the **Direct approach Poincaré transformed Hamiltonian**

$$\boxed{\bar{H}_p(\bar{q},\bar{r}) = \frac{p}{2\mathfrak{q}^{p+1}}\langle r,r\rangle + Cp\mathfrak{q}^{2p-1}f(q) + \mathfrak{r},} \tag{3.20}$$

in the phase space with extended coordinates $(\bar{q},\bar{r})$.

The corresponding Hamilton's equations are given by

$$\bar{q}' = \frac{\partial \bar{H}_p}{\partial \bar{r}} = \begin{bmatrix} \frac{p}{\mathfrak{q}^{p+1}} r \\ 1 \end{bmatrix}, \tag{3.21}$$

$$\bar{r}' = -\frac{\partial \bar{H}_p}{\partial \bar{q}} = \begin{bmatrix} -Cp\mathfrak{q}^{2p-1}\nabla f(q) \\ \frac{p(p+1)}{2\mathfrak{q}^{p+2}}\langle r,r\rangle - Cp(2p-1)\mathfrak{q}^{2p-2}f(q) \end{bmatrix}. \tag{3.22}$$

This strategy is equivalent to the usual trick to remove time-dependency consisting of considering time as an additional position variable and adding a corresponding conjugate momentum variable, which is the energy (see [Betancourt et al., 2018] for an example with Hamiltonian given by equation (3.20)). This shows that our adaptive framework from Section 2.7.1 is very general and can also be used for purposes other than solely enforcing a desired variable time-stepping.

3.3.3 Adaptive Approach

Our second approach exploits the time-rescaling property of the Bregman dynamics together with the adaptive framework from Section 2.7.1 with a carefully tuned monitor function. More precisely, we will use adaptivity to transform the Bregman Hamiltonian corresponding to a specific value of $p > 0$ into an autonomous version of the Bregman Hamiltonian corresponding to a smaller value $\mathring{p} < p$ in extended phase space. This will allow us to integrate the higher-order Bregman dynamics corresponding to the value p while benefiting from the computational efficiency of integrating the lower-order Bregman dynamics corresponding to the value $\mathring{p} < p$.

Explicitly, solving equation (3.10) for $\tau(t)$ to transform the Bregman dynamics corresponding to the values of α, β, γ as in equation (3.12) for a given value of p into the Bregman dynamics corresponding to the values of $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$ as in equation (3.12) for a given value $\mathring{p} < p$ yields $\tau(t) = t^{\mathring{p}/p}$. The corresponding monitor function is given by

$$\boxed{\frac{dt}{d\tau} = g(q,t,r) = \frac{p}{\mathring{p}} t^{1-\frac{\mathring{p}}{p}},} \tag{3.23}$$

and generates the **Adaptive approach Poincaré transformed Hamiltonian**

$$\boxed{\bar{H}_{p\to\mathring{p}}(\bar{q},\bar{r}) = \frac{1}{\mathring{p}}\left[\frac{p^2}{2\mathfrak{q}^{p+\frac{\mathring{p}}{p}}}\langle r,r\rangle + Cp^2\mathfrak{q}^{2p-\frac{\mathring{p}}{p}} f(q) + p\mathfrak{r}\mathfrak{q}^{1-\frac{\mathring{p}}{p}}\right].} \tag{3.24}$$

The corresponding Hamilton's equations are given by

$$\bar{q}' = \frac{\partial \bar{H}_{p\to\mathring{p}}}{\partial \bar{r}} = \frac{1}{\mathring{p}}\begin{bmatrix} p^2\mathfrak{q}^{-p-\frac{\mathring{p}}{p}} r \\ p\mathfrak{q}^{1-\frac{\mathring{p}}{p}} \end{bmatrix}, \tag{3.25}$$

$$\bar{r}' = -\frac{\partial \bar{H}_{p\to\mathring{p}}}{\partial \bar{q}} = \frac{1}{\mathring{p}}\begin{bmatrix} -Cp^2\mathfrak{q}^{2p-\frac{\mathring{p}}{p}}\nabla f(q) \\ \frac{p^3+p\mathring{p}}{2\mathfrak{q}^{p+1+\mathring{p}/p}}\langle r,r\rangle - C(2p^3-p\mathring{p})\mathfrak{q}^{2p-\frac{\mathring{p}}{p}-1} f(q) - \frac{p-\mathring{p}}{\mathfrak{q}^{\mathring{p}/p}}\mathfrak{r} \end{bmatrix}. \tag{3.26}$$

3.3.4 Presentation of Numerical Methods

We will now test the performance of the new adaptive framework presented in Section 2.7.1 by implementing variational and non-variational integrators in the case where $\mathcal{X} = \mathbb{R}^d$ and $\langle x, x \rangle = x^\top x$, and we will discuss the results obtained. Keeping in mind the machine learning application where data sets are very large, we will restrict ourselves to explicit first-order optimization algorithms.

Looking at the forms of Hamilton's equations in the Direct and Adaptive approaches, we note that the objective function f and its gradient ∇f only appear in the expression for $\bar{r}'$, and are functions of q only. Looking back at the construction of Hamiltonian Taylor variational integrators from Section 2.4.2, we can note that both the Type II and the Type III approaches require ρ-order Taylor approximations of r and $(\rho+1)$-order Taylor approximations of q. This means that the highest value of ρ that we can choose to obtain a gradient-based algorithm is $\rho = 1$. Now, the starting point of the Type III approach is a $(\rho+1)$-order Taylor approximation $\tilde{q}_0$ of q_0 around q_0. As a consequence, the subsequent steps in the Type III method with $\rho = 0$ and $\rho = 1$ will contain evaluations of the objective function f and its gradient ∇f at this approximation $\tilde{q}_0$. Aside from the inconvenience of the function evaluations not being at the iterates q_0 and q_1 themselves, if f is a nonlinear function, this will also generate nonlinearity in the equations for the updates. As a result, we will not be able to design an explicit algorithm, or at least not a general explicit algorithm that would work for all the functions f considered. On the other hand, the starting point of the Type II approach is a ρ-order Taylor approximation $\tilde{p}_0$ of p_0 around p_0. A similar issue as for the Type III case arises when $\rho = 1$ due to the approximations $(q_{c_i}, p_{c_i}) = \Psi^{(\rho)}_{c_i h}(\tilde{q}_0, p_0)$. Therefore, we cannot design a general explicit algorithm for the Type II case with $\rho = 1$. The remaining possibility is to construct a Type II HTVI using $\rho = 0$. This will produce explicit gradient-based algorithms, where all the evaluations of the objective function f and its gradient ∇f are performed at the iterates q_0 and q_1. Note that when $\rho = 0$, we have $(q_{c_i}, p_{c_i}) = \Psi^{(0)}_{c_i h}(\tilde{q}_0, p_0) = (\tilde{q}_0, p_0)$ for all i, so for given values of p and $\mathring{p}$, every quadrature rule generates the same integrator. Following the method of Section 2.4.2, with time-step h, we will now derive explicit gradient-based Hamiltonian Taylor variational integrators.

Type II Hamiltonian Taylor Variational Integrators (HTVIs) with $\rho = 0$.

As mentioned earlier, since $\rho = 0$, the choice of quadrature rule does not matter, so we can take the rectangular quadrature rule about the initial point ($c_1 = 0$ and $b_1 = 1$). We approximate $\bar{r}(0) = \tilde{\bar{r}}_0$ via $\bar{r}_1 = \pi_{T^*\bar{Q}} \circ \Psi_h^{(0)}(q_0, \tilde{\bar{r}}_0) = \tilde{\bar{r}}_0$ and generate approximations $(\bar{q}_{c_1}, \bar{r}_{c_1}) = \Psi_{c_1 h}^{(0)}(\bar{q}_0, \tilde{\bar{r}}_0) = (\bar{q}_0, \tilde{\bar{r}}_0)$. The Direct Approach and Adaptive Approach HTVIs are then constructed as follows:

(i) **Direct Approach Type II HTVI with $\rho = 0$**

Approximate $\bar{r}(0) = \tilde{\bar{r}}_0$ via $\bar{r}_1 = \pi_{T^*\bar{Q}} \circ \Psi_h^{(0)}(q_0, \tilde{\bar{r}}_0) = \tilde{\bar{r}}_0$.

Approximate

$$\tilde{\bar{q}}_1 \;=\; \pi_{\bar{Q}} \circ \Psi_h^{(1)}(\bar{q}_0, \tilde{\bar{r}}_0) \;=\; \bar{q}_0 + h\dot{\bar{q}}_0 \;=\; \begin{bmatrix} q_0 + h\frac{pr_0}{\mathfrak{q}_0^{p+1}} \\ \mathfrak{q}_0 + h \end{bmatrix}. \tag{3.27}$$

The approximated discrete right Hamiltonian (2.58) is then given by

$$H_d^+(\bar{q}_0, \bar{r}_1; h) = r_1^\top q_0 + \mathfrak{r}_1 \mathfrak{q}_0 + h\left(\frac{p}{2\mathfrak{q}_0^{p+1}} r_1^\top r_1 + Cp\mathfrak{q}_0^{2p-1} f(q_0) + \mathfrak{r}_1\right). \tag{3.28}$$

Solving the implicit discrete right Hamilton's equations

$$q_1 = D_2 H_d^+(q_0, p_1), \qquad p_0 = D_1 H_d^+(q_0, p_1), \tag{3.29}$$

yields the explicit variational integrator

$$\begin{aligned} r_1 &= r_0 - hCp\mathfrak{q}_0^{2p-1}\nabla f(q_0), \\ \mathfrak{r}_1 &= \mathfrak{r}_0 + h\frac{p(p+1)}{2\mathfrak{q}_0^{p+2}} r_1^\top r_1 - hCp(2p-1)\mathfrak{q}_0^{2p-2} f(q_0), \\ q_1 &= q_0 + h\frac{p}{\mathfrak{q}_0^{p+1}} r_1, \\ \mathfrak{q}_1 &= \mathfrak{q}_0 + h. \end{aligned} \tag{3.30}$$

When solving a given optimization problem, we are typically not interested in the evolution of the variable $\mathfrak{r}$, and since it does not affect the other variables, we can instead implement the simpler algorithm

$$\begin{aligned} r_1 &= r_0 - hCp\mathfrak{q}_0^{2p-1}\nabla f(q_0), \\ q_1 &= q_0 + h\frac{p}{\mathfrak{q}_0^{p+1}} r_1, \\ \mathfrak{q}_1 &= \mathfrak{q}_0 + h. \end{aligned} \tag{3.31}$$

(ii) **Adaptive Approach Type II HTVI with $\rho = 0$**

Approximate $\bar{r}(0) = \tilde{\bar{r}}_0$ via $\bar{r}_1 = \pi_{T^*\bar{Q}} \circ \Psi_h^{(0)}(q_0, \tilde{\bar{r}}_0) = \tilde{\bar{r}}_0$.

Approximate

$$\tilde{\bar{q}}_1 \;=\; \pi_{\bar{Q}} \circ \Psi_h^{(1)}(\bar{q}_0, \tilde{\bar{r}}_0) \;=\; \bar{q}_0 + h\dot{\bar{q}}_0 \;=\; \begin{bmatrix} q_0 + h\frac{p^2}{\mathring{p}}\mathfrak{q}_0^{-p-\frac{\mathring{p}}{p}} r_0 \\ \mathfrak{q}_0 + h\frac{p}{\mathring{p}}\mathfrak{q}_0^{1-\frac{\mathring{p}}{p}} \end{bmatrix}. \tag{3.32}$$

The approximated discrete right Hamiltonian (2.58) is then given by

$$H_d^+(\bar{q}_0, \bar{r}_1; h) = r_1^\top \left[q_0 + \frac{p^2}{2\mathring{p}} h\mathfrak{q}_0^{-p-\frac{\mathring{p}}{p}} r_1 \right] + \mathfrak{r}_1 \left[\mathfrak{q}_0 + \frac{p}{\mathring{p}} h\mathfrak{q}_0^{1-\frac{\mathring{p}}{p}} \right] + \frac{Cp^2}{\mathring{p}} h\mathfrak{q}_0^{2p-\frac{\mathring{p}}{p}} f(q_0). \tag{3.33}$$

Solving the implicit discrete right Hamilton's equations

$$q_1 = D_2 H_d^+(q_0, p_1), \qquad p_0 = D_1 H_d^+(q_0, p_1), \tag{3.34}$$

yields the explicit variational integrator

$$\begin{aligned}
r_1 &= r_0 - \frac{p^2}{\mathring{p}} hC\mathfrak{q}_0^{2p-\frac{\mathring{p}}{p}} \nabla f(q_0), \\
\mathfrak{r}_1 &= \left[1 - \frac{h}{\mathfrak{q}_0^{\mathring{p}/p}} \left(1 - \frac{p}{\mathring{p}} \right) \right]^{-1} \left[\mathfrak{r}_0 + \frac{p^3 + p\mathring{p}}{2\mathring{p}\mathfrak{q}_0^{1+p+\mathring{p}/p}} h r_1^\top r_1 + \frac{p\mathring{p} - 2p^3}{\mathring{p}\mathfrak{q}_0^{1-2p+\mathring{p}/p}} hCf(q_0) \right], \\
q_1 &= q_0 + \frac{p^2}{\mathring{p}} h\mathfrak{q}_0^{-p-\frac{\mathring{p}}{p}} r_1, \\
\mathfrak{q}_1 &= \mathfrak{q}_0 + \frac{p}{\mathring{p}} h\mathfrak{q}_0^{1-\frac{\mathring{p}}{p}}.
\end{aligned} \tag{3.35}$$

When solving a given optimization problem, we are typically not interested in the evolution of the variable $\mathfrak{r}$, and since it does not affect the other variables, we can instead implement the simpler algorithm

$$\begin{aligned}
r_1 &= r_0 - \frac{p^2}{\mathring{p}} hC\mathfrak{q}_0^{2p-\frac{\mathring{p}}{p}} \nabla f(q_0), \\
q_1 &= q_0 + \frac{p^2}{\mathring{p}} h\mathfrak{q}_0^{-p-\frac{\mathring{p}}{p}} r_1, \\
\mathfrak{q}_1 &= \mathfrak{q}_0 + \frac{p}{\mathring{p}} h\mathfrak{q}_0^{1-\frac{\mathring{p}}{p}}.
\end{aligned} \tag{3.36}$$

Different first-order optimization algorithms based on variational integrators can be constructed for the Poincaré transformed Hamiltonians, such as prolongation-collocation variational integrators [Leok and Shingel, 2012a], and higher-order Hamiltonian Taylor variational integrators [Schmitt et al., 2018], and Galerkin variational integrators [Leok and Zhang, 2011]. We will not consider these integrators here since they require that one solves systems of nonlinear equations and cannot be implemented explicitly in general. Note however that in practice, implicit methods for the numerical solution of ODEs that can be solved using fixed-point iterations (as opposed to Newton iterations) can be quite competitive, as there is a good initial guess which may allow them to converge in a small number of iterations, and the iterations are inexpensive as they do not require the assembly of a Jacobian. The convergence of the fixed-point iteration depends on the conditioning of the system of equations, and may impose a stringent time-step restriction. This numerical conditioning issue can be overcome by the use of exponential integrators [Hochbruck and Ostermann, 2010], and in particular, symplectic and energy-preserving exponential integrators [Shen and Leok, 2019].

We have also implemented non-variational symplectic methods for the Poincaré Hamiltonian from these Direct and Adaptive approaches.

Non-Variational Symplectic Integrators for the Poincaré Hamiltonians.

(i) **Direct and Adaptive Approaches with Splitting of the Hamiltonian.**
The Direct approach with splitting of the Hamiltonian is the approach presented in [Betancourt et al., 2018]. The three components of the Poincaré transformed Hamiltonian (3.20) are considered separately. These components generate dynamics in the extended phase space via six vector fields, and a symmetric leapfrog composition of the corresponding component dynamics is constructed to obtain a symplectic integrator (referred to in this dissertation as "Direct Splitting"):

$$
\begin{aligned}
&t \leftarrow t + \frac{h}{2},\\
&\mathfrak{r} \leftarrow \mathfrak{r} + \frac{h}{2}\frac{p(p+1)}{2t^{p+2}} r^\top r - \frac{h}{2} Cp(2p-1)t^{2p-2} f(q),\\
&r \leftarrow r - \frac{h}{2} Cpt^{2p-1} \nabla f(q),\\
&q \leftarrow q + h\frac{p}{t^{p+1}} r,\\
&r \leftarrow r - \frac{h}{2} Cpt^{2p-1} \nabla f(q),\\
&\mathfrak{r} \leftarrow \mathfrak{r} + \frac{h}{2}\frac{p(p+1)}{2t^{p+2}} r^\top r - \frac{h}{2} Cp(2p-1)t^{2p-2} f(q),\\
&t \leftarrow t + \frac{h}{2}.
\end{aligned}
\tag{3.37}
$$

A new symplectic integrator (referred to in this dissertation as "Adaptive Splitting") can also be obtained by adapting the approach presented in [Betancourt et al., 2018] to the adaptive Poincaré transformed Hamiltonian (3.24):

$$
\begin{aligned}
&t \leftarrow \left(t^{\mathring{p}/p} + \frac{h}{2}\right)^{p/\mathring{p}},\\
&\theta \leftarrow \frac{\mathring{p}}{\mathring{p}-p}\left[\left(\frac{p^3}{2\mathring{p}} + \frac{p}{2}\right) t^{-p-1} r^\top r + \left(p - \frac{2p^3}{\mathring{p}}\right) t^{2p-1} Cf(q)\right],\\
&\mathfrak{r} \leftarrow (\mathfrak{r} + \theta)\exp\left(\left(\left(1 - \frac{p}{\mathring{p}}\right)\frac{h}{2} t^{-\mathring{p}/p}\right) - \theta,\right.\\
&r \leftarrow r - h\frac{Cp^2}{2\mathring{p}} t^{2p-\mathring{p}/p} \nabla f(q),\\
&q \leftarrow q + h\frac{p^2}{\mathring{p} t^{p+\mathring{p}/p}} r,\\
&r \leftarrow r - h\frac{Cp^2}{2\mathring{p}} t^{2p-\mathring{p}/p} \nabla f(q),\\
&\theta \leftarrow \frac{\mathring{p}}{\mathring{p}-p}\left[\left(\frac{p^3}{2\mathring{p}} + \frac{p}{2}\right) t^{-p-1} r^\top r + \left(p - \frac{2p^3}{\mathring{p}}\right) t^{2p-1} Cf(q)\right],\\
&\mathfrak{r} \leftarrow (\mathfrak{r} + \theta)\exp\left(\left(\left(1 - \frac{p}{\mathring{p}}\right)\frac{h}{2} t^{-\mathring{p}/p}\right) - \theta,\right.\\
&t \leftarrow \left(t^{\mathring{p}/p} + \frac{h}{2}\right)^{p/\mathring{p}}.
\end{aligned}
\tag{3.38}
$$

As before, we are typically not interested in the evolution of the variable $\mathfrak{r}$, and since it does not affect the other variables, we can instead implement the simpler "Direct Splitting" algorithm

$$\begin{aligned} t &\leftarrow t + \frac{h}{2}, \\ r &\leftarrow r - \frac{h}{2} C p t^{2p-1} \nabla f(q), \\ q &\leftarrow q + h \frac{p}{t^{p+1}} r, \\ r &\leftarrow r - \frac{h}{2} C p t^{2p-1} \nabla f(q), \\ t &\leftarrow t + \frac{h}{2}, \end{aligned} \tag{3.39}$$

and the simpler "Adaptive Splitting" algorithm

$$\begin{aligned} t &\leftarrow \left(t^{\mathring{p}/p} + \frac{h}{2} \right)^{p/\mathring{p}}, \\ r &\leftarrow r - h \frac{C p^2}{2\mathring{p}} t^{2p-\mathring{p}/p} \nabla f(q), \\ q &\leftarrow q + h \frac{p^2}{\mathring{p} t^{p+\mathring{p}/p}} r, \\ r &\leftarrow r - h \frac{C p^2}{2\mathring{p}} t^{2p-\mathring{p}/p} \nabla f(q), \\ t &\leftarrow \left(t^{\mathring{p}/p} + \frac{h}{2} \right)^{p/\mathring{p}}. \end{aligned} \tag{3.40}$$

This method will be presented in greater generality and details in Section 5.3.1.

(ii) **Phase Space Cloning and Splitting.**

A very natural approach to integrate nonseparable Hamiltonian dynamics consists in defining a new Hamiltonian via two copies of the original Hamiltonian in a phase space of dimension twice as large [Pihajoki, 2015]:

$$\tilde{H}(q, \tilde{q}, p, \tilde{p}) = H_1(q, \tilde{p}) + H_2(\tilde{q}, p), \tag{3.41}$$

where $H_1 = H_2 = H$. Hamilton's equations are then given by

$$\dot{q} = \nabla_p H_2, \qquad \dot{\tilde{q}} = \nabla_{\tilde{p}} H_1, \qquad \dot{p} = -\nabla_q H_1, \qquad \dot{\tilde{p}} = -\nabla_{\tilde{q}} H_2. \tag{3.42}$$

We can then use symmetric symplectic splittings to integrate this new Hamiltonian system explicitly:

$$\exp\left(h\tilde{H}\right) = \prod_{k=1}^{K} \exp\left(c_k h H_1\right) \exp\left(d_k h H_2\right) + \mathcal{O}(h^{r+1}), \qquad d_K = 0. \tag{3.43}$$

Note that we can exchange the roles of H_1 and H_2 in the splitting to get a different integrator. Letting $c_0 = d_0 = 0$, the above integrator can be implemented via

while *convergence criterion is not met* **do**
  $q \leftarrow q_n, \quad \tilde{q} \leftarrow \tilde{q}_n, \quad p \leftarrow p_n, \quad \tilde{p} \leftarrow \tilde{p}_n$
  for $k = 1$ *to* K **do**
    $q \leftarrow q + c_k h \nabla_p H_2(\tilde{q}, p)$ and $\tilde{p} \leftarrow \tilde{p} - c_k h \nabla_{\tilde{q}} H_2(\tilde{q}, p)$
    $\tilde{q} \leftarrow \tilde{q} + d_k h \nabla_{\tilde{p}} H_1(q, \tilde{p})$ and $p \leftarrow p - d_k h \nabla_q H_1(q, \tilde{p})$
  $q_{n+1} \leftarrow q + c_K h \nabla_p H_2(\tilde{q}, p), \qquad \tilde{q}_{n+1} \leftarrow \tilde{q}$
  $\tilde{p}_{n+1} \leftarrow \tilde{p} - c_K h \nabla_{\tilde{q}} H_2(\tilde{q}, p), \qquad p_{n+1} \leftarrow p$

Note that when applied to the Poincaré transformed Hamiltonians for optimization, this splitting will require a significantly larger number of evaluations of the objective function f and of its gradient ∇f at each step.

The **Strang Splitting** ([Strang, 1968]) is the splitting of order $r = 2$, obtained with

$$K = 2, \qquad \text{and} \qquad c_1 = c_2 = \frac{1}{2}, \qquad d_1 = 1, \qquad d_2 = 0. \tag{3.44}$$

This method will be referred to as "CloningStrang" in this dissertation.

The **Yoshida 4 Splitting** ([Yoshida, 1990]) is the splitting of order $r = 4$, obtained with $K = 4$, and

$$c_1 = c_4 = \frac{1}{4 - 2^{4/3}}, \qquad c_2 = c_3 = \frac{1 - 2^{1/3}}{4 - 2^{4/3}}, \tag{3.45}$$

$$d_1 = d_3 = \frac{1}{2 - 2^{1/3}}, \qquad d_2 = \frac{1}{1 - 2^{2/3}} \qquad d_4 = 0. \tag{3.46}$$

This method will be referred to as "CloningY4" in this dissertation.

The **Yoshida 6 Splitting** ([Yoshida, 1990]) is the splitting of order $r = 6$, obtained with $K = 10$, $d_{10} = 0$ and

$$x_0 = \frac{1}{1 - 2^{2/3}}, \qquad x_1 = \frac{1}{2 - 2^{1/3}}, \qquad y_0 = \frac{1}{1 - 2^{4/5}}, \qquad y_1 = \frac{1}{2 - 2^{1/5}}, \tag{3.47}$$

$$d_1 = d_3 = d_7 = d_9 = x_1 y_1, \quad d_2 = d_8 = x_0 y_1, \quad d_4 = d_6 = x_1 y_0, \quad d_5 = x_0 y_0, \tag{3.48}$$

$$c_1 = \frac{1}{2} d_1, \qquad c_{10} = \frac{1}{2} d_9, \qquad c_k = \frac{1}{2}(d_{k-1} + d_k) \quad \text{for} \quad k = 2, 3, ..., 9. \tag{3.49}$$

This method will be referred to as "CloningY6" in this dissertation.

Lastly, we have also implemented non-symplectic methods based on the Direct and Adaptive approaches (the Classical 4th Order Explicit Runge–Kutta Method (RK4), and MATLAB's explicit adaptive ODE solvers `ode23`, `ode45`), and other popular optimization algorithms have been tested as well such as Nesterov's Accelerated Gradient (3.1), Trust Region Steepest Descent, ADAM [Kingma and Ba, 2014], AdaGrad [Duchi et al., 2011], and RMSprop [Tieleman and Hinton, 2012].

3.3.5 Numerical Results

We have implemented the numerical methods presented earlier in this section to test their performance on the problem of minimizing the quartic function

$$f(x) = \left[(x - 1)^\top \Sigma (x - 1)\right]^2, \tag{3.50}$$

where x is a high-dimensional variable and the matrix Σ is obtained via $\Sigma_{ij} = 0.9^{|i-j|}$. This convex function achieves its minimum value 0 at $x^* = 1$.

Unless specified otherwise, the termination criterion used in this section is

$$|f(x_k) - f(x_{k-1})| < \delta \quad \text{and} \quad \|\nabla f(x_k)\| < \delta, \qquad \text{where} \quad \delta = 10^{-10}. \tag{3.51}$$

Direct versus Adaptive approach

Numerical experiments conducted with all the symplectic algorithms presented earlier showed that a carefully tuned Adaptive approach enjoys a significantly better rate of convergence and a much smaller number of steps required to achieve convergence than the Direct approach, as can be seen in Figure 3.1 and Table 3.1 for the HTVI and CloningY4 methods. Although the Adaptive approach requires a smaller fictive h than the Direct approach, the physical time-steps resulting from $t = \tau^{p/\mathring{p}}$ in the Adaptive approach grow rapidly to values larger than the physical time-step of the Direct approach.

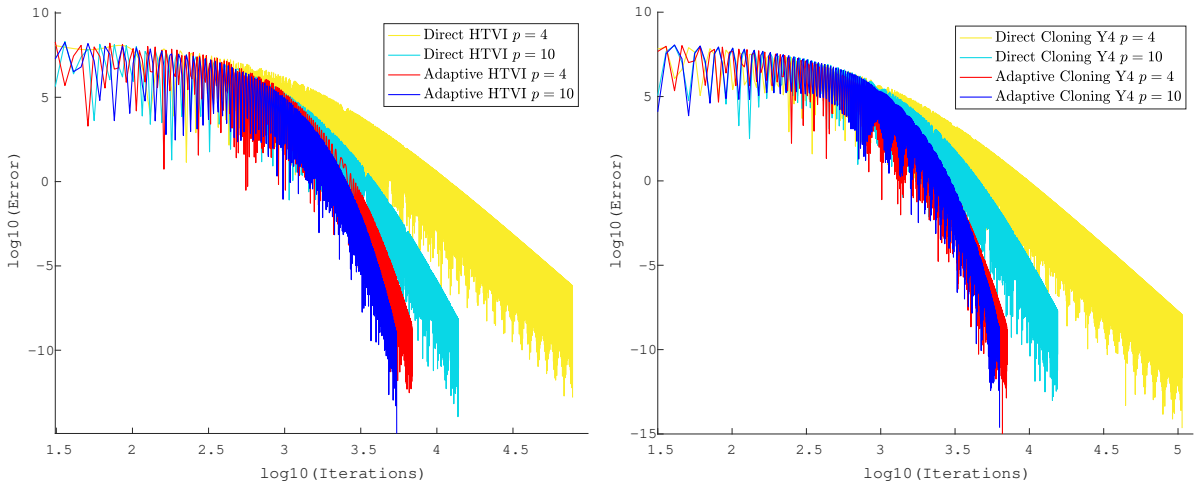


Figure 3.1: Comparison of the rates of convergence between the Direct and Adaptive approach for the HTVI method and the Cloning method with a Yoshida 4 splitting. We can clearly see that the Adaptive approach outperforms the Direct approach.

Table 3.1: Comparison of the Direct and Adaptive approach for the HTVI method and the Cloning method with a Yoshida 4 splitting. The Adaptive approach clearly outperforms the Direct approach in terms of number of iterations required.

Approach	p	$\mathring{p}$	h	**Iterations**	**Approach**	p	$\mathring{p}$	h	**Iterations**
Direct HTVI	4	-	8.00E-04	77 878	Direct CloneY4	4	-	9.00E-04	106 530
Adap. HTVI	4	0.5	1.21E-04	6 867	Adap. CloneY4	4	0.5	1.15E-04	7 101
Direct HTVI	10	-	4.00E-04	13 872	Direct CloneY4	10	-	3.30E-04	15 498
Adap. HTVI	10	0.5	1.95E-05	5 564	Adap. CloneY4	10	0.5	1.62E-05	6 300

Comparison of Methods within the Direct and Adaptive approaches

Numerical experiments were conducted to compare the various algorithms presented in Section 3.3.4, and the results are presented in Figure 3.2 and Table 3.2.

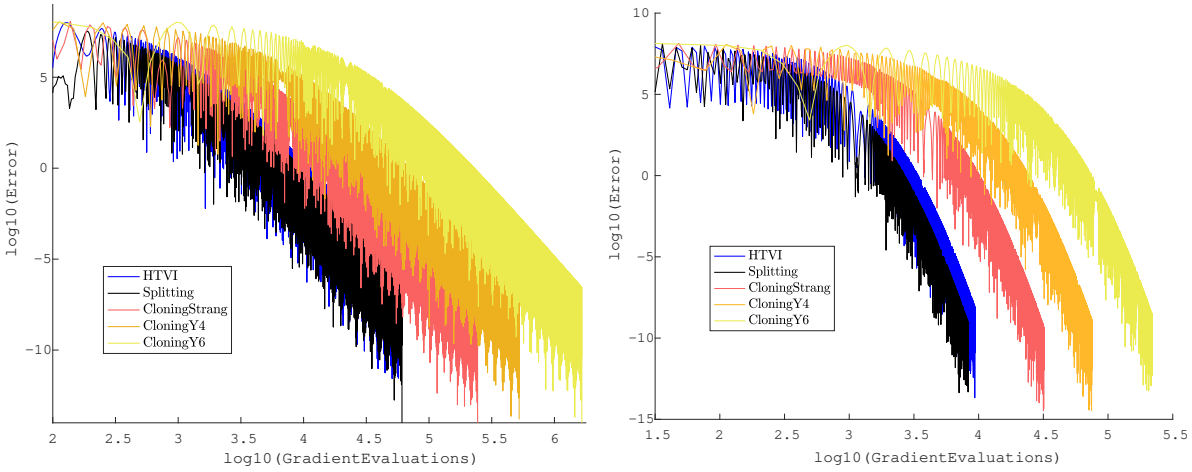


Figure 3.2: Comparison of the convergence, in terms of gradient evaluations needed, of the different symplectic integrators within the Direct approach (on the left), and within the Adaptive approach (on the right).

Table 3.2: Number of iterations needed until convergence of the different symplectic integrators within the Direct (on the left) and Adaptive (on the right) approaches.

Method	p	h	**Iterations**	**Method**	p	$\mathring{p}$	h	**Iterations**
Direct HTVI	4	8.7E-04	57 504	Adaptive HTVI	4	1	2.4E-04	9 361
Direct Splitting	4	9.5E-04	60 881	Adaptive Splitting	4	1	2.9E-04	8 313
Direct CloneStrang	4	9.7E-04	81 367	Adaptive CloneStrang	4	1	2.4E-04	10 638
Direct CloneY4	4	8.9E-04	74 747	Adaptive CloneY4	4	1	2.9E-04	10 721
Direct CloneY6	4	7.9E-04	87 075	Adaptive CloneY6	4	1	2.0E-04	11 549

Although the number of iterations for all methods were of the same order of magnitude, the HTVI methods and the Splitting methods performed much better than the methods based on the phase-space Cloning idea of [Pihajoki, 2015]. This is mostly due to the fact that these phase-space Cloning methods require several evaluations of the objective function f and of its gradient ∇f at each iteration (3 for Strang splitting, 7 for Yoshida's 4th order splitting, and 19 for Yoshida's 6th order splitting), while the HTVI and Splitting methods only required one such evaluation at each iteration. As a

result, these phase space cloning methods also required much more computational time to achieve convergence. It might be possible to improve the performance of these phase space cloning methods by adding an extra term in the final Hamiltonian which binds the two copies of the Poincaré transformed Hamiltonian, as was done in [Tao, 2016]. However, this additional term is likely to require a larger number of compositions when splitting the final Hamiltonian, which would require more gradient evaluations of the objective function at each step. Thus, even though the trick presented in [Tao, 2016] could reduce the number of iterations required to achieve convergence, it seems very unlikely that the resulting algorithm would be competitive against the HTVI and the Splitting methods, in terms of computational time and total number of gradient evaluations needed.

Dependence on p and $\mathring{p}$ in the Adaptive approach

We conducted numerical experiments with the HTVI method to study the evolution of the performance of the Adaptive approach as the parameters p and $\mathring{p}$ are varied. We can see from the results presented in Figure 3.3 and Table 3.3 that the Adaptive HTVI method becomes more and more efficient as p is increased and $\mathring{p}$ is decreased. The improvement in efficiency is very important as we increase p from $p = 2$ to $p = 4$, while it is minor but still noticeable as we increase p from $p = 4$ to $p = 8$. A possible explanation for this behavior is that the integrator might not be of high enough order to distinguish between the $p = 6$ and $p = 8$ Bregman dynamics. Note that the fictive time-step h must be reduced as p increases or $\mathring{p}$ decreases, but the time relation $t = \tau^{p/\mathring{p}}$ ensures that the resulting physical time-steps do not become significantly smaller.

Table 3.3: Evolution of the fictive time-step h and number of iterations until convergence for the HTVI method as p increases (left), and as $\mathring{p}$ decreases (right).

Method	p	$\mathring{p}$	h	**Iterations**	**Method**	p	$\mathring{p}$	h	**Iterations**
Adaptive HTVI	2	1	8.0E-04	77 855	Adaptive HTVI	4	2	3.8E-04	22 128
Adaptive HTVI	4	1	2.4E-04	9 361	Adaptive HTVI	4	1	2.4E-04	9 361
Adaptive HTVI	6	1	9.4E-05	7 785	Adaptive HTVI	4	0.5	1.2E-04	7 099
Adaptive HTVI	8	1	6.1E-05	6 133	Adaptive HTVI	4	0.1	2.4E-05	5 689

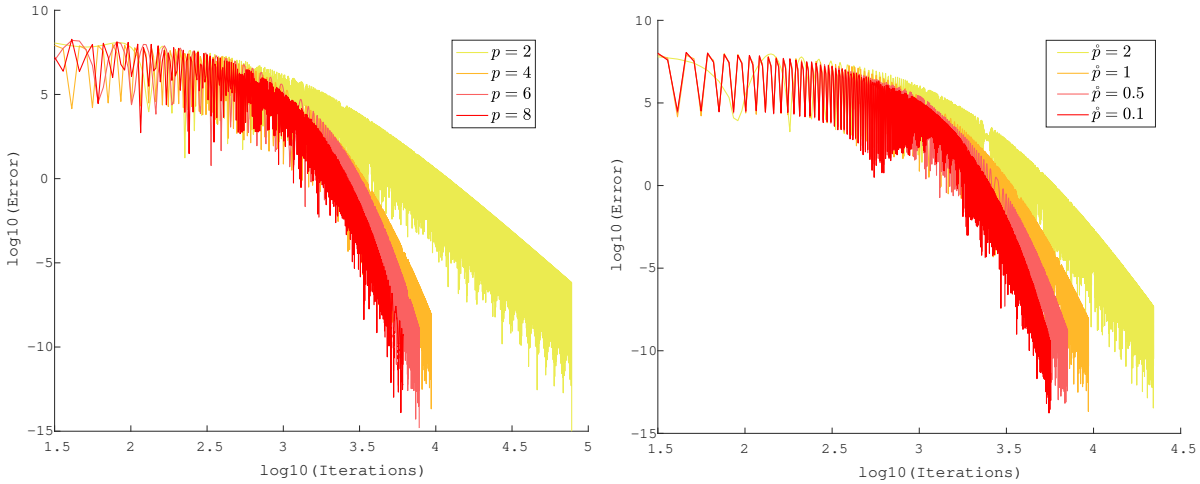


Figure 3.3: Evolution of the rates of convergence of the HTVI method as p is increased (on the left), and as $\mathring{p}$ is decreased (on the right).

Comparison to non-symplectic integrators

We now investigate the role that symplecticity plays when integrating the Bregman dynamics. We first implemented fixed time-step integrators such as the 4th-order explicit Runge–Kutta method

$$u_{k+1} = u_k + \frac{h}{6}(\xi_1 + 2\xi_2 + 2\xi_3 + \xi_4), \qquad \text{with} \quad u = [\bar{q}, \bar{r}]^\top, \tag{3.52}$$

where

$$\xi_1 = F(u_k), \quad \xi_2 = F\left(u_k + \frac{1}{2}\xi_1\right), \quad \xi_3 = F\left(u_k + \frac{1}{2}\xi_2\right), \quad \xi_4 = F\left(u_k + \xi_3\right), \tag{3.53}$$

and $F(u) = \left[\frac{\partial \bar{H}}{\partial \bar{r}}, -\frac{\partial \bar{H}}{\partial \bar{q}}\right]^\top$, but these failed to converge both in the Direct and Adaptive approaches. The reason why convergence cannot be achieved may have to do with the nonautonomous aspect of the differential equation. More precisely, explicit Runge–Kutta methods are conditionally stable, where stability intervals for the time-steps depend on the expansivity of the differential equation. Since the differential equations considered here are not autonomous, the stability intervals are time-dependent, and thus any fixed choice of time-step may eventually violate the stability condition. It might be possible to achieve low accuracy convergence using these methods, but the fact that they cannot achieve higher accuracy and are likely to lose stability eventually makes them undesirable.

We then considered variable time-step explicit Runge–Kutta methods. To this end, we tested the differential equation solvers `ode45` and `ode23` of `MATLAB`, which are explicit variable time-step Runge–Kutta pairs, and the corresponding numerical results are presented in Figure 3.4.

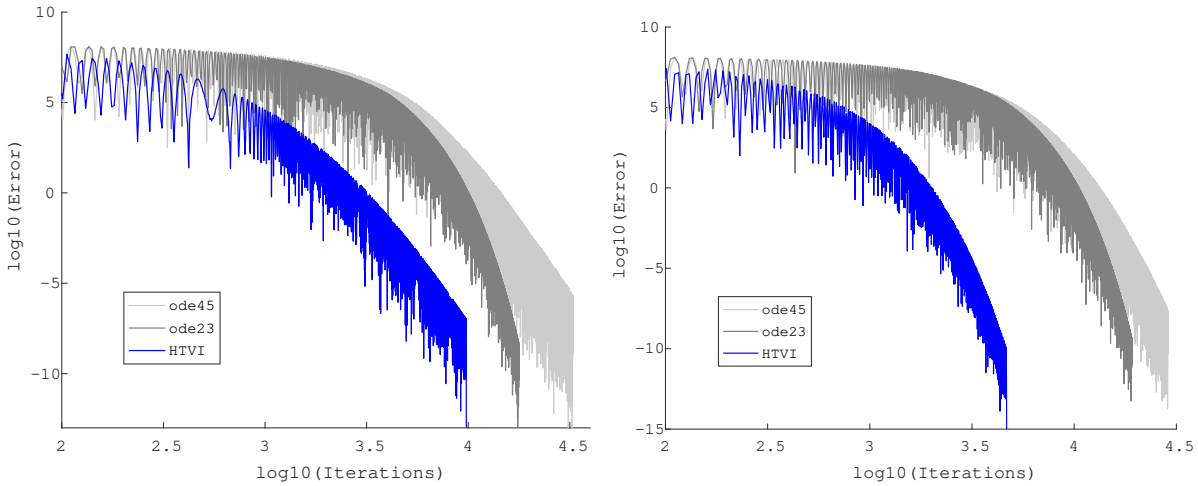


Figure 3.4: Comparison of the HTVI method with the `ode23` and `ode45` MATLAB functions in the Direct (left) and Adaptive (right) approaches with $p = 10$ and $\mathring{p} = 0.5$. The HTVI method requires outperforms the MATLAB solvers.

The HTVI method required a significantly smaller number of iterations than the MATLAB solvers. Furthermore, an inherent part of the time-step control in embedded Runge–Kutta methods is that, at each iteration, the underlying Runge–Kutta method may be executed several times to determine the appropriate time-step that satisfies the prescribed tolerances. For this reason, the MATLAB solvers require more evaluations of f and ∇f at each iteration, and since they also required more iterations than the HTVI method, these MATLAB solvers are much less competitive.

It should also be noted that the MATLAB solvers did not exhibit any improvements when used with the Adaptive approach instead of the Direct approach, while the HTVI method improved significantly. This is not surprising since the MATLAB solvers `ode23` and `ode45` both use a variable time-step strategy, regardless of the approach chosen.

Note that our Adaptive approach and the embedded Runge–Kutta methods use adaptivity in two fundamentally different ways. Our approach uses *a priori* adaptivity based on known global properties of the family of differential equations considered (i.e. the time-translation symmetry of the family of Bregman dynamics). In contrast, embedded Runge-Kutta methods use adaptivity based on *a posteriori* local error estimates. This distinction could explain why the embedded Runge–Kutta methods do not perform as well as our Adaptive approach: *a posteriori* estimators might focus mostly on the fast local oscillations of the Bregman dynamics and not on the slower global decay, and these fast oscillations might be forcing the embedded Runge–Kutta methods to adaptively take smaller time-steps than necessary.

We can also see from Figure 3.4 that for both the symplectic and non-symplectic adaptive methods, a significant number of iterations are needed before the error effectively starts decaying. The fact that this slow initial behavior persists with those two approaches, which use time-adaptivity in the two fundamentally different ways described in the previous paragraph, suggests that this behavior might be intrinsic to the continuous Bregman trajectory being discretized and that time-adaptivity might not be able to help accelerate this initial phase.

Comparison to popular optimization methods

Finally, we have compared the performance of our Adaptive HTVI algorithm to that of Nesterov's Accelerated Gradient (NAG) (3.1) with the same initial time-step $h = 2 \times 10^{-6}$, and of popular adaptive optimization algorithms such as Trust Region Steepest Descent (TRUST), ADAM [Kingma and Ba, 2014], AdaGrad [Duchi et al., 2011], and RMSprop [Tieleman and Hinton, 2012].

Figure 3.5 and Table 3.4 present the numerical results obtained when applying these algorithms to the quartic objective function (3.50). Although the Adaptive HTVI method is not the most efficient method, it significantly outperformed certain popular optimization algorithms on this particular convex problem. This suggests that the Adaptive HTVI method might be a competitive first-order explicit algorithm, and that it might be worth considering it as one of several possible options to use in practice, as the relative performance often depends on the specific choice of objective function.

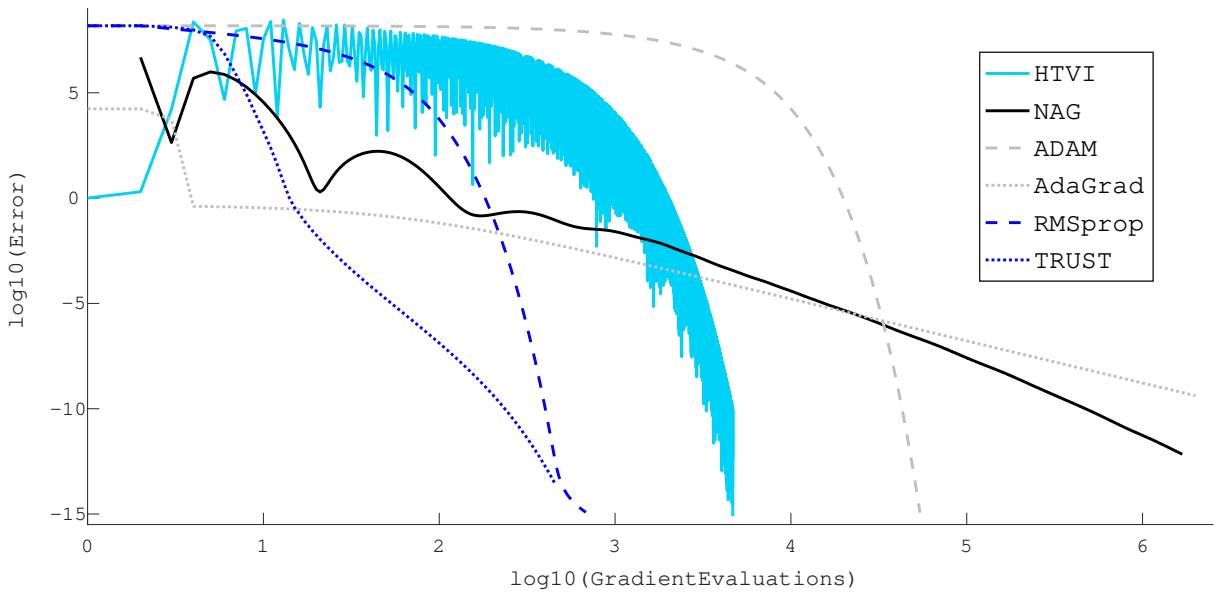


Figure 3.5: Comparison of the computational efficiency of HTVI, NAG, and other popular adaptive optimization algorithms (ADAM, AdaGrad, RMSprop, Trust Region Steepest Descent (TRUST)) to achieve convergence on the quartic objective function (3.50). Note that HTVI and NAG were implemented with the same initial time-step.

Table 3.4: Comparison of the number of iterations needed for HTVI, for NAG, and for other popular adaptive optimization algorithms (ADAM, AdaGrad, RMSprop, Trust Region Steepest Descent (TRUST)) to achieve convergence on the quartic objective function (3.50), with different values of δ as termination criterion (3.51). Note that HTVI and NAG were implemented with the same initial time-step. For all these algorithms, the number of gradient evaluations equals the number of iterations.

$\delta =$	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
HTVI	2 182	2 486	2 750	3 233	3 434	3 593	4 014	4 097	4 566
NAG	4 143	10 949	27 660	65 724	154 258	341 928	745 292	1.7E6	>1E10
ADAM	29 665	32 733	35 802	38 871	41 939	45 008	48 076	51 145	54 215
AdaGrad	520 482	2.4E06	1.1E07	5.2E07	2.4E08	—	—	—	—
RMSprop	276	305	334	363	393	422	452	498	682
TRUST	32	48	71	106	154	215	288	366	455

Remark 3.1. *In [Betancourt et al., 2018], it was observed that Nesterov's Accelerated Gradient algorithm transitions into an exponential rate of convergence once it is sufficiently close to the minimum of certain objective functions, and suggested that this behavior requires strong convexity of the objective function in the neighborhood of the minimum. Similarly to the strategy presented in [Betancourt et al., 2018], a gradient flow can be incorporated into the updates of the algorithms presented here so that for certain objective functions, the same exponential rate of convergence can be achieved close to the minimizer.*

Remark 3.2. *In high-dimensional nonconvex optimization problems of practical interest, it has been observed empirically that a main source of difficulty is not resulting from the presence of local minima but rather from the ubiquity of saddle points surrounded by high error plateaux [Dauphin et al., 2014; Jin et al., 2021]. These numerous saddle points can significantly slow down gradient-based algorithms and give the illusion of the existence of local minima. It was demonstrated in [Jin et al., 2018] via a variant of Nesterov's Accelerated Gradient algorithm that momentum techniques can escape saddle points faster than standard gradient methods and can thereby accelerate convergence in the nonconvex setting as well. This suggests that the variational framework for accelerated optimization and our Adaptive approach to obtain symplectic optimization algorithms may also be promising with regards to nonconvex optimization.*

Conclusion. We used our adaptive framework together with the variational approach to accelerated optimization presented in [Wibisono et al., 2016] to design efficient explicit Hamiltonian integrators for symplectic accelerated optimization. We observed that a careful use of adaptivity and symplecticity can result in better algorithms: time-adaptive Hamiltonian variational discretizations, which are automatically symplectic, with adaptive time-steps informed by the time-rescaling of the Bregman family of dynamics yielded the most robust and computationally efficient optimization algorithms, outperforming fixed-timestep symplectic discretizations, adaptive-timestep non-symplectic discretizations, and Nesterov's accelerated gradient algorithm which is neither time-adaptive nor symplectic.

We will discuss more practical considerations that can be implemented to improve the performance of our optimization algorithms in Chapter 5.

3.4 Symplectic Accelerated Optimization via Lagrangian Integrators

Time-Adaptive Lagrangian Taylor Variational Integrator

We now construct a time-adaptive Lagrangian Taylor variational integrator (LTVI) for the p-Bregman Lagrangian,

$$L_p(q, q', \mathfrak{q}) = \frac{\mathfrak{q}^{p+1}}{2p}\langle q', q'\rangle - Cp\mathfrak{q}^{2p-1} f(q), \tag{3.54}$$

using the strategy outlined in Section 2.4.2 together with the two sets of discrete Euler–Lagrange equations derived in Section 2.7.2.

Looking at the form of the continuous p-Bregman Euler–Lagrange equations,

$$\ddot{q} + \frac{p+1}{\mathfrak{q}}\dot{q} + Cp^2\mathfrak{q}^{p-2}\nabla f(q) = 0, \tag{3.55}$$

we can note that ∇f appears in the expression for $\ddot{q}$. Now, the construction of a Lagrangian Taylor variational integrator as presented in Section 2.4.2 requires ρ-order and $(\rho+1)$-order Taylor approximations of q. This means that if we take $\rho \geq 1$, then ∇f and higher-order derivatives of f will appear in the resulting discrete Lagrangian L_d, and as a consequence, the discrete Euler–Lagrange equations,

$$p_0 = -D_1 L_d(q_0, q_1), \qquad p_1 = D_2 L_d(q_0, q_1), \tag{3.56}$$

will yield an integrator which is not gradient-based. Keeping in mind the machine learning applications where data sets are very large, we will restrict ourselves to explicit first-order optimization algorithms, and therefore the highest value of ρ that we can choose to obtain a gradient-based algorithm is $\rho = 0$. Now, with $\rho = 0$, the choice of quadrature rule does not matter, so we can take the rectangular quadrature rule about the initial point (i.e., $c_1 = 0$ and $b_1 = 1$).

We first approximate $\dot{\bar{q}}(0) = \bar{v}_0$ by the solution $\tilde{\bar{v}}_0$ of the problem

$$\bar{q}_1 = \pi_Q \circ \Psi_h^{(1)}(\bar{q}_0, \tilde{\bar{v}}_0) = \bar{q}_0 + h\tilde{\bar{v}}_0, \tag{3.57}$$

that is $\tilde{\bar{v}}_0 = \frac{\bar{q}_1 - \bar{q}_0}{h}$. Then, applying the rectangular quadrature rule about the initial point gives the associated discrete Lagrangian,

$$L_d(\bar{q}_0, \bar{q}_1) \;=\; hL_p\left(q_0, \frac{\tilde{v}_0}{g(\mathfrak{q}_0)}, \mathfrak{q}_0\right) \;=\; \frac{\mathfrak{q}_0^{p+1}}{2p(g(\mathfrak{q}_0))^2} h\langle \tilde{v}_0, \tilde{v}_0\rangle - Chp\mathfrak{q}_0^{2p-1} f(q_0). \tag{3.58}$$

A variational integrator can then be defined using one of the two sets of discrete extended Euler–Lagrange equations derived in Section 2.7.2. In practice, we are not interested in the evolution of the conjugate momentum $\mathfrak{r}$, and since it will not appear in the updates for the other variables, the two sets of discrete extended Euler–Lagrange equations derived in Section 2.7.2 both reduce to the same updates,

$$\begin{aligned} r_k &= -D_1 L_d(q_k, \mathfrak{q}_k, q_{k+1}, \mathfrak{q}_{k+1}), \\ r_{k+1} &= \frac{g(\mathfrak{q}_k)}{g(\mathfrak{q}_{k+1})} D_3 L_d(q_k, \mathfrak{q}_k, q_{k+1}, \mathfrak{q}_{k+1}), \\ \mathfrak{q}_{k+1} &= \mathfrak{q}_k + hg(\mathfrak{q}_k). \end{aligned} \tag{3.59}$$

For the adaptive approach, substituting the monitor function

$$g(\mathfrak{q}) = \frac{p}{\mathring{p}} \mathfrak{q}^{1-\frac{\mathring{p}}{p}} \tag{3.60}$$

and the discrete Lagrangian

$$L_d(q_k, \mathfrak{q}_k, q_{k+1}, \mathfrak{q}_{k+1}) = \frac{\mathring{p}^2}{2hp^3} \mathfrak{q}_k^{p-1+2\mathring{p}/p} \langle q_{k+1} - q_k, q_{k+1} - q_k\rangle - Chp\mathfrak{q}_0^{2p-1} f(q_k), \tag{3.61}$$

yields the **Adaptive LTVI** algorithm,

$$\begin{aligned} \mathfrak{q}_{k+1} &= \mathfrak{q}_k + h\frac{p}{\mathring{p}} \mathfrak{q}_k^{1-\mathring{p}/p}, \\ q_{k+1} &= q_k + \frac{hp^3}{\mathring{p}^2 \mathfrak{q}_k^{p-1+2\mathring{p}/p}} r_k - \frac{Ch^2 p^4}{\mathring{p}^2} \mathfrak{q}_k^{p-2\mathring{p}/p} \nabla f(q_k), \\ r_{k+1} &= \frac{\mathring{p}^2 \mathfrak{q}_k^{p+\mathring{p}/p}}{hp^3 \mathfrak{q}_{k+1}^{1-\mathring{p}/p}} (q_{k+1} - q_k). \end{aligned} \tag{3.62}$$

In the direct approach, $\mathring{p} = p$ so $g(\mathfrak{q}) = 1$ and we obtain the **Direct LTVI** algorithm,

$$\begin{aligned}
\mathfrak{q}_{k+1} &= \mathfrak{q}_k + h, \\
q_{k+1} &= q_k + \frac{hp}{\mathfrak{q}_k^{p+1}} r_k - Ch^2 p^2 \mathfrak{q}_k^{p-2} \nabla f(q_k), \\
r_{k+1} &= \frac{\mathfrak{q}_k^{p+1}}{hp}(q_{k+1} - q_k).
\end{aligned} \tag{3.63}$$

An Analogous Hamiltonian Taylor Variational Integrator (HTVI)

In [Duruisseaux et al., 2021] and in Section 3.3.4, a Type II Hamiltonian Taylor Variational Integrator (HTVI) was derived following the strategy from Section 2.4.2 with $\rho = 0$ for the adaptive approach $p \to \mathring{p}$-Bregman Hamiltonian,

$$\bar{H}_{p\to\mathring{p}}(\bar{q}, \bar{r}) = \frac{p^2}{2\mathring{p}\mathfrak{q}^{p+\mathring{p}/p}} \langle r, r \rangle + \frac{Cp^2}{\mathring{p}} \mathfrak{q}^{2p-\mathring{p}/p} f(q) + \frac{p}{\mathring{p}} \mathfrak{r}\mathfrak{q}^{1-\mathring{p}/p}. \tag{3.64}$$

This adaptive HTVI is the most natural Hamiltonian analogue of the LTVI described earlier in this section, and its updates are given by

$$\begin{aligned}
\mathfrak{q}_{k+1} &= \mathfrak{q}_k + h \frac{p}{\mathring{p}} \mathfrak{q}_k^{1-\mathring{p}/p}, \\
r_{k+1} &= r_k - \frac{p^2}{\mathring{p}} Ch \mathfrak{q}_k^{2p-\mathring{p}/p} \nabla f(q_k), \\
q_{k+1} &= q_k + \frac{p^2}{\mathring{p}} h \mathfrak{q}_k^{-p-\mathring{p}/p} r_{k+1}.
\end{aligned} \tag{3.65}$$

When $\mathring{p} = p$, it reduces to the direct HTVI,

$$\begin{aligned}
\mathfrak{q}_{k+1} &= \mathfrak{q}_k + h, \\
r_{k+1} &= r_k - hCp \mathfrak{q}_k^{2p-1} \nabla f(q_k), \\
q_{k+1} &= q_k + hp \mathfrak{q}_k^{-p-1} r_{k+1}.
\end{aligned} \tag{3.66}$$

Numerical Results

Numerical experiments using the numerical methods presented in the previous section have been conducted to minimize the convex quartic function,

$$f(x) = \left[(x-1)^\intercal \Sigma (x-1)\right]^2, \tag{3.67}$$

where $x \in \mathbb{R}^d$ and $\Sigma_{ij} = 0.9^{|i-j|}$. This function achieves its minimum value 0 at $x^* = 1$.

As was observed in [Duruisseaux et al., 2021] and in Section 3.3.5 for the HTVI algorithm, the numerical experiments showed that a carefully tuned adaptive approach algorithm enjoyed a significantly better rate of convergence and required a much smaller number of steps to achieve convergence than the direct approach, as can be seen in Figure 3.6 for the LTVI methods. Although the adaptive approach requires a smaller fictive time-step h than the direct approach, the physical time-steps resulting from $t = \tau^{p/\mathring{p}}$ in the adaptive approach grow rapidly to values larger than the physical time-step of the direct approach. The results of Figure 3.6 also show that the adaptive and direct LTVI methods become more and more efficient as p is increased, which was also the case for the HTVI algorithm in [Duruisseaux et al., 2021] and Section 3.3.5.

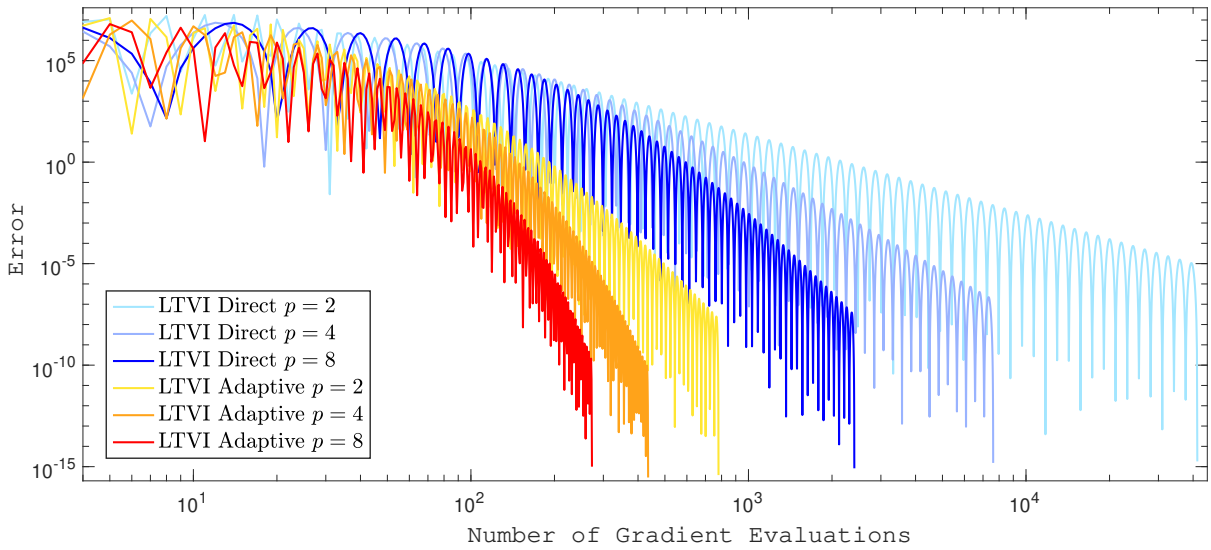


Figure 3.6: Comparison of the direct and adaptive approaches for the LTVI algorithm, when applied to the quartic function (3.67).

The Lagrangian and Hamiltonian Taylor variational integrators presented in this section perform empirically almost exactly in the same way for the same parameters and time-step, as can be seen for instance in Figure 3.7. As a result, the computational analysis carried in [Duruisseaux et al., 2021] and in Section 3.3.5 for the HTVI algorithm extends to the LTVI algorithm. In particular, this means that the LTVI algorithm is much more efficient than non-adaptive non-symplectic and adaptive non-symplectic integrators for the Bregman dynamics. It also means that the adaptive LTVI method can be a competitive first-order explicit algorithm since it can outperform certain popular optimization algorithms such as Nesterov's Accelerated Gradient [Nesterov, 1983], Trust Region Steepest Descent, ADAM [Kingma and Ba, 2014], AdaGrad [Duchi et al., 2011], and RMSprop [Tieleman and Hinton, 2012], for certain choices of objective functions.

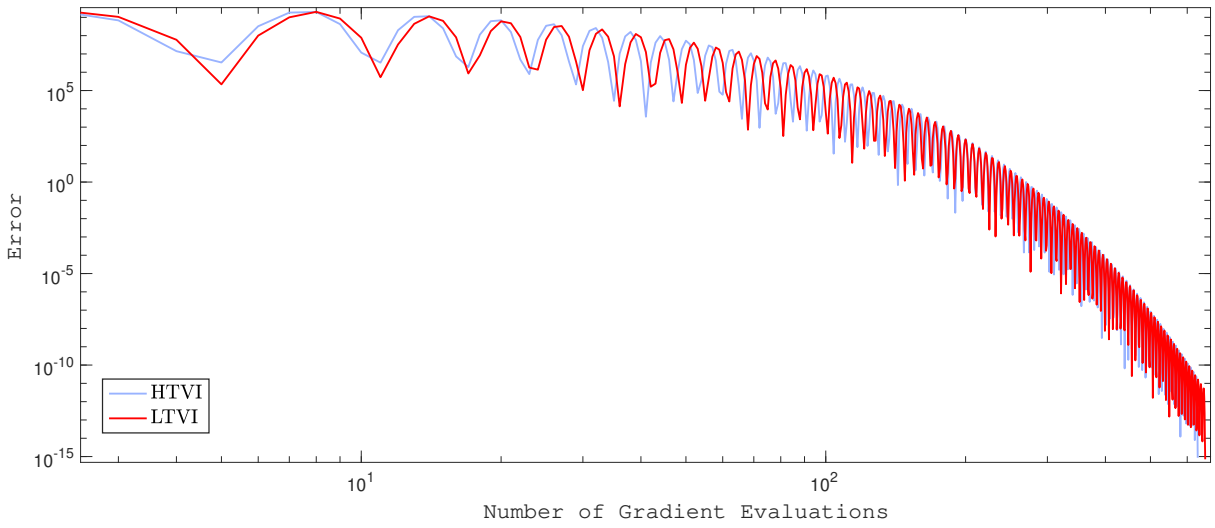


Figure 3.7: Comparison of the HTVI and LTVI algorithms with the same parameters.

Conclusion. Overall, we observed empirically that our time-adaptive Lagrangian variational integrators performed almost exactly in the same way as the time-adaptive Hamiltonian variational integrators coming from the Poincaré framework, whenever they are used with the same parameters and time-step. While time-adaptivity arises more naturally on the Hamiltonian side, our time-adaptive Lagrangian approach will prove very useful in Chapter 4 when we consider accelerated optimization on more general spaces such as Riemannian manifolds and Lie groups and will not have to face the difficulties experienced on the Hamiltonian side.

➢ Chapter 3 contains original material from

① "Adaptive Hamiltonian Variational Integrators and Applications to Symplectic Accelerated Optimization" by V. Duruisseaux, J. Schmitt, and M. Leok. *SIAM Journal on Scientific Computing,* Vol.43, No.4, A2949-A2980, 2021

② "Time-adaptive Lagrangian Variational Integrators for Accelerated Optimization on Manifolds" by V. Duruisseaux and M. Leok. *Journal of Geometric Mechanics*, Vol.15, Issue 1, pages 224-255, 2023.

The dissertation author was the primary investigator and author of these papers.

4 Accelerated Optimization via Geometric Numerical Integrators on Riemannian Manifolds

4.1 Introduction and Motivation

4.1.1 Optimization on Riemannian Manifolds

Riemannian optimization provides powerful alternatives to more general constrained optimization methods. We refer the reader to [Absil et al., 2008; Boumal, 2020] for a thorough introduction to optimization on manifolds. By exploiting the symmetries and special structure of the cost function, one can sometimes reformulate an ill-defined optimization problem on $\mathbb{R}^n$ into a nicer optimization problem on Riemannian manifolds, which then allows the design of optimization algorithms with better numerical properties.

A simple example is the Rayleigh quotient problem: eigenvectors corresponding to the largest eigenvalue of a symmetric $n \times n$ matrix A can be obtained as the solutions of the maximization problem on $\mathbb{R}^n$ of the Rayleigh quotient $\frac{v^\top A v}{v^\top v}$. These maximizers are however not isolated and as a consequence, important convergence results for optimization methods do not apply and many standard optimization algorithms fail. Restricting the Rayleigh quotient to the unit sphere $\mathbb{S}^{n-1}$ guarantees that the minimizers are isolated, and allows to design Riemannian optimization algorithms on $\mathbb{S}^{n-1}$ with convergence guarantees.

It is thus natural to ask whether the accelerated optimization framework presented in Chapter 3 can be extended to the Riemannian manifold setting.

4.1.2 Optimization Problems on Riemannian Manifolds

We will consider the following optimization problems:

Problem 4.1 (**Distance Minimization on** $\mathbb{H}^2$)**.** *Consider the problem presented in [Alimisis et al., 2020b] of minimizing the (strongly convex) distance function* $f(x) = \frac{1}{2}d(x,q)^2$ *for a given point* q*, on a subset of chosen finite diameter of the hyperbolic plane* $\mathbb{H}^2$*, which is a manifold with constant negative curvature* $K = -1$*.*

Problem 4.2 (**Rayleigh Quotient Optimization on** $\mathbb{S}^{n-1}$)**.** *Eigenvectors corresponding to the largest eigenvalue of a symmetric* $n \times n$ *matrix* A *maximize the Rayleigh quotient* $\frac{v^\top Av}{v^\top v}$ *over* $\mathbb{R}^n$*. Thus, a unit eigenvector* v^* *corresponding to the largest eigenvalue of the matrix* A *is a minimizer of the function* $f(v) = -v^\top Av$*, over the unit sphere* $\mathcal{Q} = \mathbb{S}^{n-1}$ *(see Example 1.1). Solving the Rayleigh quotient optimization problem efficiently is challenging when the given symmetric matrix* A *is ill-conditioned and high-dimensional. Note that an efficient algorithm that solves the above minimization problem can also be used to find eigenvectors corresponding to the smallest eigenvalue of* A *by using the fact that the eigenvalues of* A *are the negative of the eigenvalues of* $-A$*.*

Problem 4.3 (**Generalized Eigenvector Problem on** $\mathrm{St}(m,n)$)**.** *A generalized eigenvector problem consists of finding the* m *smallest eigenvalues of a* $n \times n$ *symmetric matrix* A *and corresponding eigenvectors. This problem can be formulated as a Riemannian optimization problem on the Stiefel manifold* $\mathrm{St}(m,n)$ *(see Example 1.2) via the Brockett cost function*

$$f : \mathrm{St}(m,n) \to \mathbb{R}, \qquad X \mapsto f(X) = \mathrm{Trace}(X^\top AXN), \tag{4.1}$$

where $N = diag(\mu_1, \ldots, \mu_m)$ *for arbitrary* $0 \le \mu_1 \le \ldots \le \mu_m$*. The columns of a global minimizer of* f *are eigenvectors corresponding to the* m *smallest eigenvalues of* A *(see [Absil et al., 2008]). If we define* $\bar{f} : \mathbb{R}^{n\times m} \to \mathbb{R}$ *via* $X \mapsto \bar{f}(X) = \mathrm{Trace}(X^\top AXN)$*, then* f *is the restriction of* $\bar{f}$ *to* $\mathrm{St}(m,n)$ *so*

$$\mathrm{grad} f(X) = P_X \mathrm{grad}\bar{f}(X), \quad \textit{where} \;\; \mathrm{grad}\bar{f}(X) = 2AXN. \tag{4.2}$$

Problem 4.4 (**Unbalanced Orthogonal Procrustes Problem on** $\mathrm{St}(m,n)$)**.** *The unbalanced orthogonal Procrustes problem consists of minimizing the function*

$$f : \mathrm{St}(m,n) \to \mathbb{R}, \qquad X \mapsto f(X) = \|AX - B\|_F^2, \tag{4.3}$$

on the Stiefel manifold $\mathrm{St}(m,n)$ *(see Example 1.2), for given matrices* $A \in \mathbb{R}^{l\times n}$ *and* $B \in \mathbb{R}^{l\times m}$ *with* $l \geq n$ *and* $l > m$*, where* $\|\cdot\|_F$ *is the Frobenius norm*

$$\|X\|_F^2 \;=\; \mathrm{Trace}(X^\top X) \;=\; \sum_{ij} X_{ij}^2. \tag{4.4}$$

If we define $\bar{f} : \mathbb{R}^{n\times m} \to \mathbb{R}$ *via* $X \mapsto \bar{f}(X) = \|AX - B\|_F^2$*, then* f *is the restriction of* $\bar{f}$ *to* $\mathrm{St}(m,n)$ *so*

$$\mathrm{grad} f(X) = P_X \mathrm{grad}\bar{f}(X), \qquad \textit{where} \;\; \mathrm{grad}\bar{f}(X) = 2A^\top(AX - B). \tag{4.5}$$

Note that the special case where $n = m$ *is the balanced orthogonal Procrustes problem. In this case,* $\mathrm{St}(m,n) = \mathrm{O}(n)$ *so* $\|AX\|_F^2 = \|A\|_F^2$ *and the problem of minimizing the function* $f(X) = \|AX - B\|_F^2$ *is replaced by the problem of maximizing* $\mathrm{Trace}(X^\top A^\top B)$ *over* $X \in \mathrm{O}(n)$*. A solution is then given by* $X^* = UV^\top$ *where* $B^\top A = U\Sigma V^\top$ *is the Singular Value Decomposition of* $B^\top A$ *with square orthogonal matrices* U *and* V*, and the solution is unique provided* $B^\top A$ *is nonsingular [Eldén and Park, 1999; Golub and Van Loan, 2013].*

Problem 4.5 (**Wahba's problem on** $\mathrm{SO}(3)$)**.** *Remember that can think of Lie groups as Riemannian manifolds (see Example 1.3), and this applies in particular to the Special Orthogonal group* $\mathrm{SO}(3)$ *(see Example 1.6). Wahba's problem [Wahba, 1965] on* $\mathrm{SO}(3)$ *concerns the least-squares estimation of attitude. More precisely, we wish to minimize the objective function,*

$$f(R) \;=\; \frac{1}{2}\|A - R\|_F^2 \;=\; \frac{1}{2}\left(\|A\|_F^2 + 3\right) - \mathrm{Trace}(A^\top R), \tag{4.6}$$

over $R \in \mathrm{SO}(3)$*, where* $\|\cdot\|_F$ *denotes the Frobenius norm. Its left-trivialized gradient is given by*

$$\nabla_L f(R) = \left(A^\top R - R^\top A\right)^\vee. \tag{4.7}$$

The optimal attitude is explicitly given by

$$R^* = U \mathrm{diag}\left[1, 1, \det(UV)\right] V^\top, \tag{4.8}$$

where $A = USV^\top$ *is the singular value decomposition of* A *with* $U, V \in \mathrm{O}(3)$ *and* S *diagonal.*

4.1.3 Outline of the Chapter

In recent years, there has been some effort to derive algorithms for accelerated optimization on Riemannian manifolds setting [Zhang and Sra, 2016; Liu et al., 2017; Zhang and Sra, 2018; Alimisis et al., 2020b,a; Ahn and Sra, 2020]. In particular, a second-order ODE was proposed in [Alimisis et al., 2020b] as the continuous-time limit of a Riemannian accelerated algorithm, and it was shown that the objective function $f(x(t))$ converges to its optimal value at a rate of $\mathcal{O}(1/t^2)$ along solutions of this ODE, thereby generalizing the Euclidean result obtained in [Su et al., 2016] to the Riemannian manifold setting.

We will show in Section 4.2 that in continuous time, the convergence rate of $f(x(t))$ to its optimal value can be accelerated to an arbitrary convergence rate on Riemannian manifolds by considering a family of time-dependent Riemannian Bregman Lagrangian and Hamiltonian systems, thereby generalizing the results of [Wibisono et al., 2016] to the Riemannian setting. This also provides a variational framework for Riemannian accelerated optimization, generalizing the vector space variational framework introduced in [Wibisono et al., 2016]. In particular, we will establish results for objective functions on Riemannian manifolds that are geodesically convex, weakly quasi-convex, and strongly convex. We will then illustrate the derived theoretical convergence rates in Section 4.3 by integrating the Bregman Euler–Lagrange equations using a simple numerical scheme to solve eigenvalue and distance minimization problems on Riemannian manifolds.

We will also show that the family of Bregman dynamics on Riemannian manifolds is closed under time rescaling in Section 4.4, and we will draw inspiration from the Adaptive approach from Section 3.3 to take advantage of this invariance property via a Poincaré transformation that will allow for the integration of higher-order dynamics with the computational efficiency of integrating lower-order dynamics. This lays the foundation for constructing similarly efficient optimization algorithms on Riemannian manifolds. The experience with the numerical discretization of variational accelerated optimization flows on vector spaces suggests that the combination of time-adaptivity and symplecticity is important for the efficient, robust, and stable discretization of these variational flows describing accelerated optimization. One expects that a geometric numerical integrator that is time-adaptive, symplectic, and Riemannian manifold preserving will yield a class of similarly promising optimization algorithms on manifolds.

Finally, this time-adaptive variational framework for accelerated optimization will be exploited in Sections 4.5 and 4.6. In the Hamiltonian setting, we will take advantage of the Whitney and Nash Embedding Theorems [Whitney, 1944a,b; Nash, 1956] to reduce Riemannian manifolds to submanifolds of Euclidean spaces, and exploit the structure of the embedding Euclidean spaces. More precisely, we will construct discrete constrained variational integrators by incorporating the manifold constraint directly into variational principles in Section 4.5.2, and design projection-based variational integrators in Section 4.5.3. In the Lagrangian setting, we will design integrators which evolve intrinsically on the Riemannian manifold in Section 4.6 by leveraging the framework for time-adaptive Lagrangian integrators from Section 2.7.2.

4.2 A Variational Formulation of Accelerated Optimization on Riemannian Manifolds

We now extend the normed vector space results of Section 3.2 to the Riemannian manifold setting. We will make the following assumptions on the function $f : \mathcal{Q} \to \mathbb{R}$ to be minimized and on the ambient Riemannian manifold $\mathcal{Q}$, which are standard assumptions in Riemannian optimization [Zhang and Sra, 2016, 2018; Alimisis et al., 2020b,a]:

Assumption 1. *Solutions of the differential equations derived in this chapter remain inside a geodesically uniquely convex subset A of a complete Riemannian manifold $\mathcal{Q}$ (that is, any two points in $\mathcal{Q}$ can be connected by a geodesic), such that* $\operatorname{diam}(A)$ *is bounded above by some constant D, that the sectional curvature is bounded from below by $K_{\min}$ on A, and that* Exp_q *is well-defined for any $q \in A$, and its inverse* Log_q *is well-defined and differentiable on A for any $q \in A$. Furthermore, f is bounded below, geodesically L-smooth and all its minima are inside A.*

Given a Riemannian manifold $\mathcal{Q}$ with sectional curvature bounded below by $K_{\min}$, and an upper bound D for the diameter of the considered domain, we define

$$\zeta = \begin{cases} \sqrt{-K_{\min}} D \coth\left(\sqrt{-K_{\min}} D\right) & \text{if } K_{\min} < 0 \\ 1 & \text{if } K_{\min} \geq 0 \end{cases}. \tag{4.9}$$

4.2.1 Convex and Weakly Quasi-Convex Cases

Suppose that $f : \mathcal{Q} \to \mathbb{R}$ is a given geodesically λ-weakly quasi-convex function, and that Assumption 1 holds true. Since a geodesically convex function is λ-weakly quasi-convex with $\lambda = 1$, the following treatment also applies to the case where f is geodesically convex. We define a family of **Bregman Lagrangians** $\mathscr{L}_{\alpha,\beta,\gamma} : T\mathcal{Q} \times \mathbb{R} \to \mathbb{R}$ parametrized by smooth functions of time α, β, γ by

$$\boxed{\mathscr{L}_{\alpha,\beta,\gamma}(X, V, t) \;=\; \frac{1}{2} e^{\lambda^{-1}\zeta\gamma_t - \alpha_t} \langle V, V \rangle - e^{\alpha_t + \beta_t + \lambda^{-1}\zeta\gamma_t} f(X),} \tag{4.10}$$

and the corresponding **Bregman Hamiltonians** $\mathscr{H}_{\alpha,\beta,\gamma} : T^*\mathcal{Q} \times \mathbb{R} \to \mathbb{R}$ are given by

$$\boxed{\mathscr{H}_{\alpha,\beta,\gamma}(X, R, t) \;=\; \frac{1}{2} e^{\alpha_t - \lambda^{-1}\zeta\gamma_t} \langle\!\langle R, R \rangle\!\rangle + e^{\alpha_t + \beta_t + \lambda^{-1}\zeta\gamma_t} f(X),} \tag{4.11}$$

where $X \in \mathcal{Q}$ denotes position on the manifold $\mathcal{Q}$, V is the velocity vector field, R is the momentum covector field, t is the time variable, and ζ is given by equation (4.9). This family of functions is a generalization of the Bregman Lagrangians and Hamiltonians introduced in [Wibisono et al., 2016] for the convex continuously differentiable function $h(x) = \frac{1}{2}\langle x, x \rangle$. Throughout this dissertation, we will assume that the parameter functions α, β, γ satisfy the ideal scaling conditions (3.7).

Theorem 4.1 ([Duruisseaux and Leok, 2022d])**.** *The Bregman Euler–Lagrange equation corresponding to the Bregman Lagrangian $\mathscr{L}_{\alpha,\beta,\gamma}$ is given by*

$$\boxed{\nabla_{\dot{X}} \dot{X} + \left(\lambda^{-1}\zeta e^{\alpha_t} - \dot{\alpha}_t\right) \dot{X} + e^{2\alpha_t + \beta_t} \mathrm{gradf}(X) \;=\; 0.} \tag{4.12}$$

Proof. See Appendix A.4.1.

Theorem 4.2 ([Duruisseaux and Leok, 2022d])**.** *Suppose that $f : \mathcal{Q} \to \mathbb{R}$ is a geodesically λ-weakly quasi-convex function, and that Assumption 1 is satisfied. Then, any solution $X(t)$ to the Bregman Euler–Lagrange equation* (4.12) *converges to a minimizer x^* of f with rate*

$$\boxed{f(X(t)) - f(x^*) \;\leq\; \frac{2\lambda^2 e^{\beta_0} \left(f(x_0) - f(x^*)\right) + \zeta \|\mathrm{Log}_{x_0}(x^*)\|^2}{2\lambda^2 e^{\beta_t}} = \mathcal{O}(e^{-\beta_t}).} \tag{4.13}$$

Proof. See Appendix A.5.

A $p > 0$ parametrized subfamily of Bregman Lagrangians and Hamiltonians, that is of particular practical interest, is given by the choice of parameter functions

$$\boxed{\alpha_t = \log p - \log t, \qquad \beta_t = p\log t + \log C, \qquad \gamma_t = p\log t,} \tag{4.14}$$

where $C > 0$ is a constant. This yields the ***p*-Bregman Lagrangians and Hamiltonians** given by

$$\boxed{\mathcal{L}_p(X, V, t) \;=\; \frac{t^{\lambda^{-1}\zeta p+1}}{2p}\langle V, V\rangle - Cpt^{(\lambda^{-1}\zeta+1)p-1} f(X),} \tag{4.15}$$

$$\boxed{\mathcal{H}_p(X, R, t) \;=\; \frac{p}{2t^{\lambda^{-1}\zeta p+1}}\langle\!\langle R, R\rangle\!\rangle + Cpt^{(\lambda^{-1}\zeta+1)p-1} f(X),} \tag{4.16}$$

and the corresponding p-Bregman Euler–Lagrange equations are given by

$$\boxed{\nabla_{\dot X}\dot X + \frac{\zeta p + \lambda}{\lambda t}\dot X + Cp^2 t^{p-2}\mathrm{gradf}(X) \;=\; 0.} \tag{4.17}$$

Theorem 4.3 ([Duruisseaux and Leok, 2022d])**.** *Suppose that $f : \mathcal{Q} \to \mathbb{R}$ is a geodesically weakly quasi-convex function, and that Assumption 1 is satisfied. Then, the p-Bregman Euler–Lagrange equation* (4.17) *has a solution, and any solution $X(t)$ converges to a minimizer x^* of f with rate*

$$\boxed{f(X(t)) - f(x^*) \;\leq\; \mathcal{O}(1/t^p)}. \tag{4.18}$$

Proof. See Appendix A.6.1 for the existence of a solution to the p-Bregman Euler–Lagrange equations. The $\mathcal{O}(1/t^p)$ convergence rate follows directly from Theorem 4.2.

Note that this theorem reduces to Theorem 5 from [Alimisis et al., 2020b] when $p = 2$ and $C = 1/4$.

Remark 4.1. *To construct this variational framework for accelerated optimization, we first constructed candidate p-equations with the desired $\mathcal{O}(1/t^p)$ convergence rates, and then designed Lagrangians whose p-Bregman Euler–Lagrange equations matched the candidate p-equations, by inspection. We then used a similar approach to extend these results to the general α, β, γ case presented here.*

Remark 4.2. *In our generalization of the Bregman Lagrangian and Hamiltonian to Riemannian manifolds, we have specialized to the case where* $h(x) = \frac{1}{2}\|x\|^2$*, because its Hessian* $\nabla^2 h(x)$ *is the identity matrix, which significantly simplifies the Euler–Lagrange equations and the analysis. In addition, it avoids the complication of making intrinsic sense of terms like* $X + e^{-\alpha}V$ *in the vector space Bregman Lagrangians and Hamiltonians, which requires the use of Riemannian geodesics and exponentials since* $X \in \mathcal{Q}$ *while* $V \in T_X\mathcal{Q}$*.*

4.2.2 Strongly Convex Case

Suppose that $f : \mathcal{Q} \to \mathbb{R}$ is a geodesically μ-strongly convex function on $\mathcal{Q}$, and that Assumption 1 is satisfied. With ζ given by equation (4.9), let

$$\eta = \left(\frac{1}{\sqrt{\zeta}} + \sqrt{\zeta}\right)\sqrt{\mu}. \tag{4.19}$$

We define the corresponding Lagrangian $\mathcal{L}^{SC} : T\mathcal{Q} \times \mathbb{R} \to \mathbb{R}$ by

$$\boxed{\mathcal{L}^{SC}(X, V, t) = \frac{e^{\eta t}}{2}\langle V, V\rangle - e^{\eta t} f(X),} \tag{4.20}$$

and the corresponding Hamiltonian $\mathcal{H}^{SC} : T^*\mathcal{Q} \times \mathbb{R} \to \mathbb{R}$ is given by

$$\boxed{\mathcal{H}^{SC}(X, R, t) = \frac{e^{-\eta t}}{2}\langle\langle R, R\rangle\rangle + e^{\eta t} f(X).} \tag{4.21}$$

Theorem 4.4 ([Duruisseaux and Leok, 2022d])**.** *The Bregman Euler–Lagrange equation corresponding to the Lagrangian* $\mathcal{L}^{SC}$ *is given by*

$$\boxed{\nabla_{\dot{X}}\dot{X} + \eta\dot{X} + \operatorname{gradf}(X) = 0.} \tag{4.22}$$

Proof. The derivation of the Euler–Lagrange equation is presented in Appendix A.4.2.

Theorem 4.5 ([Alimisis et al., 2020b; Duruisseaux and Leok, 2022d])**.** *Suppose* $f : \mathcal{Q} \to \mathbb{R}$ *is a geodesically* μ*-strongly convex function, and suppose that Assumption 1 is satisfied. Then, the Euler–Lagrange equation* (4.22) *has a solution, and any solution* $X(t)$ *converges to a minimizer* x^* *of* f *with rate*

$$\boxed{f(X(t)) - f(x^*) \;\leq\; \frac{\mu\|\operatorname{Log}_{x_0}(x^*)\|^2 + 2\left(f(x_0) - f(x^*)\right)}{2e^{\sqrt{\frac{\mu}{\zeta}}t}}.} \tag{4.23}$$

Proof. See Appendix A.6.2 for the existence of a solution to the Bregman Euler–Lagrange equation (4.22), and Theorem 7 from [Alimisis et al., 2020b] for the convergence rate.

4.3 Numerical Experiments

The p-Bregman Euler–Lagrange equation (4.17) can be rewritten as the first-order system of differential equations

$$\dot{X} = V, \qquad \nabla_V V = -\frac{\zeta p + \lambda}{\lambda t} V - Cp^2 t^{p-2} \mathrm{gradf}(X), \tag{4.24}$$

for the geodesically λ-weakly quasi-convex case, and the Euler–Lagrange equation (4.22) corresponding to the Lagrangian $\mathscr{L}^{SC}$ can be rewritten as the first-order system

$$\dot{X} = V, \qquad \nabla_V V = -\left(\frac{1}{\sqrt{\zeta}} + \sqrt{\zeta}\right)\sqrt{\mu} V - \mathrm{gradf}(X), \tag{4.25}$$

for the μ-strongly convex case.

As in [Alimisis et al., 2020b], we can adapt a semi-implicit Euler scheme (explicit Euler update for the velocity V followed by an update for the position X based on the updated value of V) to the Riemannian setting to obtain Algorithm 2:

Algorithm 2: Semi-Implicit Euler Integration of the p-Bregman Euler–Lagrange Equations

Input: A function $f : \mathcal{Q} \to \mathbb{R}$. Constants $C, h, p > 0$. $X_0 \in \mathcal{Q}$. $V_0 \in T_{X_0}\mathcal{Q}$.

while *convergence criterion is not met* **do**
  if *f is μ-geodesically strongly convex* **then**
    $b_k \leftarrow 1 - h\left(\frac{1}{\sqrt{\zeta}} + \sqrt{\zeta}\right)\sqrt{\mu}, \quad c_k \leftarrow 1$
  else if *f is λ-weakly quasi-convex* **then**
    $b_k \leftarrow 1 - \frac{\zeta p + \lambda}{\lambda k}, \quad c_k \leftarrow Cp^2(kh)^{p-2}$
  Version I: $a_k \leftarrow b_k V_k - h c_k \mathrm{gradf}(X_k)$
  Version II: $a_k \leftarrow b_k V_k - h c_k \mathrm{gradf}\left(\mathrm{Exp}_{X_k}(h b_k V_k)\right)$
  $X_{k+1} \leftarrow \mathrm{Exp}_{X_k}(h a_k), \quad V_{k+1} \leftarrow \Gamma_{X_k}^{X_{k+1}} a_k$

Version I of Algorithm 2 corresponds to the usual update for the Semi-Implicit Euler scheme, while Version II is inspired by the reformulation of Nesterov's method from [Sutskever et al., 2013] which uses a corrected gradient $\nabla f(X_k + hb_k V_k)$ instead of the traditional gradient $\nabla f(X_k)$. Note that the Semi-Implicit Riemannian Nesterov's Accelerated Gradient algorithm (SIRNAG) presented in [Alimisis et al., 2020b] corresponds to the special case where $p = 2$ and $C = 1/4$.

Numerical experiments carried out in [Alimisis et al., 2020b] showed that SIRNAG (the convex $p = 2$ Algorithm 2) and the strongly convex Algorithm 2 were of comparable efficiency or more efficient than the standard Riemannian Gradient Descent Algorithm 4, depending on the properties of the objective function and on the geometry of the Riemannian manifold.

We have conducted further numerical experiments to investigate how the simple discretization of higher-order $p = 6$ Bregman dynamics compared to its $p = 2$ counterpart, and to see whether it matches the $\mathcal{O}(k^{-p})$ convergence rate. The numerical results obtained for the Distance Minimization Problem 4.1 and Rayleigh Minimization Problem 4.2 are illustrated in Figure 4.1, where all the algorithms were implemented with the same fixed time-step. We can see that the $p = 6$ algorithms outperform their $p = 2$ counterparts, and that the efficiency improvement is very important. Furthermore, both versions of the $p = 6$ Algorithm 2 exhibit a faster convergence rate than $\mathcal{O}(k^{-6})$. While Version I of Algorithm 2 exhibits polynomial rates of $\mathcal{O}(k^{-10.8})$ and $\mathcal{O}(k^{-9})$ on the objective functions considered, Version II of Algorithm 2 exhibits a much faster exponential rate of convergence on both examples.

Figure 4.2 displays the evolution of the rates of convergence of Version 1 of the convex Algorithm 2 as the value of the parameter p is increased from $p = 4$ to $p = 16$ for the Distance Minimization Problem 4.1 and Rayleigh Minimization Problem 4.2. We can clearly see an improvement in the convergence rates as the value of p increases, and for each value of p the algorithm achieves a faster rate of convergence than $\mathcal{O}(k^{-p})$.

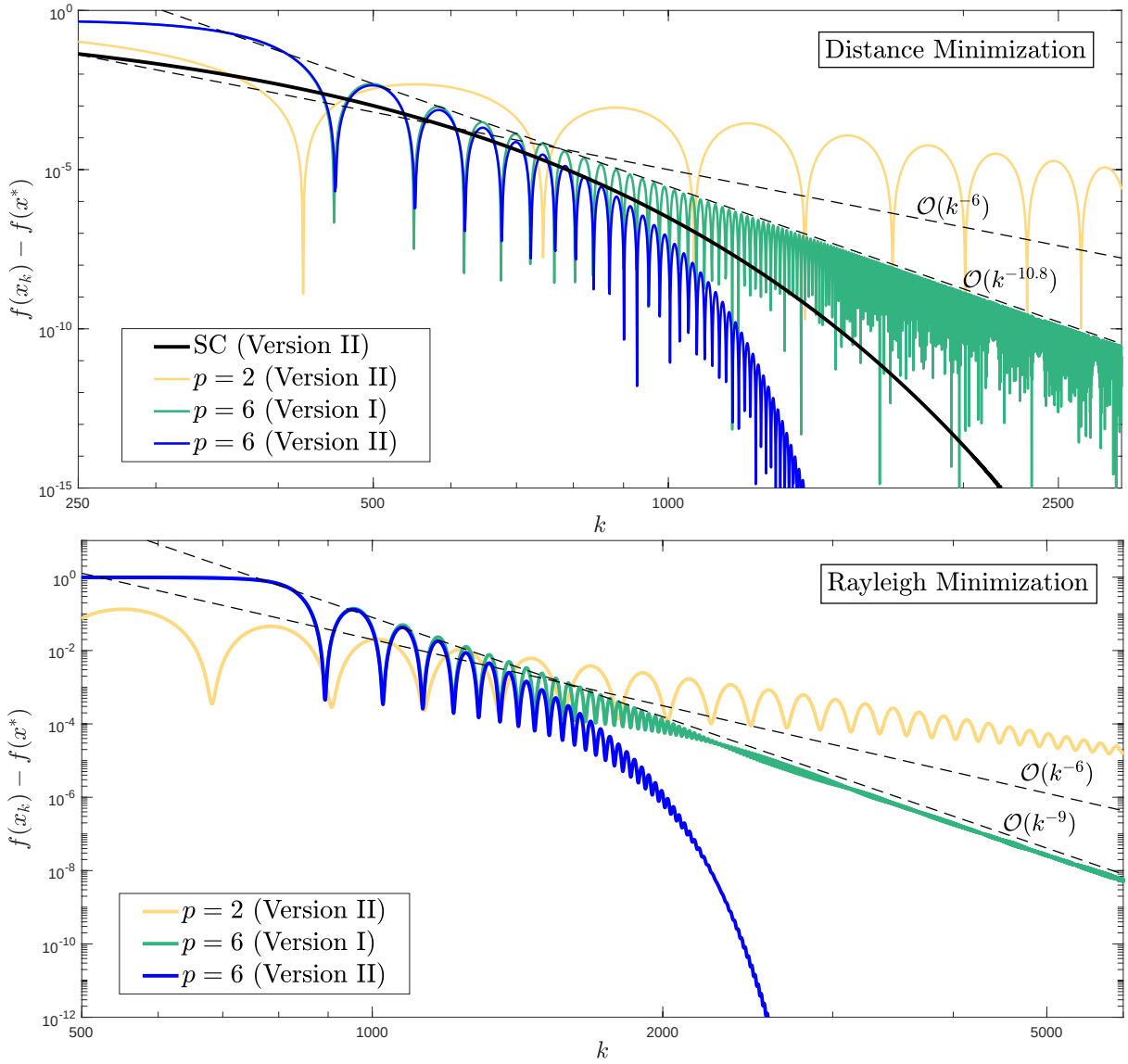


Figure 4.1: Comparison of the rates of convergence of the convex Algorithms 2 and the μ-strongly convex (SC) Algorithm 2 with different values of p and with the two versions of the update corresponding to the traditional and corrected gradients. Note that all the algorithms were implemented with the same time-step h.

Note however that an increase in the value of p in Algorithm 2, which corresponds to an increase in the order of the Bregman dynamics integrated, requires a decrease in the time-step h, in agreement with intuitive expectations. This time-step reduction requirement is especially important due to the polynomially growing $h(kh)^{p-2}$ coefficient

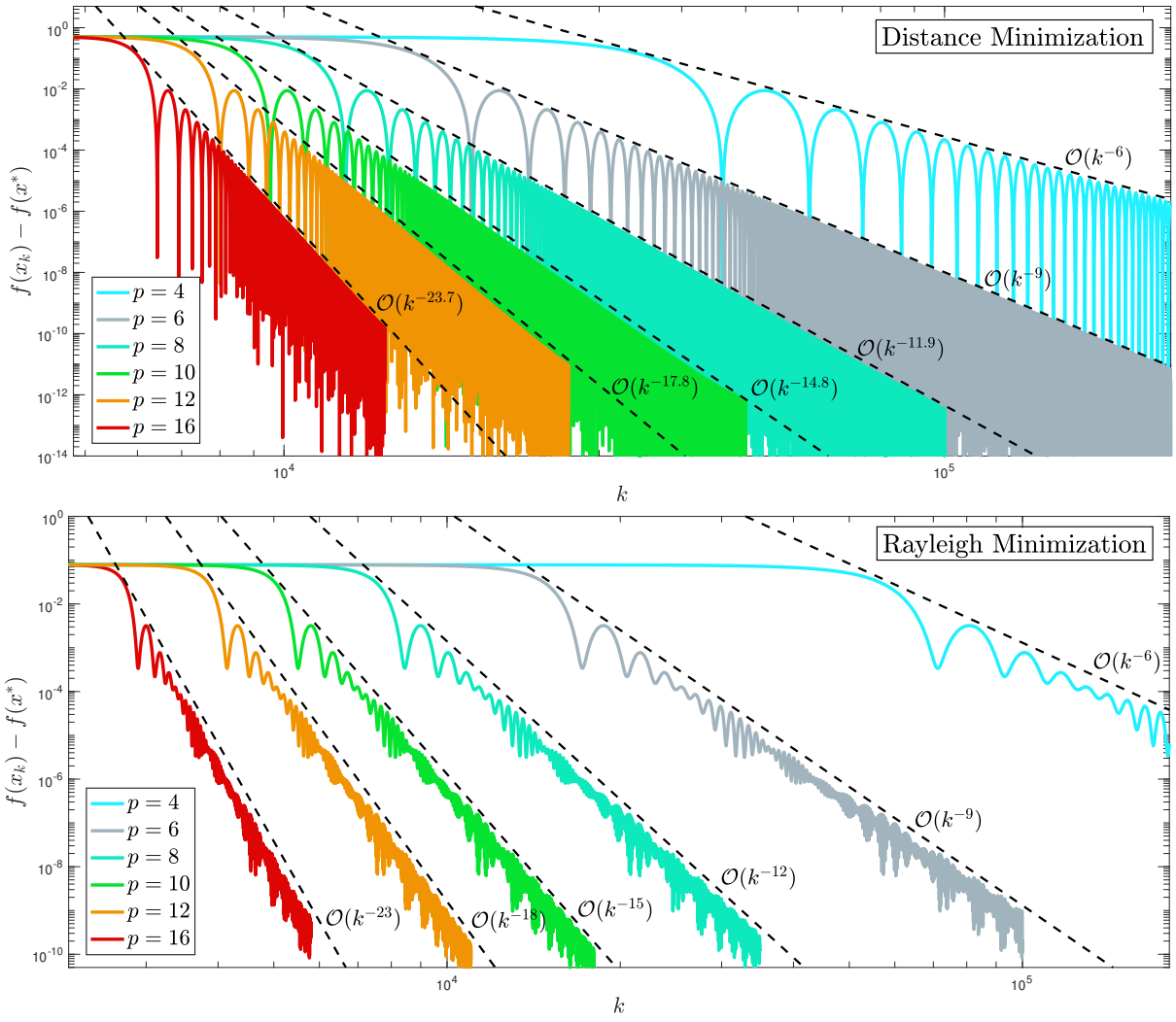


Figure 4.2: Evolution of the rates of convergence of Version 1 of the convex Algorithm 2 with different values of p. All the algorithms are implemented with the same time-step h.

multiplying the gradient of f in the updates of the algorithm. Such a decrease in the time-step does not really affect the convergence rate, but the transition between the initialization and convergence phases takes longer. As a consequence, by using larger time-steps, the algorithm corresponding to a smaller value of p might achieve a desired convergence criterion with fewer iterations than the algorithm corresponding to a larger value of p, despite having a slower convergence rate. Similar issues arise when discretizing the continuous Euler–Lagrange flow associated with accelerated optimization on vector spaces, and in that situation, it was observed that time-adaptive symplectic integrators

based on Hamiltonian variational integrators resulted in dramatically improved robustness and stability. As such, it will be natural to explore generalizations of time-adaptive symplectic integrators based on Hamiltonian variational integrators applied to Poincaré transformed Hamiltonians, that respect the Riemannian manifold structure in order to yield more robust and stable numerical discretizations of the flows presented in Section 4.2 in order to construct accelerated optimization algorithms on Riemannian manifolds. We will lay the foundation for such time-adaptive symplectic integrators in Section 4.4.

Finally, Figure 4.3 shows that the semi-implicit discretization of the Bregman Euler–Lagrange equation empirically converges to the true continuous solution, as the time-step h goes to 0. Note however that although all the discretizations follow the ODE trajectory closely, smaller time-steps result in a larger number of iterations, especially to transition from the initialization plateau to the convergence phase (around time $t = 4$ in the example presented in Figure 4.3).

A theoretical shadowing result bounding the error between the discrete-time RGD and its continuous-time limiting ODE was obtained in [Alimisis et al., 2020b] thanks to the uniform contraction property of the dynamical system associated with RGD. It would be desirable to obtain similar shadowing results for discretizations of the class of ODEs considered in this section, perhaps drawing inspiration from [Zhang et al., 2018]. However, such a result might be very difficult to obtain because momentum methods lack contraction, are nondescending, and are highly oscillatory [Orvieto and Lucchi, 2019; Alimisis et al., 2020b].

While it is hoped that the continuous analysis presented earlier in Section 4.2 will eventually guide the convergence analysis of discrete-time algorithms, this does not appear to be a straightforward exercise, as one would first need to reconcile the arbitrarily fast $\mathcal{O}(1/t^p)$ rate of convergence of the continuous-time trajectories with Nesterov's barrier theorem of $\mathcal{O}(1/k^2)$ for discrete-time algorithms. Even on normed vector spaces, obtaining theoretical guarantees was a challenging task, achieved in [Zhang et al., 2018] in the special case where $p > 2$ under additional assumptions on the objective function and on its

derivatives. Generalizing these results to the general family of α, β, γ Bregman Lagrangians on Riemannian manifolds would be much more challenging since the notions of derivatives become more complicated and all the usual vector space operations and objects have to be replaced by their Riemannian generalization which involve geodesics, parallel transport, Riemannian exponentials and Riemannian logarithms.

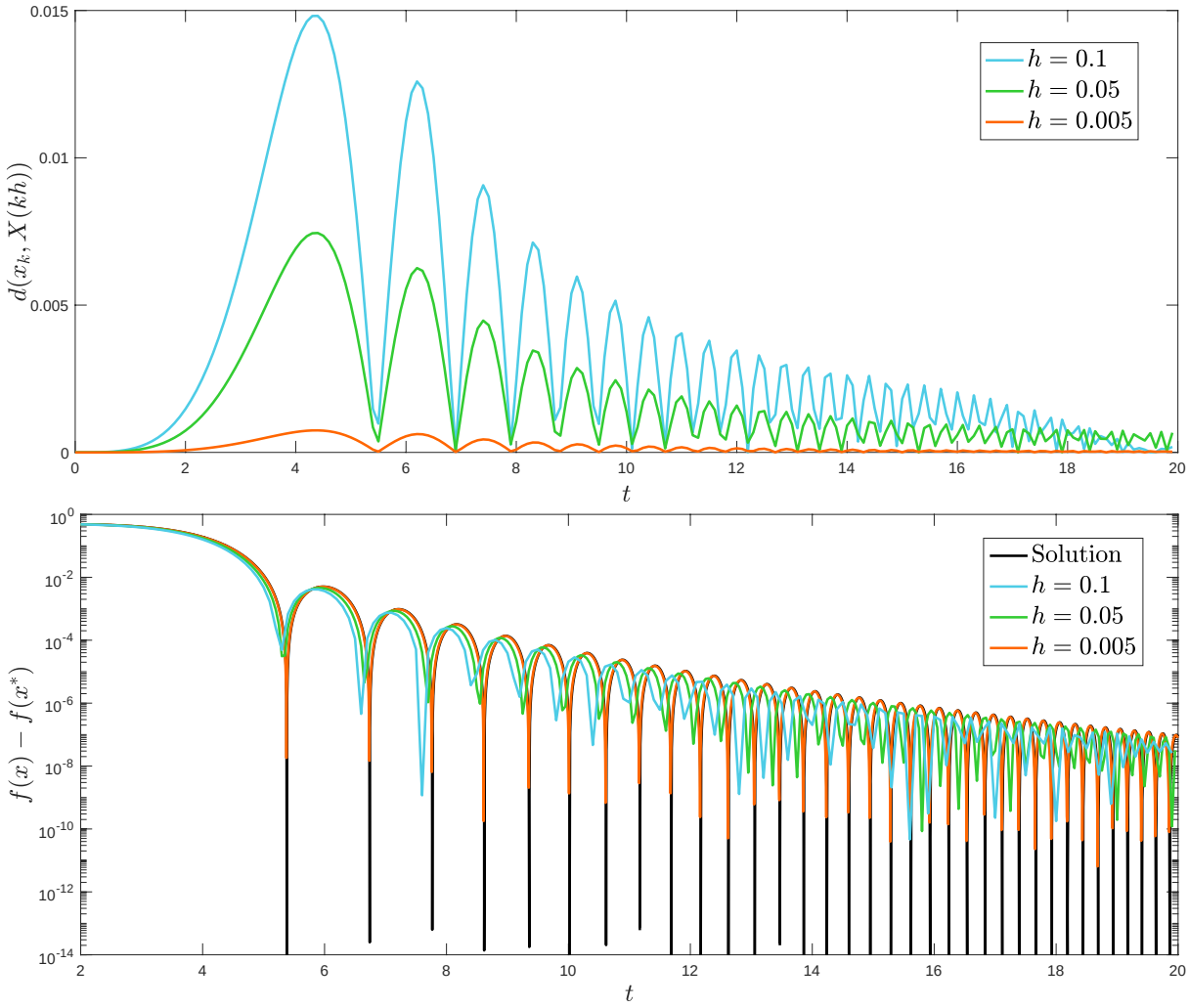


Figure 4.3: Discretization errors (top graph) and convergence rates (bottom graphs) of Version I of the $p = 5$ convex Algorithm 2 with different values of h for the distance minimization problem. The true solution of the differential equation was approximated by the same algorithm with a very small time-step $h = 10^{-5}$.

4.4 Time-Invariance Property of the Riemannian Bregman Family

Let $f : \mathcal{Q} \to \mathbb{R}$ be a given λ-weakly quasi-convex objective function, and suppose that Assumption 1 is satisfied. We now extend the time-invariance Theorem 3.1 from [Wibisono et al., 2016] to Riemannian manifolds.

Theorem 4.6 ([Duruisseaux and Leok, 2022d])**.** *Suppose that Assumption 1 is satisfied and that $X(t)$ satisfies the Riemannian Bregman Euler–Lagrange equation* (4.12) *corresponding to $\mathcal{L}_{\alpha,\beta,\gamma}$. Then the reparametrized curve $X(\tau(t))$ satisfies the Bregman Euler–Lagrange equation* (4.12) *corresponding to the modified Riemannian Bregman Lagrangian $\mathcal{L}_{\tilde{\alpha},\tilde{\beta},\tilde{\gamma}}$ where $\tilde{\alpha}_t = \alpha_{\tau(t)} + \log \dot{\tau}(t)$, $\tilde{\beta}_t = \beta_{\tau(t)}$, and $\tilde{\gamma}_t = \gamma_{\tau(t)}$. Furthermore α, β, γ satisfy the ideal scaling conditions* (3.7) *if and only if $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$ do.*

Proof. See Appendix A.7.

As a special case, we have the following theorem:

Theorem 4.7 ([Duruisseaux and Leok, 2022d])**.** *Suppose that $f : \mathcal{Q} \to \mathbb{R}$ is a geodesically λ-weakly quasi-convex function, and that Assumption 1 is satisfied. Suppose $X(t)$ satisfies the p-Bregman Euler–Lagrange equation* (4.17)*. Then, the reparametrized curve $X(t^{\mathring{p}/p})$ satisfies the $\mathring{p}$-Bregman Euler–Lagrange equation* (4.17)*.*

Thus, the entire subfamily of Bregman trajectories indexed by the parameter p can be obtained by speeding up or slowing down along the Bregman curve in spacetime corresponding to any specific value of p. Inspired by the computational efficiency of the time-adaptive approaches for optimization presented in Sections 3.3 and 3.4 and first introduced in [Duruisseaux et al., 2021; Duruisseaux and Leok, 2023a], it is natural to attempt to exploit the time-rescaling property of the Bregman dynamics together with a carefully chosen time transformation to transform the p-Bregman dynamics into an autonomous version of the $\mathring{p}$-Bregman dynamics. This would allow us to integrate the higher-order p-Bregman dynamics while benefiting from the computational efficiency of integrating the lower-order $\mathring{p}$-Bregman dynamics.

4.5 Riemannian Accelerated Optimization via Time-Adaptive Hamiltonian Integrators

4.5.1 Introduction

As in the normed space setting, we can exploit the time-rescaling property of the Bregman dynamics together with a carefully chosen Poincaré transformation to transform the Riemannian p-Bregman Hamiltonian into an autonomous version of the $\mathring{p}$-Bregman Hamiltonian in extended phase-space, where $\mathring{p} < p$. Explicitly, the time transformation $\tau(t) = t^{\mathring{p}/p}$ is associated to the monitor function

$$\frac{dt}{d\tau} = g_{p\to\mathring{p}}(t) = \frac{p}{\mathring{p}} t^{1-\mathring{p}/p}, \tag{4.26}$$

and generates the Poincaré transformed Hamiltonian

$$\bar{\mathcal{H}}_{p\to\mathring{p}}(\bar{X}, \bar{R}) = g_{p\to\mathring{p}}(\mathfrak{X}) \left(\mathcal{H}_p \left(\bar{X}, R \right) + \mathfrak{R} \right), \tag{4.27}$$

in the extended space $\bar{\mathcal{Q}} = \mathcal{Q} \times \mathbb{R}$ where

$$\bar{X} = \begin{bmatrix} X \\ \mathfrak{X} \end{bmatrix} \qquad \text{and} \qquad \bar{R} = \begin{bmatrix} R \\ \mathfrak{R} \end{bmatrix}. \tag{4.28}$$

We will make the conventional choice $\mathfrak{X} = t$, with conjugate momentum $\mathfrak{R}$, and

$$\mathfrak{R}(0) = -\mathcal{H}_p(X(0), R(0), 0) = -H_0, \tag{4.29}$$

which is chosen so that $\bar{\mathcal{H}}_{p\to\mathring{p}}(\bar{X}, \bar{R}) = 0$ along all integral curves through $(\bar{X}(0), \bar{R}(0))$. The time t shall be referred to as the physical time, while τ will be referred to as the fictive time. The corresponding Hamiltonian equations of motion in the extended phase space are then given by

$$\dot{\bar{X}} = \frac{\partial \bar{\mathcal{H}}_{p\to\mathring{p}}}{\partial \bar{R}}, \qquad \dot{\bar{R}} = -\frac{\partial \bar{\mathcal{H}}_{p\to\mathring{p}}}{\partial \bar{X}}. \tag{4.30}$$

Now, suppose $(\bar{X}(\tau), \bar{R}(\tau))$ are solutions to these extended equations of motion, and let $(x(t), r(t))$ solve Hamilton's equations for the original Hamiltonian $\mathcal{H}_p$. Then

$$\bar{\mathcal{H}}_{p\to\mathring{p}}(\bar{X}(\tau), \bar{R}(\tau)) = \bar{\mathcal{H}}_{p\to\mathring{p}}(\bar{X}(0), \bar{R}(0)) = 0. \tag{4.31}$$

Thus, the components $(X(\tau), R(\tau))$ in the original phase space of $(\bar{X}(\tau), \bar{R}(\tau))$ satisfy

$$\mathcal{H}_p(X(\tau), R(\tau), \tau) = -\mathfrak{R}(\tau), \quad \mathcal{H}_p(X(0), R(0), 0) = -\mathfrak{R}(0) = \mathcal{H}_p(x(0), r(0), 0). \tag{4.32}$$

Therefore, $(X(\tau), R(\tau))$ and $(x(t), r(t))$ both satisfy Hamilton's equations for the original Hamiltonian $\mathcal{H}_p$ with the same initial values, so they must be the same.

As a consequence, instead of integrating the p-Bregman Hamiltonian system (4.16), we can focus on the Poincaré transformed Hamiltonian $\bar{\mathcal{H}}_{p\to\mathring{p}}$ in extended phase-space given by equation (4.27), with $\mathcal{H}_p$ and $g_{p\to\mathring{p}}$ given by equations (4.16) and (4.26), that is

$$\boxed{\bar{\mathcal{H}}_{p\to\mathring{p}}(\bar{X}, \bar{R}) \;=\; \frac{p^2}{2\mathring{p}\mathfrak{X}^{\lambda^{-1}\zeta p+\mathring{p}/p}}\langle\!\langle R, R\rangle\!\rangle + \frac{Cp^2}{\mathring{p}}\mathfrak{X}^{(\lambda^{-1}\zeta+1)p-\mathring{p}/p} f(X) + \frac{p}{\mathring{p}}\mathfrak{X}^{1-\mathring{p}/p}\mathfrak{R}.} \tag{4.33}$$

The resulting integrator has constant time-step in fictive time τ but variable time-step in physical time t. Similarly, we also have the Direct approach Riemannian p-Bregman Hamiltonian in the special case where $p = \mathring{p}$:

$$\boxed{\bar{\mathcal{H}}_p(\bar{X}, \bar{R}) \;=\; \frac{p}{2\mathfrak{X}^{\lambda^{-1}\zeta p+1}}\langle\!\langle R, R\rangle\!\rangle + Cp\mathfrak{X}^{(\lambda^{-1}\zeta+1)p-1} f(X) + \mathfrak{R}.} \tag{4.34}$$

Remark 4.3. *In the vector space setting, these Riemannian Bregman Hamiltonians reduce to the Adaptive and Direct approach Bregman Hamiltonians from Section 3.3 first derived in [Duruisseaux et al., 2021] for convex functions:*

$$\bar{H}_{p\to\mathring{p}}(\bar{q}, \bar{r}) = \frac{p^2}{2\mathring{p}\mathfrak{q}^{p+\mathring{p}/p}}\langle r, r\rangle + \frac{Cp^2}{\mathring{p}}\mathfrak{q}^{2p-\mathring{p}/p} f(q) + \frac{p}{\mathring{p}}\mathfrak{q}^{1-\mathring{p}/p}\mathfrak{r}. \tag{4.35}$$

$$\bar{H}_p(\bar{q}, \bar{r}) = \frac{p}{2\mathfrak{q}^{p+1}}\langle r, r\rangle + Cp\mathfrak{q}^{2p-1} f(q) + \mathfrak{r}, \tag{4.36}$$

Outline. We will integrate the Riemannian Poincaré transformed Hamiltonian in two different ways. The Whitney and Nash Embedding Theorems [Whitney, 1944a,b; Nash, 1956] imply that any Riemannian manifold can be embedded into some Euclidean space, and both strategies considered here will exploit the structure of the embedding Euclidean space. In Section 4.5.2, we will construct discrete constrained variational integrators by incorporating the manifold constraint directly into variational principles, and we will design projection-based variational integrators in Section 4.5.3.

4.5.2 Riemannian Accelerated Optimization via Discrete Constrained Hamiltonian Integrators

The Whitney Embedding Theorems [Whitney, 1944a,b] state that any smooth manifold of dimension $n \geq 2$ can be embedded in $\mathbb{R}^{2n}$ and immersed in $\mathbb{R}^{2n-1}$, and is, as a result, diffeomorphic to a submanifold of $\mathbb{R}^{2n}$. Furthermore, the Nash Embedding Theorems [Nash, 1956] state that any Riemannian manifold can be globally isometrically embedded into some Euclidean space. As a consequence of these embedding theorems, the study of Riemannian manifolds can in principle be reduced to the study of submanifolds of Euclidean spaces, which motivates the introduction of time-adaptive variational integrators on Riemannian manifolds which exploit the structure of the embedding Euclidean space.

We will first consider here the case of Riemannian manifolds embedded in some Euclidean space that can be characterized as the level set of a submersion. Our approach consists in integrating the Direct and Adaptive Riemannian Bregman Hamiltonian systems derived in [Duruisseaux and Leok, 2022d] and presented in Sections 4.2 and 4.4, which evolve on the Riemannian manifold $\mathcal{Q}$, via the discrete constrained variational Hamiltonian integrators constructed as in Section 2.6.2 to enforce the numerical solution to lie on the Riemannian manifold $\mathcal{Q}$.

We will use the Direct approach and Adaptive approach $r = 0$ Type II Hamiltonian Taylor variational integrators constructed in [Duruisseaux et al., 2021] and Section 3.3.4 based on the Direct and Adaptive discrete right Hamiltonians (respectively)

$$H_d^+(\bar{q}_0, \bar{r}_1; h) = r_1^\top q_0 + \mathfrak{r}_1 \mathfrak{q}_0 + h \frac{p}{2\mathfrak{q}_0^{p+1}} r_1^\top r_1 + hCp\mathfrak{q}_0^{2p-1} f(q_0) + h\mathfrak{r}_1, \tag{4.37}$$

$$H_d^+(\bar{q}_0, \bar{r}_1; h) = r_1^\top q_0 + \mathfrak{r}_1 \mathfrak{q}_0 + h \frac{p^2}{2\mathring{p}\mathfrak{q}_0^{p+\frac{\mathring{p}}{p}}} r_1^\top r_1 + hC\frac{p^2}{\mathring{p}}\mathfrak{q}_0^{2p-\frac{\mathring{p}}{p}} f(q_0) + h\frac{p}{\mathring{p}}\mathfrak{q}_0^{1-\frac{\mathring{p}}{p}}\mathfrak{r}_1. \tag{4.38}$$

The resulting HTVI algorithms are presented in Algorithm 3 (the Direct approach algorithm can be obtained by setting $\mathring{p} = p$):

Algorithm 3: Riemannian Hamiltonian Taylor Variational Integrators

Input: A function $f : \mathcal{Q} \to \mathbb{R}$, constants $C, h, p, \mathring{p} > 0$, $\mathfrak{q}_0, \mathfrak{r}_0 \in \mathbb{R}$, and $(q_0, r_0, \lambda_0) \in T^*_{q_0}\mathcal{Q} \times \Lambda$.

1 **while** convergence criterion is not met, **solve** the following system of equations:

$$\begin{cases} 0 = r_{k+1} - r_k + \dfrac{hCp^2}{\mathring{p}} \mathfrak{q}_k^{2p-\frac{\mathring{p}}{p}} \nabla f(q_k) + \lambda_k^\top \nabla C(q_k) \\ 0 = \mathfrak{r}_{k+1} - \mathfrak{r}_k + \dfrac{p\mathring{p} - 2p^3}{\mathring{p}} hC\mathfrak{q}_k^{2p-\frac{\mathring{p}}{p}-1} f(q_k) + h\dfrac{p^3 + p\mathring{p}}{2\mathring{p}\mathfrak{q}_k^{p+\frac{\mathring{p}}{p}+1}} r_{k+1}^\top r_{k+1} + \dfrac{\mathring{p} - p}{\mathring{p}\mathfrak{q}_k^{\frac{\mathring{p}}{p}}} h\mathfrak{r}_{k+1} \\ 0 = q_{k+1} - q_k - \dfrac{p^2}{\mathring{p}} h\mathfrak{q}_k^{-p-\frac{\mathring{p}}{p}} r_{k+1} \\ 0 = \mathfrak{q}_{k+1} - \mathfrak{q}_k - \dfrac{p}{\mathring{p}} h\mathfrak{q}_k^{1-\frac{\mathring{p}}{p}} \\ 0 = C(q_{k+1}) \end{cases}$$

We will compare these integrators to the Riemannian Euler–Lagrange discretization presented in Algorithm 2 and to the following Riemannian generalization of Gradient Descent which involves the Riemannian gradient and a retraction:

Algorithm 4: Riemannian Gradient Descent (RGD)

Input: A function $f : \mathcal{Q} \to \mathbb{R}$, a retraction $\mathcal{R}$ from $T\mathcal{Q}$ to $\mathcal{Q}$, $h > 0$, and $X_0 \in \mathcal{Q}$.

1 **while** *convergence criterion is not met* **do**

2 $\quad X_{k+1} = \mathcal{R}_{X_k}(-h \operatorname{grad} f(X_k))$

Numerical Results

Numerical experiments were conducted for the Rayleigh quotient Minimization Problem 4.2 on $\mathbb{S}^{n-1}$ (see Example 1.1), and for the Generalized Eigenvalue Problem 4.3 and Procrustes Problems 4.4 on the Stiefel manifold $\mathrm{St}(m, n)$ (see Example 1.2).

The results, presented in Figure 4.4, show how the Riemannian Hamiltonian Taylor variational integrators compare to the Euler–Lagrange discretizations from [Duruisseaux and Leok, 2022d] and the standard Riemannian gradient descent. Note that for certain instances of the Procrustes problem with certain initial values, all the algorithms converged to a local minimizer, and not the global minimizer, of the objective function.

We can observe from Figure 4.4 that for the same value of the time-step h, the Adaptive Hamiltonian variational integrator clearly outperforms its Direct counterpart, Riemannian gradient descent and the Euler–Lagrange discretizations in terms of number of iterations required. Furthermore, unlike the Euler–Lagrange discretizations (Algorithm 2) and the Riemannian gradient descent (Algorithm 4), the HTVI methods (Algorithm 3) do not require the use of retractions or parallel transports.

We note that the Rayleigh minimization results indicate that the Euler–Lagrange discretizations suffer from stability issues leading to a loss of convergence, as the polynomially growing unbounded coefficient $Cp^2(kh)^{p-2}$ is multiplied with $\operatorname{grad} f$, so for this product to be bounded, the gradient has to decay to zero, but due to finite numerical precision, the gradient remains bounded away from zero, thereby causing the product to grow without bound. This issue can be resolved by adding a suitable upper bound to the coefficient $Cp^2(kh)^{p-2}$ in the updates, as can be seen both for the Euler–Lagrange discretizations and Hamiltonian variational integrators for the problems on $\mathrm{St}(m,n)$.

However, the algorithms generated by these constrained Hamiltonian variational integrators are implicit, which can significantly increase the cost per iteration as the dimension of the problem becomes very large. Further, note that the implementation of the Hamiltonian variational integrators needs a very careful tuning of the various parameters at play, which may be challenging. It might therefore be beneficial to consider other options using the unconstrained explicit Hamiltonian Taylor variational integrator. A first possibility is to incorporate the constraints within the objective function as a penalty, although this might not constrain the solution trajectory to lie exactly on the manifold. Another option involves projections if they can be computed efficiently and accurately for the Riemannian manifold of interest. The latter option, originally explored in [Duruisseaux and Leok, 2022c], will be presented in Section 4.5.3.

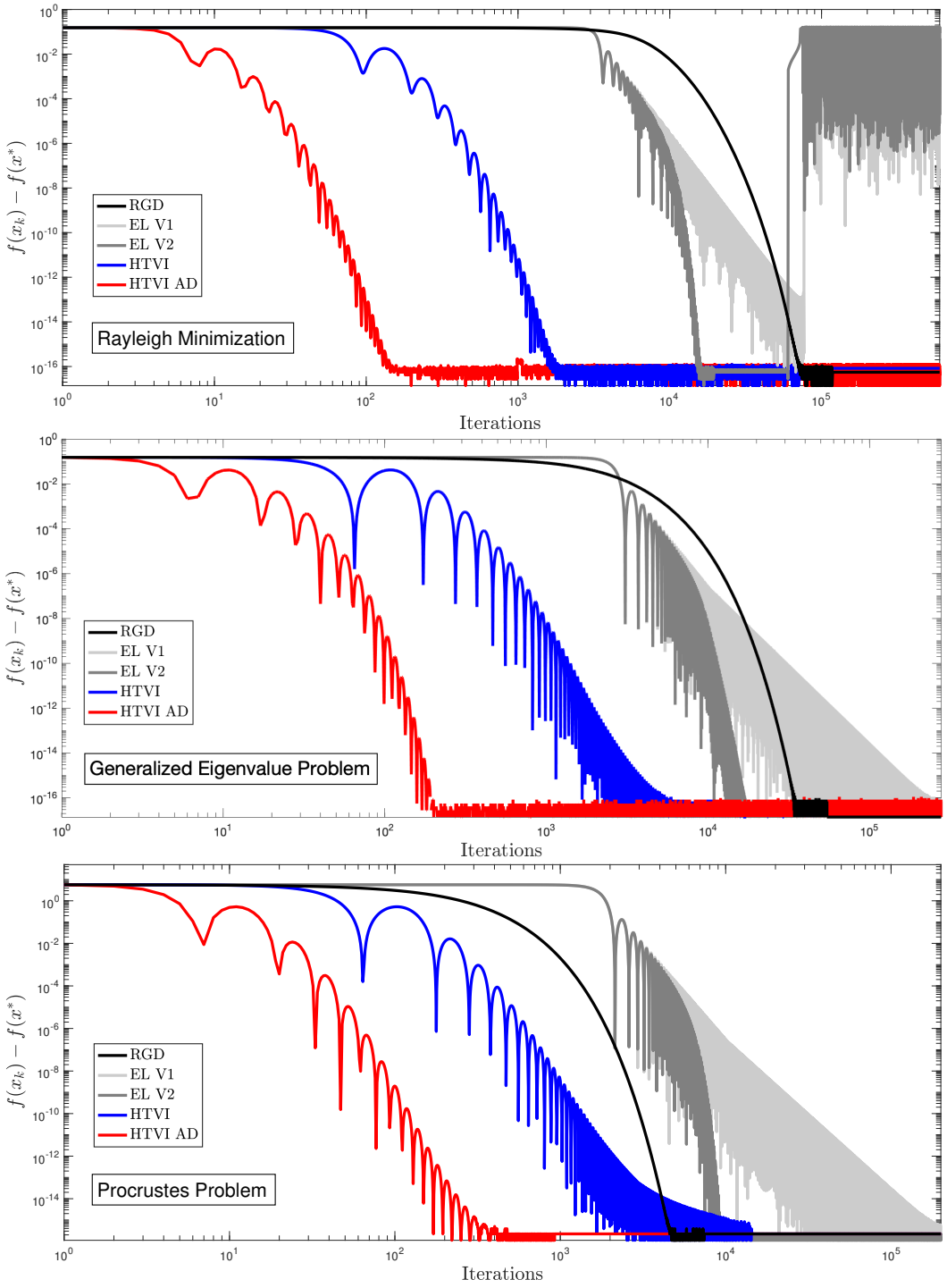


Figure 4.4: Comparison of the Direct and Adaptive (AD) Type II HTVIs with the Riemannian Gradient Descent (RGD) method and the Euler–Lagrange discretizations (EL V1 and EL V2) from [Duruisseaux and Leok, 2022d] with $p = 6$ and the same time-step $h = 0.001$, for the Rayleigh quotient minimization problem on the unit sphere $\mathbb{S}^{n-1}$, and for the generalized eigenvalue and Procrustes problems on the Stiefel manifold $\mathrm{St}(m, n)$.

4.5.3 Riemannian Accelerated Optimization via Projected Hamiltonian Integrators

We have just seen that the time-adaptive Hamiltonian approach to accelerated optimization relying on a Poincaré transformation in the Riemannian manifold setting introduced in [Duruisseaux and Leok, 2022d] can be exploited by incorporating holonomic constraints into variational integrators as discussed in Section 4.5.2 to constrain the numerical solution to the Riemannian manifold. Although these constrained integrators were carefully justified geometrically as coming from discrete variational principles, they were difficult to tune and implicit in nature, which can significantly increase their cost per iteration and can make them unpractical as the dimension of the problem becomes large.

We now present new algorithms based on explicit variational integrators in the embedding space where the manifold constraints are enforced via projections. More explicitly, we use fact that the Bregman Hamiltonian in the embedding space restricts to the Riemannian Bregman Hamiltonian on the Riemannian submanifold $\mathcal{Q}$, and the projection of the Bregman Hamiltonian vector field in the embedding space onto the tangent bundle $T\mathcal{Q}$ of the Riemannian submanifold recovers the Hamiltonian vector field of the Riemannian Bregman Hamiltonian. As such, we will numerically integrate the Bregman dynamics in the embedding space and use projections to force the numerical solution to lie on $\mathcal{Q}$. If projections onto the constraint manifold $\mathcal{Q}$ can be computed exactly or approximately very efficiently, we can simply project the updated position onto $\mathcal{Q}$ after every iteration. Furthermore, if projections onto tangent spaces $T_q\mathcal{Q}$ for any point $q \in \mathcal{Q}$ are also available at a low computational cost, it might sometimes be helpful to project the update vector onto $T_q\mathcal{Q}$.

Projections have been found for most Riemannian manifolds of practical interest (see [Absil et al., 2008; Boumal, 2020]). These typically involve standard matrix factorizations which can be efficiently computed using iterative methods, and if they are expensive to compute, there are usually ways to accelerate the computations via approximations.

We will now use projection-based versions of the Hamiltonian Taylor Variational Integrators (HTVIs) constructed in [Duruisseaux et al., 2021] and presented in 3.3.4. Given an objective function $f : \mathcal{Q} \to \mathbb{R}$, a projection operator $\mathcal{P}_{\mathcal{Q}}$ onto the manifold $\mathcal{Q}$, $(q_0, r_0) \in T^*_{q_0}\mathcal{Q}$, and constants $C, h, p, \mathring{p}, \mathfrak{q}_0 > 0$, the Direct and Adaptive HTVIs are obtained by iterating the updates given in Algorithm 5:

Algorithm 5: Direct and Adaptive HTVIs

1 **Direct HTVI**

$$\begin{aligned} r_{k+1} &= r_k - phC\mathfrak{q}_k^{2p-1}\mathrm{grad}f(q_k), \\ q_{k+1} &= \mathcal{P}_{\mathcal{Q}}\left(q_k + ph\mathfrak{q}_k^{-p-1}r_{k+1}\right), \\ \mathfrak{q}_{k+1} &= \mathfrak{q}_k + h. \end{aligned}$$

2 **Adaptive HTVI**

$$\begin{aligned} r_{k+1} &= r_k - \frac{p^2}{\mathring{p}}hC\mathfrak{q}_k^{2p-\mathring{p}/p}\mathrm{grad}f(q_k), \\ q_{k+1} &= \mathcal{P}_{\mathcal{Q}}\left(q_k + \frac{p^2}{\mathring{p}}h\mathfrak{q}_k^{-p-\mathring{p}/p}r_{k+1}\right), \\ \mathfrak{q}_{k+1} &= \mathfrak{q}_k + \frac{p}{\mathring{p}}h\mathfrak{q}_k^{1-\mathring{p}/p}. \end{aligned}$$

We will compare these integrators to the Riemannian Euler–Lagrange discretization presented in Algorithm 2 and to the Riemannian Gradient Descent Algorithm 4.

Comparison of the Adaptive and Direct approaches

Numerical experiments were conducted for the Rayleigh quotient minimization Problem 4.2 on the unit sphere $\mathbb{S}^{n-1}$ with the projection based method (see Example 1.1 for more information about $\mathbb{S}^{n-1}$). As was observed in [Duruisseaux et al., 2021] in the vector space setting, Figure 4.5 shows that the Adaptive approach can be significantly more efficient than the Direct approach, and that both methods enjoy faster convergence as the value of the parameter p is increased.

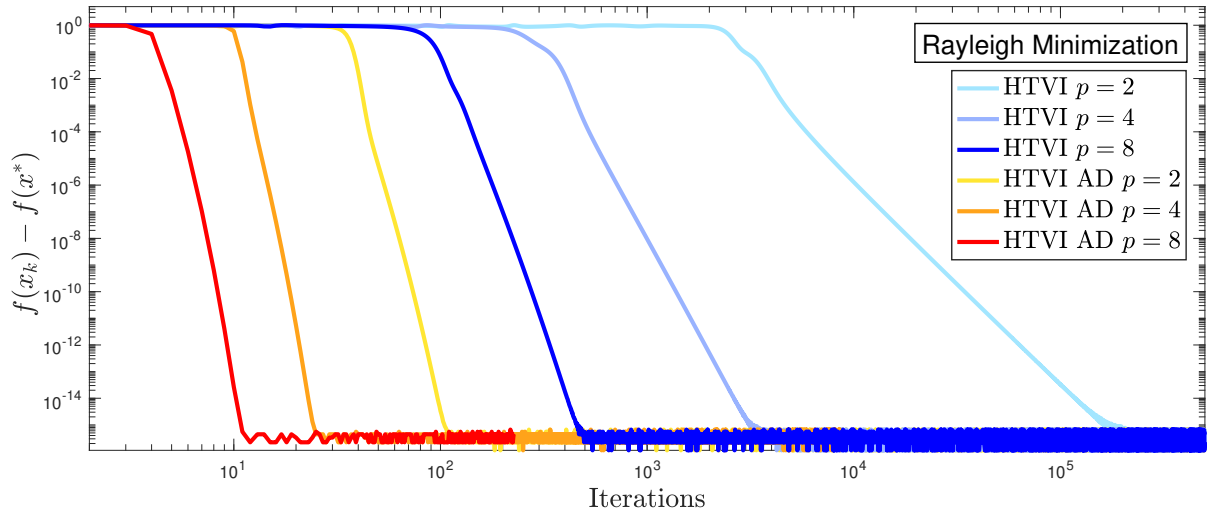


Figure 4.5: Comparison of the Direct and Adaptive (AD) projection-based HTVIs with different values of p and the same time-step h, for the Rayleigh minimization problem 4.2.

Rayleigh minimization problem on the unit sphere $\mathbb{S}^{n-1}$

Additional numerical experiments have been conducted for the Rayleigh quotient minimization problem 4.2 on $\mathbb{S}^{n-1}$, and the results are presented in Figure 4.6.

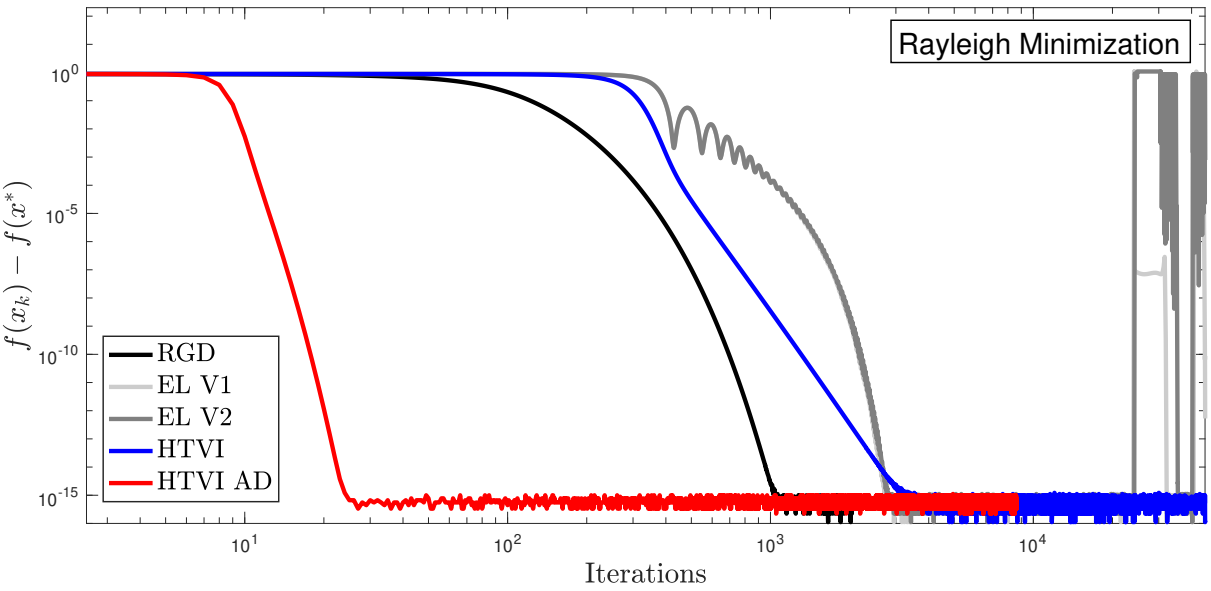


Figure 4.6: Comparison of the Direct and Adaptive (AD) projection-based HTVIs with the Riemannian Gradient Descent (RGD) method and the Euler–Lagrange discretizations (EL V1 and EL V2), with $p = 4$ and the same time-step h.

The Adaptive HTVI clearly outperforms the other algorithms for this problem. Note that the Euler–Lagrange discretizations suffer from stability issues leading to a loss of convergence (after $\approx 10^4$ iterations), due to the polynomially growing unbounded term $Cp^2(kh)^{p-2}$ paired with the $\mathrm{grad} f$ term to 0 which can only achieve a finite order of accuracy due to numerical roundoff error. This issue can be fixed by adding a suitable upper bound to the coefficient $Cp^2(kh)^{p-2}$ in the updates, or by stopping the iterating process once a desired convergence criterion is achieved.

Optimization Problems on the Stiefel manifold $\mathbf{St}(m,n)$

Numerical experiments were conducted for the generalized eigenvalue problem 4.3 and for the Procrustes problem 4.4 on $\mathrm{St}(m,n)$ to investigate how the projection based Hamiltonian Taylor variational integrators compare to the Euler–Lagrange discretizations from Section 4.3 and [Duruisseaux and Leok, 2022d] and the standard Riemannian gradient descent. We experimented both with the projection based on polar decomposition and the projection based on matrix QR orthogonalization presented in Example 1.2. The results are presented in Figures 4.7 and 4.8. Note that for certain instances of the Procrustes problem with certain initial values $X_0 \in \mathrm{St}(m,n)$, all the algorithms converged to a local minimizer which was not a global minimizer.

The projection based Adaptive Hamiltonian Taylor variational integrators clearly outperform their Direct approach counterparts, Riemannian gradient descent and both versions of the Euler–Lagrange discretization in terms of number of iterations required, when all the algorithms are implemented with the same time-step (see the two bottom plots in Figure 4.4). As can be seen from the top two plots in Figure 4.4, the Adaptive HTVIs are still the best performing algorithms, even when larger time-steps are taken for the other algorithms and in particular even when the Riemannian gradient descent algorithm has been tuned optimally. Note that both the Euler–Lagrange discretizations and Hamiltonian variational integrators suffered from the numerical roundoff issue described in the previous subsection, but this issue was resolved by adding a suitable upper bound to the ever-growing problematic coefficient in the updates.

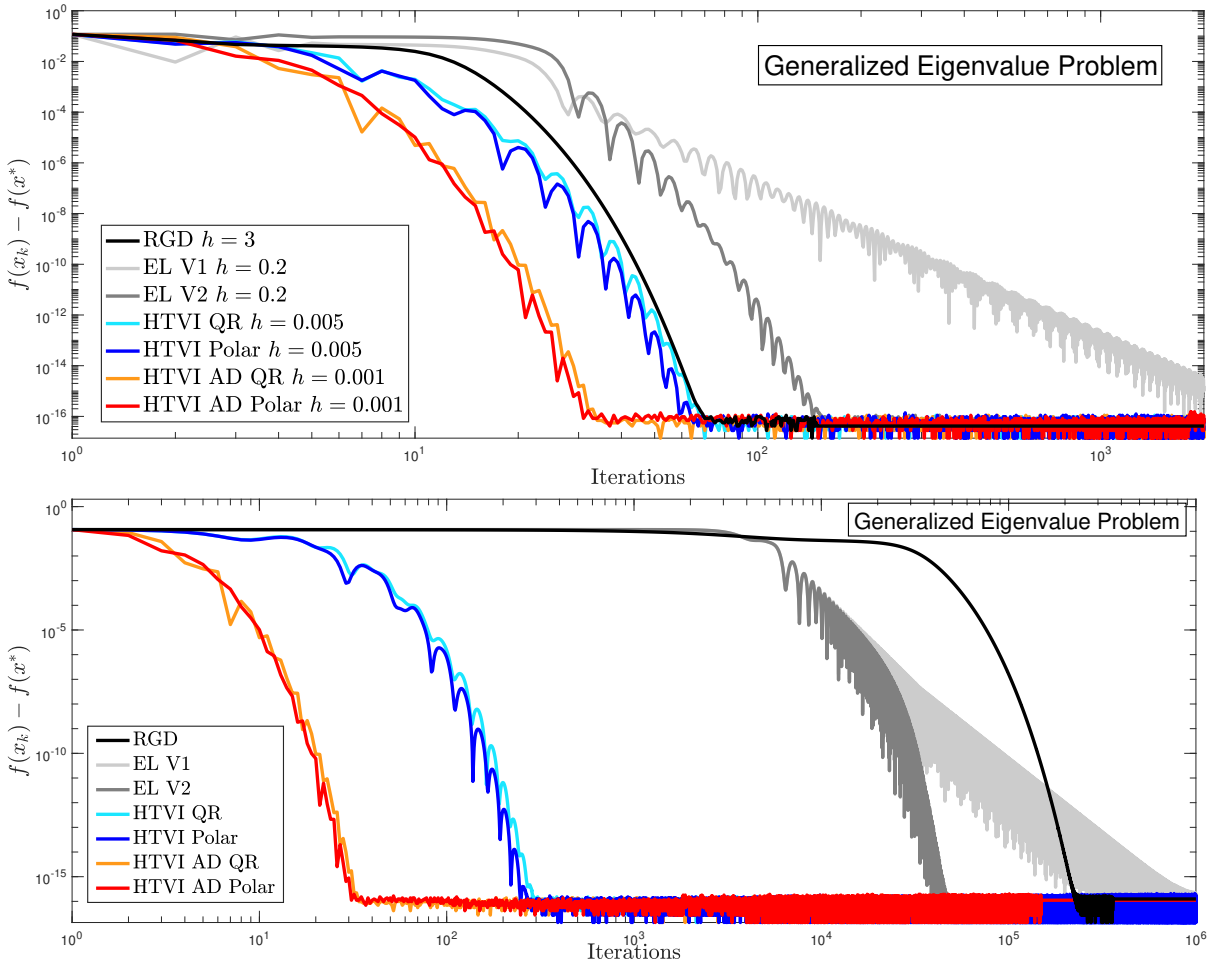


Figure 4.7: Direct and Adaptive (AD) HTVIs, Riemannian Gradient Descent (RGD), and Euler–Lagrange discretizations (EL V1 and EL V2) with different time-steps (top) and with the same time-step (bottom) for the generalized eigenvalue problem 4.3 on $\mathrm{St}(m, n)$.

Our numerical experiments do not suggest that there is a clear benefit in using the polar decomposition based projection over the matrix orthogonalization, or vice versa. Both projection strategies led to very efficient algorithms for Riemannian accelerated optimization with seemingly similar performance and stability properties. Computing the QR decomposition of a $n \times m$ matrix via the standard Householder QR algorithm requires approximately $2m^2(n - m/3)$ floating point operations, while computing the Singular Value Decomposition of a $n \times m$ is more expensive and often relies on intermediate QR decompositions [Trefethen and Bau, 1997]. Thus, these operations can become very costly

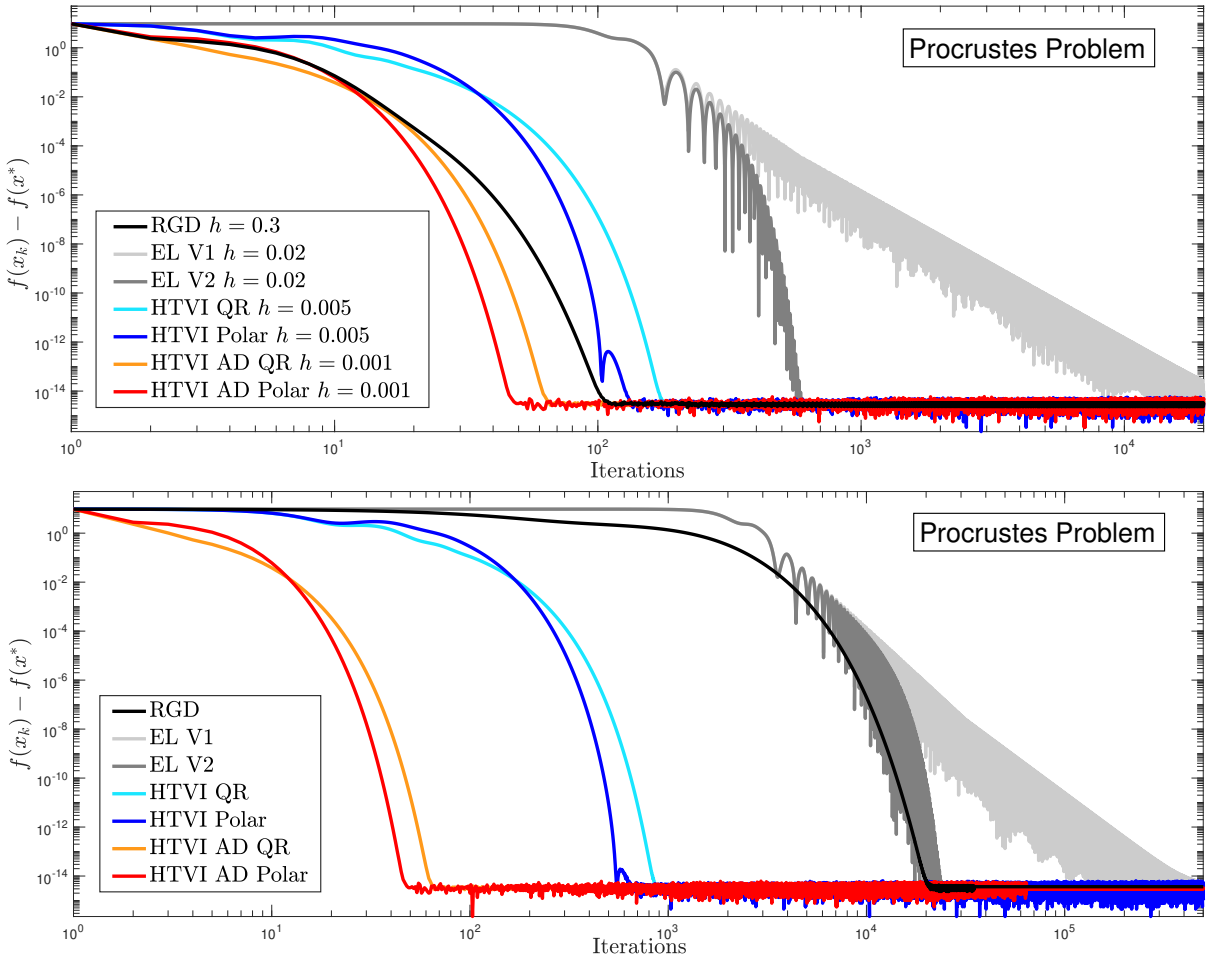


Figure 4.8: Direct and Adaptive (AD) HTVIs, Riemannian Gradient Descent (RGD), and Euler–Lagrange discretizations (EL V1 and EL V2) with different time-steps (top plot) and with the same time-step (bottom plot), for the Procrustes problem 4.4 on $\mathrm{St}(m, n)$.

as the dimension of the problem becomes large, in which case it might be beneficial to use approximate QR decompositions and Singular Value Decompositions. For instance, the projection based on the polar decomposition $Q \mapsto Q(Q^\top Q)^{-1/2}$ can be rewritten as $Q \mapsto Q(\mathbb{I}_m + (Q^\top Q - \mathbb{I}_m))^{-1/2}$, and provided the distance away from the Stiefel manifold is sufficiently small, the norm of $E = (Q^\top Q - \mathbb{I}_m)$ is small and we can approximate the projection by truncating its series expansion

$$Q(I_m + D)^{-1/2} \;=\; Q\left(\mathbb{I}_m - \frac{1}{2}D + \frac{3}{8}D^2 - \frac{5}{16}D^3 + \dots\right). \tag{4.39}$$

We have also tested the projection algorithm against the implicit algorithm from Section 4.5.2 and [Duruisseaux and Leok, 2022a] on the same optimization problems on $\mathbb{S}^{n-1}$ and $\mathrm{St}(m,n)$. Although both algorithms produced similar graphs for the error as a function of the iteration number, the explicit nature of our projection algorithm made every iteration significantly faster and overall the running time was reduced by several orders of magnitude, even on low-dimensional problems (for instance, 3 orders of magnitude on $\mathbb{S}^{5-1}$ and $\mathrm{St}(3,2)$, and 4 orders of magnitude on $\mathbb{S}^{100-1}$). Note that the projection algorithm was also easier to implement and tune than the implicit algorithm.

Overall, the gain in computational efficiency is preserved when the constraints are enforced via projections instead of being incorporated directly into the variational principles, and that the explicit nature of the resulting algorithms makes every iteration significantly faster and easier to tune than for the implicit algorithms from Section 4.5.2 and [Duruisseaux and Leok, 2022a]. As a consequence, if projections onto the constraint manifold can be computed efficiently, these projection based variational integrators form a class of efficient explicit algorithms for Riemannian accelerated optimization, and we believe that these algorithms are the most efficient methods to date which exploit the variational framework from [Duruisseaux and Leok, 2022d] on the Hamiltonian side.

4.5.4 Conclusion

Motivated by the observation made in the vector space setting in Section 3.3.5 and [Duruisseaux et al., 2021] that a careful use of adaptivity and symplecticity within the variational formulation of accelerated optimization could result in a significant gain in computational efficiency, discrete constrained variational integrators and projection-based variational integrators were constructed on the Hamiltonian side within the variational framework for Riemannian accelerated optimization of [Duruisseaux and Leok, 2022d]. Both approaches performed well in terms of number of iterations required to achieve convergence, but the implicit nature of the constrained integrators makes them much less desirable than the explicit projection-based algorithms which are easier to tune and become significantly more computational efficient as the dimension of the problem increases.

Although the Whitney and Nash Embedding Theorems [Whitney, 1944a,b; Nash, 1956] imply that there is no loss of generality when studying Riemannian manifolds only as submanifolds of Euclidean spaces, there are limitations to the constrained integration and projection-based strategies which rely on embeddings. Designing intrinsic methods that would exploit and preserve the symmetries and geometric properties of the Riemannian manifold and of the problem at hand could have computational advantages and could help improve our understanding of the acceleration phenomenon on Riemannian manifolds. Indeed, the embedding approach usually leads to higher-dimensional computations, and requires an effective way of constructing the embedding or a natural way of writing down equations that constrain the problem and the numerical solutions to the Riemannian manifold. Furthermore, most results in Riemannian geometry or results concerning specific Riemannian manifolds are proven from an intrinsic perspective because the embedding approach tends to flood intrinsic geometric properties of the manifold with superfluous information inherited from the additional dimensions of the embedding Euclidean space.

Developing an intrinsic extension of Hamiltonian variational integrators to manifolds will require some additional work, since the current approach involves Type II/III generating functions $H_d^+(q_k, p_{k+1})$, $H_d^-(p_k, q_{k+1})$, which depend on the position at one boundary point, and on the momentum at the other boundary point. This does not make intrinsic sense on a manifold, since one needs a base point in order to specify the corresponding cotangent space, and one should ideally consider constructing Hamiltonian variational integrator based on discrete Dirac mechanics [Leok and Ohsawa, 2011], which would yield a generating function $E_d^+(q_k, q_{k+1}, p_{k+1})$, $E_d^-(q_k, p_k, q_{k+1})$, that depends on the position at both boundary points and the momentum at one of the boundary points. This approach can be viewed as a discretization of the generalized energy $E(q, v, p) = \langle p, v\rangle - L(q, v)$, in contrast to the Hamiltonian $H(q, p) = \operatorname{ext}_v \langle p, v\rangle - L(q, v) = \langle p, v\rangle - L(q, v)\big|_{p=\frac{\partial L}{\partial v}}$. On the other hand, the formulation of Lagrangian variational integrators presented in the introduction of Section 2.6.1 makes sense intrinsically on manifolds,so we will exploit the framework for variable time-stepping in Lagrangian variational integration from Section 2.7.2 [Duruisseaux and Leok, 2023a] next, to design intrinsic time-adaptive Lagrangian variational integrators for accelerated optimization on Riemannian manifolds.

On a side note, it would also be interesting to extend the approach using discrete constrained variational integrators to the problem of optimization with nonintegrable constraints, which naturally leads to the question of whether vakonomic mechanics or nonholonomic mechanics is the appropriate description [Cortés et al., 2002]. In the context of optimization with nonintegrable constraints, the relevant extension will likely involve vakonomic variational integrators [Benito and Martín de Diego, 2005; Jiménez and Martín de Diego, 2012]. However, it would be interesting to relate the methods introduced in Section 4.5.2 and in [Duruisseaux and Leok, 2022a] to the existing work on variational integrators applied to optimal control problems [Junge et al., 2005; de León et al., 2007], and the discrete optimal control of nonholonomic dynamical systems would likely require a combination of the methods described in this section and nonholonomic integrators [Cortés and Martínez, 2001; de León et al., 2004; Fedorov and Zenkov, 2005; McLachlan and Perlmutter, 2006].

4.6 Riemannian Accelerated Optimization via Time-Adaptive Lagrangian Integrators

We have just discussed why designing integrators which evolve intrinsically on the Riemannian manifold and which would exploit and preserve the symmetries and geometric properties of the Riemannian manifold and of the problem at hand could prove very advantageous, and why the current formulation of Hamiltonian variational integration is not suitable to achieve this task. On the other hand, Lagrangian variational integrators are well-defined on manifolds, and many Lagrangian variational integrators have been derived on Riemannian manifolds, especially in the Lie group [Leok, 2004; Lee et al.; Hussein et al., 2006; Lee et al., 2007a,b; Lee, 2008; Bou-Rabee and Marsden, 2009; Nordkvist and Sanyal, 2010; Hall and Leok, 2015] and homogeneous space [Lee et al., 2009] settings.

The time-adaptive framework developed in Section 2.7.2 and [Duruisseaux and Leok, 2023a] makes it now possible to design time-adaptive Lagrangian integrators for accelerated optimization on these more general spaces, where it is more natural and easier to work on the Lagrangian side than on the Hamiltonian side.

We now construct time-adaptive Lagrangian variational integrators for accelerated optimization on Riemannian manifolds. Recall from Section 4.2 that the Riemannian p-Bregman Lagrangian $\mathcal{L}_p : T\mathcal{Q} \times \mathbb{R} \to \mathbb{R}$ is given by

$$\mathcal{L}_p(X, V, t) = \frac{t^{\lambda^{-1}\zeta p+1}}{2p}\langle V, V\rangle - Cpt^{(\lambda^{-1}\zeta+1)p-1} f(X), \tag{4.40}$$

where ζ and λ are constants having to do with the curvature of the manifold and the convexity of the objective function f.

Although it is possible to work on Riemannian manifolds, we will restrict ourselves to Lie groups for simplicity of exposition since there is more literature available on Lie group integrators than Riemannian integrators. We refer the reader to [Iserles et al., 2000; Christiansen et al., 2011; Celledoni et al., 2014, 2022] for very thorough surveys of the literature on Lie group methods, which acknowledge all the foundational contributions leading to the current state of Lie group integrator theory. In particular, the Crouch and Grossman approach [Crouch and Grossman, 1993], the Lewis and Simo approach [Lewis and Simo, 1994], Runge–Kutta–Munthe–Kaas methods [Munthe-Kaas, 1995, 1998, 1999; Casas and Owren, 2003], Magnus and Fer expansions [Iserles and Nørsett, 1999; Zanna, 1999; Blanes et al., 2008], and commutator-free Lie group methods [Celledoni et al., 2003] are outlined in these surveys.

Variational integrators have also been derived on the Lagrangian side in the Lie group setting [Leok, 2004; Lee et al.; Hussein et al., 2006; Lee et al., 2007a,b; Lee, 2008; Bou-Rabee and Marsden, 2009; Nordkvist and Sanyal, 2010; Hall and Leok, 2015].

Note as well that prior work is available on accelerated optimization via numerical integration of Bregman dynamics in the Lie group setting [Tao and Ohsawa, 2020; Lee et al., 2021].

Here, we will work in the setting described in Example 1.3 for considering a Lie group G with an associated Lie algebra $\mathfrak{g}$ as a Riemannian manifold. More precisely, we use a left trivialization $TG \simeq G \times \mathfrak{g}$, obtained via $(q, \dot{q}) \mapsto (q, L_{q^{-1}}\dot{q}) \equiv (q, \xi)$, and the Lie algebra $\mathfrak{g}$ is equipped with an inner product which induces an inner product on T_qG via left trivialization

$$(v \bullet w)_{T_qG} = (T_qL_{q^{-1}}v \bullet T_qL_{q^{-1}}w)_{\mathfrak{g}} \qquad \forall v, w \in T_qG. \tag{4.41}$$

With this inner product, we can identify $\mathfrak{g} \simeq \mathfrak{g}^*$ and also $T_qG \simeq T_q^*G \simeq G \times \mathfrak{g}^*$ via the Riesz representation. Then, $\mathbf{J} : \mathfrak{g} \to \mathfrak{g}^*$ is chosen such that $(\mathbf{J}(\xi) \bullet \zeta)$ is positive-definite and symmetric as a bilinear form of $\xi, \zeta \in \mathfrak{g}$. The resulting metric $\langle \cdot, \cdot \rangle : \mathfrak{g} \times \mathfrak{g} \to \mathbb{R}$ defined via $\langle \xi, \zeta \rangle = (\mathbf{J}(\xi) \bullet \zeta)$ serves as a left-invariant Riemannian metric on G.

We now introduce a discrete variational formulation of time-adaptive Lagrangian mechanics on Lie groups. Suppose we are given a partition $0 = \tau_0 < \tau_1 < \ldots < \tau_N = \mathcal{T}$ of the interval $[0, \mathcal{T}]$, and a discrete curve in $G \times \mathbb{R} \times \mathbb{R}$ denoted by $\{(q_k, \mathfrak{q}_k, \lambda_k)\}_{k=0}^N$ such that

$$q_k \approx q(\tau_k), \qquad \mathfrak{q}_k \approx \mathfrak{q}(\tau_k), \qquad \lambda_k \approx \lambda(\tau_k). \tag{4.42}$$

The discrete kinematics equation is chosen to be

$$q_{k+1} = q_k \star z_k, \tag{4.43}$$

where $z_k \in G$ represents the relative update over a single step.

Now, consider the discrete action functional,

$$\bar{\mathfrak{S}}_d\left(\{(q_k, \mathfrak{q}_k, \lambda_k)\}_{k=0}^N\right) = \sum_{k=0}^{N-1} \left[L_d(q_k, z_k, \mathfrak{q}_k, \mathfrak{q}_{k+1}) - \lambda_k \frac{\mathfrak{q}_{k+1} - \mathfrak{q}_k}{\tau_{k+1} - \tau_k} + \lambda_k g(\mathfrak{q}_k) \right] \frac{\mathfrak{q}_{k+1} - \mathfrak{q}_k}{\tau_{k+1} - \tau_k}, \tag{4.44}$$

where,

$$L_d(q_k, z_k, \mathfrak{q}_k, \mathfrak{q}_{k+1}) \approx \underset{\substack{(q,\mathfrak{q}) \in C^2([\tau_k, \tau_{k+1}], G \times \mathbb{R}) \\ (q,\mathfrak{q})(\tau_k) = (q_k, \mathfrak{q}_k),\ (q,\mathfrak{q})(\tau_{k+1}) = (q_k z_k, \mathfrak{q}_{k+1})}}{\operatorname{ext}} \int_{\tau_k}^{\tau_{k+1}} L\left(q, \frac{\xi}{g(\mathfrak{q})}, \mathfrak{q}\right) d\tau. \tag{4.45}$$

We can derive the following result which relates a discrete Type I variational principle to a set of discrete Euler–Lagrange equations:

Theorem 4.8. *The Type I discrete Hamilton's variational principle,*

$$\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=0, \tag{4.46}$$

where,

$$\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=\sum_{k=0}^{N-1}\left[L_d(q_k,z_k,\mathfrak{q}_k,\mathfrak{q}_{k+1})-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}, \tag{4.47}$$

is equivalent to the discrete extended Euler–Lagrange equations,

$$\mathfrak{q}_{k+1}=\mathfrak{q}_k+(\tau_{k+1}-\tau_k)g(\mathfrak{q}_k), \tag{4.48}$$

$$\mathrm{Ad}^*_{z_k^{-1}}\left(\mathrm{T}_e^*\mathrm{L}_{z_k}D_2L_{d_k}\right)=\mathrm{T}_e^*\mathrm{L}_{q_k}D_1L_{d_k}+\frac{\tau_{k+1}-\tau_k}{\mathfrak{q}_{k+1}-\mathfrak{q}_k}\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}\mathrm{T}_e^*\mathrm{L}_{z_{k-1}}D_2L_{d_{k-1}}, \tag{4.49}$$

and

$$\begin{aligned}&\left[D_3L_{d_k}+\lambda_k\frac{1}{\tau_{k+1}-\tau_k}+\lambda_k\nabla g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}-\frac{1}{\tau_{k+1}-\tau_k}\left[L_{d_k}-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\\&+\left[D_4L_{d_k}-\lambda_{k-1}\frac{1}{\tau_k-\tau_{k-1}}\right]\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\frac{1}{\tau_k-\tau_{k-1}}\left[L_{d_{k-1}}-\lambda_{k-1}\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\lambda_{k-1}g(\mathfrak{q}_{k-1})\right]=0,\end{aligned} \tag{4.50}$$

where L_{d_k} denotes $L_d(q_k,z_k,\mathfrak{q}_k,\mathfrak{q}_{k+1})$.

Proof. See Appendix A.3.3. □

Now, define two new quantities,

$$\mathfrak{p}_k=-D_3L_d(q_k,z_k,\mathfrak{q}_k,\mathfrak{q}_{k+1}), \tag{4.51}$$

$$\mu_k=\mathrm{Ad}^*_{z_k^{-1}}\left(\mathrm{T}_e^*\mathrm{L}_{z_k}D_2L_d(q_k,z_k,\mathfrak{q}_k,\mathfrak{q}_{k+1})\right)-\mathrm{T}_e^*\mathrm{L}_{q_k}D_1L_d(q_k,z_k,\mathfrak{q}_k,\mathfrak{q}_{k+1}), \tag{4.52}$$

which imply that

$$\mu_{k+1}=\frac{\tau_{k+2}-\tau_{k+1}}{\mathfrak{q}_{k+2}-\mathfrak{q}_{k+1}}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\mathrm{T}_e^*\mathrm{L}_{z_k}D_2L_d(q_k,z_k,\mathfrak{q}_k,\mathfrak{q}_{k+1}). \tag{4.53}$$

Then, with these definitions, if we use a constant time-step h in τ and substitute

$$g(\mathfrak{q}) = \frac{p}{\mathring{p}}\mathfrak{q}^{1-\mathring{p}/p}, \tag{4.54}$$

the discrete Euler–Lagrange equations can be rewritten as

$$\mu_k = \mathrm{Ad}^*_{z_k^{-1}}\left(\mathrm{T}^*_e\mathrm{L}_{z_k} D_2 L_d(q_k, z_k, \mathfrak{q}_k, \mathfrak{q}_{k+1})\right) - \mathrm{T}^*_e\mathrm{L}_{q_k} D_1 L_d(q_k, z_k, \mathfrak{q}_k, \mathfrak{q}_{k+1}), \tag{4.55}$$

$$\mu_{k+1} = \frac{\mathfrak{q}_k^{1-\mathring{p}/p}}{\mathfrak{q}_{k+1}^{1-\mathring{p}/p}}\mathrm{T}^*_e\mathrm{L}_{z_k} D_2 L_d(q_k, z_k, \mathfrak{q}_k, \mathfrak{q}_{k+1}), \tag{4.56}$$

$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + h\frac{p}{\mathring{p}}\mathfrak{q}_k^{1-\mathring{p}/p}, \tag{4.57}$$

$$\mathfrak{p}_{k+1} = \frac{\mathring{p}\left[\lambda_{k+1} - \lambda_k + L_{d_k} - L_{d_{k+1}}\right]}{hp\mathfrak{q}_{k+1}^{1-\mathring{p}/p}} + D_4 L_{d_k} + \frac{\lambda_{k+1}}{\mathfrak{q}_{k+1}}\left(1 - \frac{\mathring{p}}{p}\right). \tag{4.58}$$

In the Lie group setting, the Riemannian p-Bregman Lagrangian becomes

$$\mathscr{L}_p(q, \xi, t) = \frac{t^{\kappa p+1}}{2p}\langle \xi, \xi\rangle - Cpt^{(\kappa+1)p-1} f(q), \tag{4.59}$$

with corresponding Euler–Lagrange equation,

$$\frac{d\mathbf{J}(\xi)}{dt} + \frac{\kappa p + 1}{t}\mathbf{J}(\xi) - \mathrm{ad}^*_\xi \mathbf{J}(\xi) + Cp^2 t^{p-2}\nabla_{\mathrm{L}} f(q) = 0, \tag{4.60}$$

where $\nabla_{\mathrm{L}} f$ is the left-trivialized derivative of f, given by $\nabla_{\mathrm{L}} f(q) = \mathrm{T}^*_e\mathrm{L}_q(\mathrm{D}_q f(q))$.

We then consider the discrete Lagrangian,

$$L_d(q_k, z_k, \mathfrak{q}_k, \mathfrak{q}_{k+1}) = \frac{\mathfrak{q}_k^{\kappa p+1}}{hp(g(\mathfrak{q}_k))^2} T_d(z_k) - Chp\mathfrak{q}_k^{(\kappa+1)p-1} f(q_k), \tag{4.61}$$

where $T_d(z_k) \approx \frac{1}{2}\langle h\xi_k, h\xi_k\rangle$, which approximates

$$L_d(q_k, z_k, \mathfrak{q}_k, \mathfrak{q}_{k+1}) \approx \operatorname*{ext}_{\substack{(q,\mathfrak{q})\in C^2([\tau_k,\tau_{k+1}], G\times\mathbb{R}) \\ (q,\mathfrak{q})(\tau_k)=(q_k,\mathfrak{q}_k),\ (q,\mathfrak{q})(\tau_{k+1})=(q_k z_k, \mathfrak{q}_{k+1})}} \int_{\tau_k}^{\tau_{k+1}} L\left(q, \frac{\xi}{g(\mathfrak{q})}, \mathfrak{q}\right) d\tau. \tag{4.62}$$

Numerical Experiment on SO(3)

We work on the 3-dimensional Special Orthogonal group SO(3) (see Example 1.6 for more details about SO(3)). The inner product on $\mathfrak{so}(3)$ is given by

$$\left(\hat{\eta} \bullet \hat{\xi}\right)_{\mathfrak{so}(3)} = \frac{1}{2}\mathrm{Trace}\left(\hat{\eta}^{\mathsf{T}}\hat{\xi}\right) = \eta^{\mathsf{T}}\xi, \tag{4.63}$$

and the metric is chosen so that

$$\langle\hat{\eta},\hat{\xi}\rangle = \left(\mathbf{J}(\hat{\eta}) \bullet \hat{\xi}\right)_{\mathfrak{so}(3)} = \mathrm{Trace}\left(\hat{\eta}^{\mathsf{T}}J_d\hat{\xi}\right) = \eta^{\mathsf{T}}J\xi, \tag{4.64}$$

where $J \in \mathbb{R}^{3\times3}$ is a symmetric positive-definite matrix and $J_d = \frac{1}{2}\mathrm{Trace}(J)\mathbb{I}_3 - J$.

On SO(3), the Riemannian p-Bregman Lagrangian becomes

$$\mathcal{L}_p(R,\Omega,t) = \frac{t^{p+1}}{2p}\Omega^{\mathsf{T}}J\Omega - Cpt^{2p-1}f(R), \tag{4.65}$$

and the corresponding Euler–Lagrange equations are given by

$$J\dot{\Omega} + \frac{p+1}{t}J\Omega + \hat{\Omega}J\Omega + Cp^2t^{p-2}\nabla_{\mathrm{L}}f(R) = 0, \qquad \dot{R} = R\hat{\Omega}. \tag{4.66}$$

The discrete kinematics equation is written as

$$R_{k+1} = R_kZ_k, \tag{4.67}$$

where $Z_k \in \mathrm{SO}(3)$, and $\kappa = 1$ so we get the discrete Lagrangian,

$$L_d(R_k,Z_k,\mathfrak{R}_k,\mathfrak{R}_{k+1}) = \frac{\mathring{p}^2}{hp^3}\mathfrak{R}_k^{p-1+2\mathring{p}/p}T_d(Z_k) - Chp\mathfrak{R}_k^{2p-1}f(R_k). \tag{4.68}$$

As in [Lee et al., 2007a, 2021], the angular velocity is approximated with

$$\hat{\Omega}_k \approx \frac{1}{h}R_k^{\mathsf{T}}(R_{k+1}-R_k) = \frac{1}{h}(Z_k - \mathbb{I}_3) \tag{4.69}$$

so we can take

$$T_d(Z_k) = \mathrm{Trace}\left(\left[\mathbb{I}_3 - Z_k\right]J_d\right). \tag{4.70}$$

Differentiating this equation and using the identity $\mathrm{Trace}(-\hat{x}A) = (A-A^{\mathsf{T}})^{\vee}\cdot x$ yields

$$\mathrm{T}_I^*\mathrm{L}_{Z_k}\left(D_{Z_k}T_d(Z_k)\right) = \left(J_dZ_k - Z_k^{\mathsf{T}}J_d\right)^{\vee}. \tag{4.71}$$

Then, the discrete Euler–Lagrange equations for μ_k and μ_{k+1} become

$$\mu_k = \frac{\mathring{p}^2}{hp^3}\mathfrak{R}_k^{p-1+2\mathring{p}/p}\left(Z_k J_d - J_d Z_k^\top\right)^\vee + Chp\mathfrak{R}_k^{2p-1}\nabla_{\mathrm{L}} f(R_k), \tag{4.72}$$

$$\mu_{k+1} = \frac{\mathfrak{q}_k^{1-\mathring{p}/p}}{\mathfrak{q}_{k+1}^{1-\mathring{p}/p}} Z_k^\top \left[\mu_k - Chp\mathfrak{R}_k^{2p-1}\nabla_{\mathrm{L}} f(R_k)\right]. \tag{4.73}$$

Now, as described in [Lee et al., 2021], equation (4.72) can be solved explicitly when $J = \mathbb{I}_3$:

$$Z_k = \exp\left(\frac{\sin^{-1}\|a_k\|}{\|a_k\|}\hat{a}_k\right), \quad \text{where} \quad a_k = \frac{hp^3}{\mathring{p}^2}\mathfrak{R}_k^{1-p-2\mathring{p}/p}\left[\mu_k - Chp\mathfrak{R}_k^{2p-1}\nabla_{\mathrm{L}} f(R_k)\right]. \tag{4.74}$$

Therefore, we get the following **Adaptive LLGVI** (Adaptive Lagrangian Lie Group Variational Integrator)

$$Z_k = \exp\left(\frac{\sin^{-1}\|a_k\|}{\|a_k\|}\hat{a}_k\right), \quad \text{where} \quad a_k = \frac{hp^3}{\mathring{p}^2}\mathfrak{R}_k^{1-p-2\mathring{p}/p}\left[\mu_k - Chp\mathfrak{R}_k^{2p-1}\nabla_{\mathrm{L}} f(R_k)\right], \tag{4.75}$$

$$\mathfrak{R}_{k+1} = \mathfrak{R}_k + h\frac{p}{\mathring{p}}\mathfrak{R}_k^{1-\mathring{p}/p}, \tag{4.76}$$

$$\mu_{k+1} = \frac{\mathfrak{R}_k^{1-\mathring{p}/p}}{\mathfrak{R}_{k+1}^{1-\mathring{p}/p}} Z_k^\top \left[\mu_k - Chp\mathfrak{R}_k^{2p-1}\nabla_{\mathrm{L}} f(R_k)\right], \tag{4.77}$$

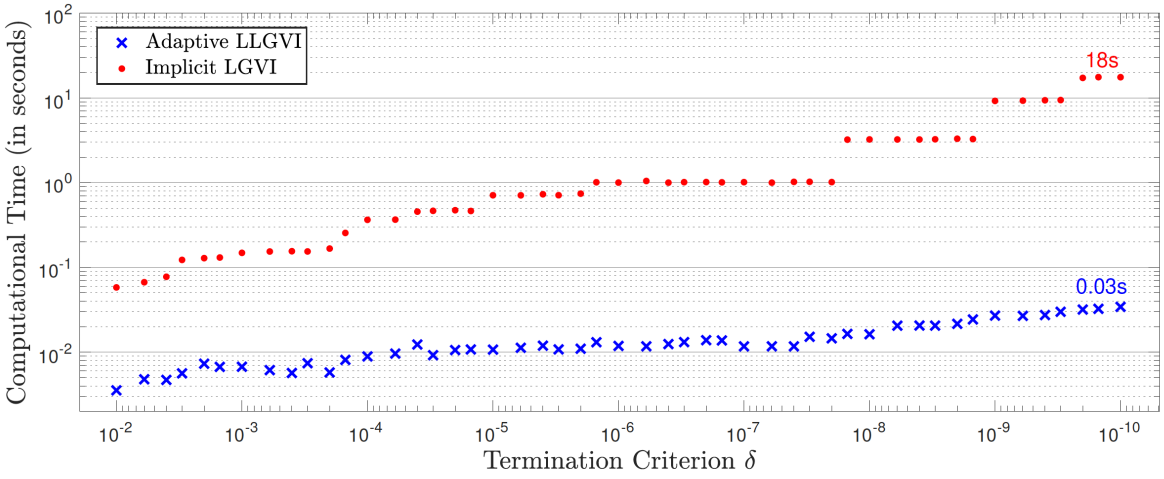
$$R_{k+1} = R_k Z_k. \tag{4.78}$$

We have tested this Adaptive LLGVI integrator on Wahba's Problem 4.5 on SO(3) against the Implicit Lie Group Variational Integrator (**Implicit LGVI**) introduced in [Lee et al., 2021]. The Implicit LGVI is a Lagrangian Lie group variational integrator which adaptively adjusts the stepsize at every step. It should be noted that these two time-adaptive approaches use adaptivity in two fundamentally different ways: our Adaptive LLGVI method uses *a priori* adaptivity based on known global properties of the family of differential equations considered (i.e. the time-rescaling symmetry of the family of Bregman dynamics), while the implicit method from [Lee et al., 2021] adapts the stepsize in an *a posteriori* way, by solving a system of nonlinear equations coming from an extended variational principle.

The results of our numerical experiments are presented in Figures 4.10 and 4.9. In these numerical experiments, we have used the termination criteria

$$|f(R_k) - f(R^*)| < \delta \qquad \text{and} \qquad |f(R_k) - f(R_{k-1})| < \delta. \tag{4.79}$$

We can see from Figure 4.10 that both algorithms preserve the orthogonality condition $R_k^\intercal R_k = \mathbb{I}_3$ very well. Now, we can observe from Figure 4.10 that although both algorithms follow the same curve in time t, they do not travel along this curve at the same speed. Despite the fact that the Adaptive LLGVI algorithm initially takes smaller time-steps, those time-steps eventually become much larger than for the Implicit LGVI algorithm, and as a result, the Adaptive LLGVI algorithm achieves the termination criteria in a smaller number of iterations, which can also be seen more explicitly in the table from Figure 4.9. Unlike the Implicit LGVI algorithm, the Adaptive LLGVI algorithm is explicit, so each iteration is much cheaper and is therefore significantly faster, as can be seen from the running times displayed in Figure 4.9. Furthermore, the Adaptive LLGVI algorithm is significantly easier to implement.



Termination Criterion δ	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
Adaptive LLGVI: Iterations	350	510	648	722	724	1181	1604	2237
Implicit LGVI: Iterations	179	474	970	1380	1392	4506	11992	16758
Adaptive LLGVI: Time (in seconds)	0.007	0.009	0.011	0.012	0.012	0.016	0.027	0.034
Implicit LGVI: Time (in seconds)	0.15	0.36	0.71	1.00	1.01	3.24	9.22	17.55

Figure 4.9: Time and number of iterations needed by the Adaptive LLGVI and Implicit LGVI algorithms to satisfy the termination criterion (4.79) on Wahba's Problem 4.5.

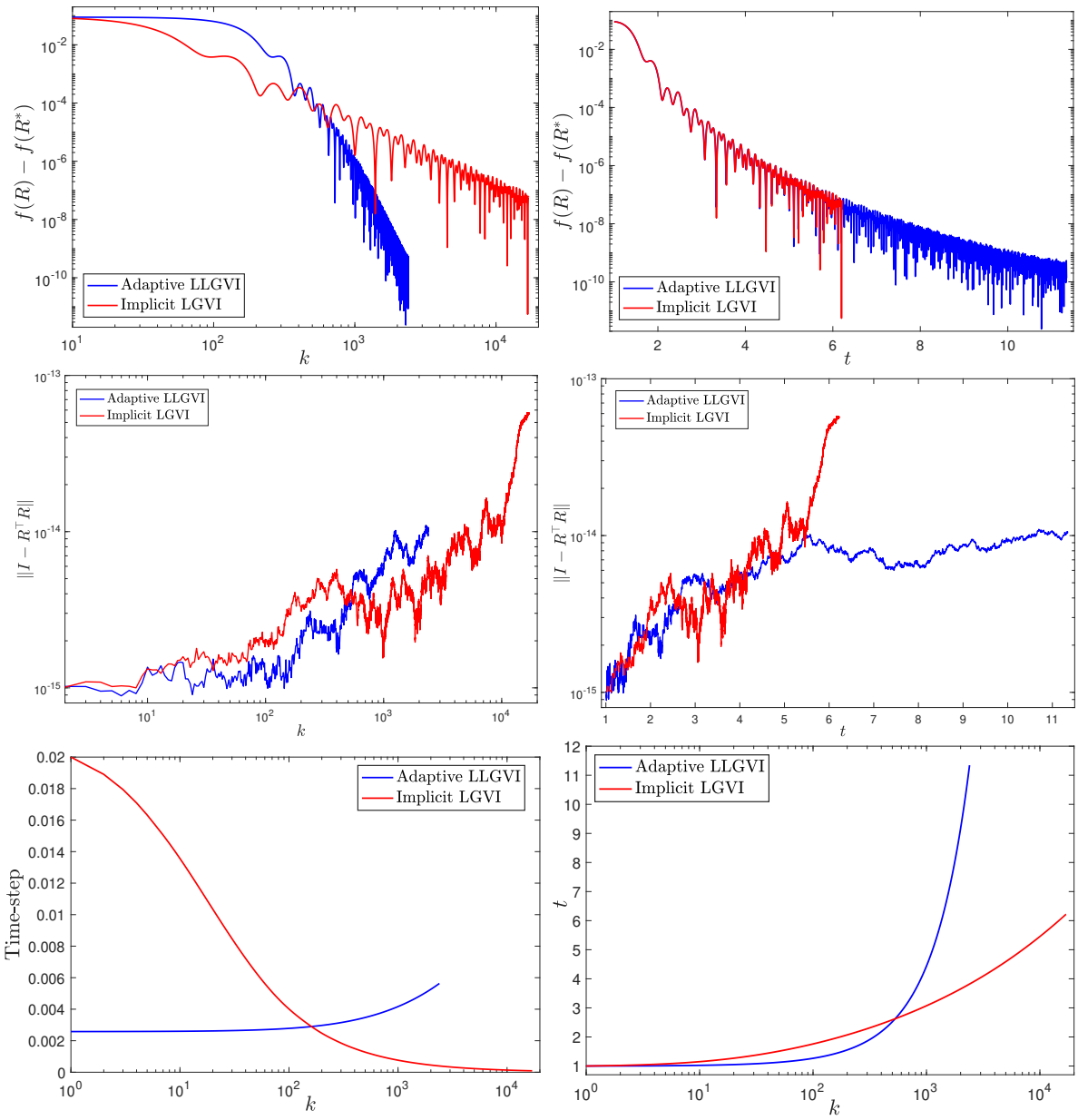


Figure 4.10: Comparison of the Adaptive LLGVI algorithm and of the Implicit LGVI algorithm from [Lee et al., 2021] with $p = 6$, to solve Wahba's problem (4.6).

Overall, we showed that the time-adaptive Lagrangian approach from Section 3.4 extends naturally to more general spaces such as Riemannian manifolds and Lie groups without having to face the difficulties experienced on the Hamiltonian side, and we saw that the resulting algorithms were significantly faster and easier to implement than other recently proposed time-adaptive variational integrators for optimization on Lie groups.

Conclusion

We have shown that on Riemannian manifolds, the convergence rate in continuous time of a geodesically convex or weakly quasi-convex function $f(x(t))$ to its optimal value can be accelerated to an arbitrary convergence rate, which extended the results of [Wibisono et al., 2016] from normed vector spaces to the Riemannian manifold setting. This accelerated rate of convergence is achieved along solutions of the Euler–Lagrange and Hamilton's equations corresponding to a family of time-dependent Bregman Lagrangian and Hamiltonian systems on Riemannian manifolds. As was demonstrated in the normed vector space setting, such families of Bregman dynamics can be used to construct practical, robust, and computationally efficient numerical optimization algorithms that outperform Nesterov's accelerated gradient method by considering geometric structure-preserving discretizations of the continuous-time flows. In analogy to what was done in the normed vector space setting in [Wibisono et al., 2016], we were also able to prove that the family of time-dependent Bregman Lagrangian and Hamiltonians on Riemannian manifolds is closed under time rescaling.

Inspired by the computational efficiency of the time-adaptive approach introduced in [Duruisseaux et al., 2021] in the normed vector space setting, we exploited the time rescaling property of the Bregman family on the Hamiltonian side via a carefully chosen Poincaré transformation that allowed us to integrate higher-order Bregman dynamics, using discrete constrained and variational integrators projection-based variational integrators, while benefiting from the computational efficiency of integrating a lower-order system. Both these Hamiltonian variational integrators exploited the structure of Euclidean spaces embedding the Riemannian manifolds. Although the construction of intrinsic Hamiltonian integrators that would exploit and preserve the symmetries and geometric properties of the Riemannian manifold and of the problem at hand could prove very advantageous, technical difficulties and issues coming from the current formulation of Hamiltonian variational integrators make the Hamiltonian perspective inadequate for the development of such intrinsic methods, at least for the moment. On the other hand, we noted that the Type I Lagrangian formulation of variational integrators was suitable for this task, and equipped

with the framework for variable time-stepping in Lagrangian integrators from [Duruisseaux and Leok, 2023a], we were able to construct intrinsic variational integrators for Riemannian accelerated optimization, with very promising results.

It would be desirable in future work to analyze the resulting discrete-time algorithms to have some convergence guarantees and rigorously establish their rates of convergence. However, proving that the discrete time algorithms perform analogously to the continuous dynamics is far from direct, even in the much simpler normed vector space setting, as the $\mathcal{O}(1/t^p)$ convergence rate for the continuous-time dynamics conflicts with the $\mathcal{O}(1/k^2)$ Nesterov barrier theorem for discrete-time algorithms. It would be very beneficial to better understand how to reconcile theoretically the arbitrarily high rate of convergence one expects from the continuous-time analysis, with Nesterov's barrier theorem on the rate of convergence of discrete-time algorithms.

Although some theoretical shadowing results have already been derived for certain simpler discrete Riemannian optimization algorithms for which the associated dynamical system is uniformly contracting, such a result might be very difficult to obtain for the momentum-based algorithms presented in this chapter because momentum methods lack contraction, are nondescending and highly oscillatory [Orvieto and Lucchi, 2019; Alimisis et al., 2020b]. Even on vector spaces, obtaining theoretical guarantees was a very challenging task, achieved in [Zhang et al., 2018] under additional assumptions. Generalizing these results to the Riemannian manifold setting would be much more challenging than a trivial generalization of these convergence results since the usual vector space operations and objects have to be replaced by their much more convoluted Riemannian generalizations which involve geodesics, parallel transport, covariant derivatives, Riemannian exponentials and logarithms.

It would also be desirable to improve the performance of the algorithms presented in this chapter, especially in terms of robustness and stability, and ease the tuning process before these algorithms can be used more conveniently and efficiently in practice. It could be advantageous to explore how the practical considerations explored in the normed vector space setting in the upcoming Chapter 5 extend to the Riemannian manifold setting.

➢ Chapter 4 contains original material from

① "A Variational Formulation of Accelerated Optimization on Riemannian Manifolds" by V. Duruisseaux and M. Leok. *SIAM Journal on Mathematics of Data Science*, Vol.4, No.2, pages 649-674, 2022

② "Accelerated Optimization on Riemannian Manifolds via Discrete Constrained Variational Integrators" by V. Duruisseaux and M. Leok. *Journal of Nonlinear Science*, Vol.32, No.42, 2022

③ "Accelerated optimization on Riemannian manifolds via Projected Variational Integrators" by V. Duruisseaux and M. Leok, 2022

④ "Time-adaptive Lagrangian Variational Integrators for Accelerated Optimization on Manifolds" by V. Duruisseaux and M. Leok. *Journal of Geometric Mechanics*, Vol.15, Issue 1, pages 224-255, 2023.

The dissertation author was the primary investigator and author of these papers.

5 Practical Perspectives on Symplectic Accelerated Optimization

5.1 Motivation

While the symplectic optimization approach provides a very general framework for constructing accelerated optimization algorithms, the real-world performance of these methods depends on the choice of numerous parameters.

We will now investigate practical considerations which can significantly boost the computational performance of the accelerated optimization algorithms presented in Chapters 3 and 4, and considerably simplify the tuning process. In particular, we will investigate how momentum restarting schemes ameliorate computational efficiency and robustness by reducing the undesirable effect of oscillations, and ease the tuning process by making time-adaptivity superfluous. We will also discuss how temporal looping helps avoiding instability issues caused by numerical precision, without harming the computational efficiency of the algorithms. Finally, we will compare the efficiency and robustness of different geometric integration techniques, and study the effects of the different parameters in the algorithms to inform and simplify tuning in practice. From this section will emerge symplectic accelerated optimization algorithms whose computational efficiency, stability and robustness have been improved, and which are now much simpler to use and tune for practical applications. The computational study presented in this chapter was first carried in [Duruisseaux and Leok, 2023b].

5.2 Special Bregman Subfamilies and Time-Rescalings of Interest

5.2.1 Review of the General Framework

Recall from Section 3.2 that the general Bregman Lagrangian and Hamiltonian are scalar-valued functions given by

$$L_{\alpha,\beta,\gamma}(q,v,t) \;=\; e^{\alpha_t+\gamma_t}\left[D_h(q+e^{-\alpha_t}v,q)-e^{\beta_t}f(q)\right], \tag{5.1}$$

$$H_{\alpha,\beta,\gamma}(q,r,t) \;=\; e^{\alpha_t+\gamma_t}\left[D_{h^*}(\nabla h(q)+e^{-\gamma_t}r,\nabla h(q))+e^{\beta_t}f(q)\right]. \tag{5.2}$$

Further recall that if the parameter functions α,β,γ satisfy the ideal scaling conditions

$$\dot{\beta}_t \le e^{\alpha_t} \qquad \text{and} \qquad \dot{\gamma}_t = e^{\alpha_t}, \tag{5.3}$$

then it follows from Theorem 1.1 in [Wibisono et al., 2016] that

$$f(q(t))-f(q^*) \;\le\; \mathcal{O}(e^{-\beta_t}), \tag{5.4}$$

where q^* is the desired minimizer of the objective function f.

From now on, we will take $h(q)=\frac{1}{2}\langle q,q\rangle$. Assuming that the parameter functions α,β,γ satisfy the ideal scaling conditions (3.7), the Bregman Lagrangian and Hamiltonian become

$$L_{\alpha,\beta,\gamma}(q,v,t) \;=\; \frac{1}{2}e^{\gamma_t-\alpha_t}\langle v,v\rangle - e^{\alpha_t+\beta_t+\gamma_t}f(q), \tag{5.5}$$

$$H_{\alpha,\beta,\gamma}(q,r,t) \;=\; \frac{1}{2}e^{\alpha_t-\gamma_t}\langle r,r\rangle + e^{\alpha_t+\beta_t+\gamma_t}f(q), \tag{5.6}$$

with corresponding Euler–Lagrange equation given by

$$\ddot{q}(t)+(e^{\alpha_t}-\dot{\alpha}_t)\,\dot{q}(t)+e^{2\alpha_t+\beta_t}\nabla f(q(t)) \;=\; 0. \tag{5.7}$$

We will focus here on two specific subfamilies of Bregman dynamics: polynomial and exponential Bregman dynamics.

5.2.2 Polynomial Subfamily

A first subfamily of Bregman dynamics of interest, indexed by a parameter $p > 0$, is given by the choice of parameter functions

$$\alpha_t = \log p - \log t, \qquad \beta_t = p \log t + \log C, \qquad \gamma_t = p \log t, \tag{5.8}$$

where $C > 0$ is a constant. These parameter functions α, β, γ satisfy the ideal scaling conditions (5.3), and the corresponding Lagrangian and Hamiltonian are given by

$$L_p(q, v, t) = \frac{t^{p+1}}{2p}\langle v, v\rangle - Cpt^{2p-1} f(q), \tag{5.9}$$

$$H_p(q, r, t) = \frac{p}{2t^{p+1}}\langle r, r\rangle + Cpt^{2p-1} f(q), \tag{5.10}$$

with corresponding Euler–Lagrange equation given by

$$\ddot{q}(t) + \frac{p+1}{t}\dot{q}(t) + Cp^2 t^{p-2} \nabla f(q(t)) = 0. \tag{5.11}$$

From Theorem 1.1 in [Wibisono et al., 2016], the evolution $q(t)$ resulting from this dynamical system satisfies the convergence rate

$$f(q(t)) - f(q^*) \leq \mathcal{O}(1/t^p). \tag{5.12}$$

Note that this Bregman subfamily has been exploited extensively in Chapter 3 and in [Wibisono et al., 2016; Betancourt et al., 2018; Duruisseaux et al., 2021; Duruisseaux and Leok, 2023a]. Also note that the special case where $p = 2$ and $C = 1/4$ corresponds to the limiting continuous differential equation introduced in [Su et al., 2016] for Nesterov's Accelerated Gradient method.

5.2.3 Exponential Subfamily

Another subfamily of Bregman dynamics of interest, indexed by a parameter $\eta > 0$, is given by the choice of parameter functions

$$\alpha_t = \log \eta, \qquad \beta_t = \eta t + \log C, \qquad \gamma_t = \eta t, \tag{5.13}$$

where $C > 0$ is a constant.

These parameter functions α, β, γ satisfy the ideal scaling conditions (5.3), and the corresponding Lagrangian and Hamiltonian are given by

$$L^\eta(q,v,t) \;=\; \frac{e^{\eta t}}{2\eta}\langle v,v\rangle - C\eta e^{2\eta t} f(q), \tag{5.14}$$

$$H^\eta(q,r,t) \;=\; \frac{\eta}{2e^{\eta t}}\langle r,r\rangle + C\eta e^{2\eta t} f(q), \tag{5.15}$$

with corresponding Euler–Lagrange equation given by

$$\ddot{q}(t) + \eta\dot{q} + C\eta^2 e^{\eta t}\nabla f(q(t)) \;=\; 0 \tag{5.16}$$

From Theorem 1.1 in [Wibisono et al., 2016], the evolution $q(t)$ resulting from this dynamical system satisfies the convergence rate

$$f(q(t)) - f(q^*) \;\leq\; \mathcal{O}\left(e^{-\eta t}\right). \tag{5.17}$$

5.2.4 Time-Rescalings of Interest

Recall from Theorem 3.1 that the family of Bregman dynamics is closed under time reparameterization: if $q(t)$ satisfies the Euler–Lagrange equations corresponding to the Bregman Lagrangian $L_{\alpha,\beta,\gamma}$, then the reparametrized curve $y(t) = q(\tau(t))$ satisfies the Euler–Lagrange equations corresponding to the Bregman Lagrangian $L_{\tilde{\alpha},\tilde{\beta},\tilde{\gamma}}$ where

$$\tilde{\alpha}_t = \alpha_{\tau(t)} + \log\dot{\tau}(t), \qquad \tilde{\beta}_t = \beta_{\tau(t)}, \qquad \tilde{\gamma}_t = \gamma_{\tau(t)}. \tag{5.18}$$

This allows us to transform the time-dependent Bregman dynamics into simpler autonomous systems in some appropriate extended phase-space via the use of carefully chosen time reparameterization. In Section 3.3.3, we rescaled time in a solution to the p-Bregman Euler–Lagrange equations from With the polynomial subfamily of Section 5.2.2 via $\tau(t) = t^{\mathring{p}/p}$ to obtain a solution to the $\mathring{p}$-Bregman Euler–Lagrange equations. We can similarly jump from one solution of Bregman dynamics from the exponential subfamily of Section 5.2.3 to another via $\tau(t) = \frac{\mathring{\eta}}{\eta}t$, or jump from exponential Bregman dynamics to polynomial Bregman dynamics via $\tau(t) = \frac{p}{\eta}\log t$, and vice-versa via $\tau(t) = e^{\eta t/p}$.

5.2.5 Transformed Extended Hamiltonians and Lagrangians

As before, to benefit from the advantages of symplectic integration of conservative Hamiltonian and Lagrangian systems while using variable time-steps, we will use the adaptive time-stepping frameworks presented in Sections 2.7.1 and 2.7.2.

On the Hamiltonian side, the Poincaré transformation from Section 2.7.1 together with the time reparameterizations presented in Section 5.2.4 allow to jump from one form of Bregman dynamics to another as follows:

1. **Polynomial-p to Polynomial-$\mathring{p}$**:

$$\tau(t) = t^{\mathring{p}/p} \quad \text{and} \quad g(t) = \frac{p}{\mathring{p}} t^{1-\mathring{p}/p} \tag{5.19}$$

yield the Poincaré Hamiltonian

$$\bar{H}_{p\to\mathring{p}}(\bar{q},\bar{r}) = \frac{p^2}{2\mathring{p}\mathfrak{q}^{p+\mathring{p}/p}}\langle r,r\rangle + \frac{Cp^2}{\mathring{p}}\mathfrak{q}^{2p-\mathring{p}/p} f(q) + \frac{p}{\mathring{p}}\mathfrak{r}\mathfrak{q}^{1-\mathring{p}/p}. \tag{5.20}$$

2. **Exponential-η to Exponential-$\mathring{\eta}$**:

$$\tau(t) = \frac{\mathring{\eta}}{\eta} t \quad \text{and} \quad g(t) = \frac{\eta}{\mathring{\eta}} \tag{5.21}$$

yield the Poincaré Hamiltonian

$$\bar{H}^{\eta\to\mathring{\eta}}(\bar{q},\bar{r}) = \frac{\eta^2}{2\mathring{\eta}e^{\eta\mathfrak{q}}}\langle r,r\rangle + \frac{C\eta^2}{\mathring{\eta}} e^{2\eta\mathfrak{q}} f(q) + \frac{\eta}{\mathring{\eta}}\mathfrak{r}. \tag{5.22}$$

3. **Exponential-η to Polynomial-p**:

$$\tau(t) = \frac{p}{\eta}\log t \quad \text{and} \quad g(t) = \frac{\eta}{p} t \tag{5.23}$$

yield the Poincaré Hamiltonian

$$\bar{H}^{\eta}_{\to p}(\bar{q},\bar{r}) = \frac{\mathfrak{q}\eta^2}{2pe^{\eta\mathfrak{q}}}\langle r,r\rangle + \frac{C\mathfrak{q}\eta^2}{p} e^{2\eta\mathfrak{q}} f(q) + \frac{\eta}{p}\mathfrak{q}\mathfrak{r}. \tag{5.24}$$

4. **Polynomial-p to Exponential-η**:

$$\tau(t) = e^{\eta t/p} \quad \text{and} \quad g(t) = \frac{p}{\eta} e^{-\eta t/p} \tag{5.25}$$

yield the Poincaré Hamiltonian

$$\bar{H}_p^{\to\eta}(\bar{q},\bar{r}) = e^{-\frac{\eta}{p}\mathfrak{q}}\left(\frac{p^2}{2\eta\mathfrak{q}^{p+1}}\langle r,r\rangle + \frac{Cp^2}{\eta}\mathfrak{q}^{2p-1} f(q) + \frac{p}{\eta}\mathfrak{r}\right). \tag{5.26}$$

On the Lagrangian side, we can use the framework of Section 2.7.2 and consider the discrete Lagrangian

$$L_d(q_k, \mathfrak{q}_k, q_{k+1}, \mathfrak{q}_{k+1}) \approx \underset{\substack{(q,\mathfrak{q})\in C^2([\tau_k,\tau_{k+1}],\mathcal{Q}\times\mathbb{R}) \\ (q,\mathfrak{q})(\tau_k)=(q_k,\mathfrak{q}_k),\ (q,\mathfrak{q})(\tau_{k+1})=(q_{k+1},\mathfrak{q}_{k+1})}}{\operatorname{ext}} \int_{\tau_k}^{\tau_{k+1}} L\left(q, \frac{q'}{g(\mathfrak{q})}, \mathfrak{q}\right) d\tau, \quad (5.27)$$

where $0 = \tau_0 < \tau_1 < \ldots < \tau_N$ partitions the time interval of interest, and $\{(q_k, \mathfrak{q}_k)\}_{k=0}^N$ is such that $q_k \approx q(\tau_k)$ and $\mathfrak{q}_k \approx \mathfrak{q}(\tau_k)$. The corresponding discrete extended Euler–Lagrange equations of interest for the accelerated optimization application are given by

$$\begin{aligned} p_k &= -D_1 L_d(q_k, \mathfrak{q}_k, q_{k+1}, \mathfrak{q}_{k+1}), \\ p_{k+1} &= \frac{g(\mathfrak{q}_k)}{g(\mathfrak{q}_{k+1})} D_3 L_d(q_k, \mathfrak{q}_k, q_{k+1}, \mathfrak{q}_{k+1}), \\ \mathfrak{q}_{k+1} &= \mathfrak{q}_k + h g(\mathfrak{q}_k). \end{aligned} \quad (5.28)$$

We can then use one of the monitor functions

$$g(t) = \frac{p}{\mathring{p}} t^{1-\mathring{p}/p}, \qquad g(t) = \frac{\eta}{\mathring{\eta}}, \qquad g(t) = \frac{\eta}{p} t, \qquad g(t) = \frac{p}{\eta} e^{-\eta t/p}, \quad (5.29)$$

to jump from one type of Bregman Lagrangian to another.

5.3 Numerical Methods and Problems of Interest

5.3.1 Numerical Methods

We now present four different methods for designing symplectic integrators for the Bregman Lagrangian and Bregman Hamiltonian systems. Keeping in mind the desired applications in machine learning where problem sizes and data sets are very large, we restrict ourselves to explicit first-order optimization algorithms. Each of these four methods will be used within the four different adaptive approaches presented in Section 5.2.4 (polynomial, exponential, polynomial-to-exponential, and exponential-to-polynomial), to obtain sixteen distinct algorithms.

Hamiltonian Taylor Variational Integrator (HTVI)

Proceeding as in Section 3.3, we can derive the Hamiltonian Taylor Variational Integrator (HTVI),

$$\begin{aligned} p_{k+1} &= p_k - hD_1H(q_k, p_{k+1}), \\ q_{k+1} &= q_k + hD_2H(q_k, p_{k+1}). \end{aligned} \tag{5.30}$$

These updates recover the Symplectic Euler method [Hairer et al., 2006], which is a popular symplectic integrator of order 1.

Lagrangian Taylor Variational Integrator (LTVI)

As in Section 3.4, we can define a discrete Lagrangian,

$$L_d(\bar{q}_0, \bar{q}_1) = hL_p\left(q_0, \frac{q_1 - q_0}{hg(\mathfrak{q}_0)}, \mathfrak{q}_0\right), \tag{5.31}$$

and the updates for the Lagrangian Taylor Variational Integrator (LTVI) can be obtained from the discrete extended Euler–Lagrange equations (5.28).

Störmer-Verlet (SV)

A popular symplectic integrator is the Störmer–Verlet (SV) method,

$$\begin{aligned} p_{k+1/2} &= p_k - \frac{h}{2}D_1H(q_k, p_{k+1/2}), \\ q_{k+1} &= q_k + \frac{h}{2}\left[D_2H(q_k, p_{k+1/2}) + D_2H(q_{k+1}, p_{k+1/2})\right], \\ p_{k+1} &= p_{k+1/2} - \frac{h}{2}D_1H(q_{k+1}, p_{k+1/2}), \end{aligned} \tag{5.32}$$

which is a symmetric symplectic integrator of order 2 (see [Hairer et al., 2006]). A very detailed description of the Störmer–Verlet method, its different interpretations, and its beneficial numerical properties can be found in [Hairer et al., 2003]. Note however that in the polynomial and polynomial-to-exponential frameworks, the update for $\mathfrak{q}$ in the resulting integrators becomes implicit (see the algorithms PolySV and PolyToExpoSV), which makes these integrators less desirable. For the accelerated optimization application, we will usually be able to combine the first and last updates for the momentum vector p into a single update and save roughly a third of the computational time. This is because Störmer–Verlet is conjugate to symplectic Euler.

Symmetric Leapfrog Composition of Component Dynamics (SLC)

The idea is to decompose the vector field into its components,

$$\frac{d}{d\tau} = \frac{dq}{d\tau}\frac{d}{dq} + \frac{d\mathfrak{q}}{d\tau}\frac{d}{d\mathfrak{q}} + \frac{dr}{d\tau}\frac{d}{dr} + \frac{d\mathfrak{r}}{d\tau}\frac{d}{d\mathfrak{r}} = \frac{\partial H}{\partial r}\frac{d}{dq} + \frac{\partial H}{\partial \mathfrak{r}}\frac{d}{d\mathfrak{q}} - \frac{\partial H}{\partial q}\frac{d}{dr} - \frac{\partial H}{\partial \mathfrak{q}}\frac{d}{d\mathfrak{r}} = \mathcal{A} + \mathcal{B} + \mathcal{C} + \mathcal{D},$$

and then combine the corresponding component dynamics using a symmetric leapfrog composition

$$\Phi_h = \exp\left(\frac{h}{2}\mathcal{D}\right)\circ\exp\left(\frac{h}{2}\mathcal{C}\right)\circ\exp\left(\frac{h}{2}\mathcal{B}\right)\circ\exp\left(h\mathcal{A}\right)\circ\exp\left(\frac{h}{2}\mathcal{B}\right)\circ\exp\left(\frac{h}{2}\mathcal{C}\right)\circ\exp\left(\frac{h}{2}\mathcal{D}\right) \quad (5.33)$$

which satisfies $\Phi_h = \exp(hH) + \mathcal{O}(h^3)$ (can be proven using the Baker–Campbell–Hausdorff formula). This strategy is similar to the integrator from Section 3.3 of [Betancourt et al., 2018] and the Splitting algorithms from Section 3.3.4.

As an example, for

$$\bar{H}_{p\to\mathring{p}}(\bar{q},\bar{r}) = \frac{p^2}{2\mathring{p}\mathfrak{q}^{p+\mathring{p}/p}}\langle r, r\rangle + \frac{Cp^2}{\mathring{p}}\mathfrak{q}^{2p-\mathring{p}/p}f(q) + \frac{p}{\mathring{p}}\mathfrak{r}\mathfrak{q}^{1-\mathring{p}/p}, \quad (5.34)$$

the components of the vector field are given by

$$\mathcal{A} = \frac{p^2}{\mathring{p}\mathfrak{q}^{p+\mathring{p}/p}} r\frac{d}{dq}, \quad \mathcal{B} = \frac{p}{\mathring{p}}\mathfrak{q}^{1-\mathring{p}/p}\frac{d}{d\mathfrak{q}}, \quad \mathcal{C} = -\frac{Cp^2}{\mathring{p}}\mathfrak{q}^{2p-\mathring{p}/p}\nabla f(q)\frac{d}{dr}, \quad \mathcal{D} = -\frac{\partial \bar{H}_{p\to\mathring{p}}}{\partial \mathfrak{q}}\frac{d}{d\mathfrak{r}}. \quad (5.35)$$

Then, the corresponding component dynamics are given as follows:

- $\exp(h\mathcal{A})$ yields the update $q \leftarrow q + h\frac{p^2}{\mathring{p}\mathfrak{q}^{p+\mathring{p}/p}}r$
- $\exp(h\mathcal{C})$ yields the update $r \leftarrow r - h\frac{Cp^2}{\mathring{p}}\mathfrak{q}^{2p-\mathring{p}/p}\nabla f(q)$
- $\exp(h\mathcal{B})$ yields the differential equation $\mathfrak{q}' = \frac{p}{\mathring{p}}\mathfrak{q}^{1-\mathring{p}/p}$, which can be solved exactly to obtain the update $\mathfrak{q} \leftarrow (\mathfrak{q}^{\mathring{p}/p} + h)^{p/\mathring{p}}$

Note that the updates corresponding to $\exp(h\mathcal{A})$, $\exp(h\mathcal{B})$, and $\exp(h\mathcal{C})$ do not involve the variable $\mathfrak{r}$, and in practice, we are not interested in the evolution of $\mathfrak{r}$, so we can simplify the composition into

$$\Phi_h = \exp\left(\frac{h}{2}\mathcal{C}\right) \circ \exp\left(\frac{h}{2}\mathcal{B}\right) \circ \exp\left(h\mathcal{A}\right) \circ \exp\left(\frac{h}{2}\mathcal{B}\right) \circ \exp\left(\frac{h}{2}\mathcal{C}\right), \quad (5.36)$$

This gives the PolySLC algorithm,

$$\begin{aligned} r &\leftarrow r - \frac{Cp^2}{2\mathring{p}} h\mathfrak{q}^{2p-\mathring{p}/p} \nabla f(q), \\ \mathfrak{q} &\leftarrow \left(\mathfrak{q}^{\mathring{p}/p} + \frac{h}{2}\right)^{p/\mathring{p}}, \\ q &\leftarrow q + \frac{hp^2}{\mathring{p}\mathfrak{q}^{p+\mathring{p}/p}} r, \\ \mathfrak{q} &\leftarrow \left(\mathfrak{q}^{\mathring{p}/p} + \frac{h}{2}\right)^{p/\mathring{p}}, \\ r &\leftarrow r - \frac{Cp^2}{2\mathring{p}} h\mathfrak{q}^{2p-\mathring{p}/p} \nabla f(q). \end{aligned} \tag{5.37}$$

We have chosen to place $\exp(h\mathcal{A})$ in the middle of the symmetric composition (5.36) so that the gradient ∇f only needs to be evaluated at the iterates $\{q_k\}_{k\in\mathbb{N}}$ and can also be used without further computations in stopping criteria or momentum restarting schemes. We have also chosen to place $\exp(h\mathcal{C})$ at the left-hand and right-hand of the symmetric composition (5.36) so that in practice, we may combine the first and last updates for the vector r into a single update (instead of only being able to combine the first and last updates for the scalar $\mathfrak{q}$ into a single update), and thus save roughly a third of the computational time.

Remark 5.1. *It was observed in [Duruisseaux et al., 2021] that the symplecticity of the integrator was essential for the efficient, robust, and stable discretization of these variational flows describing accelerated optimization. Therefore, we will not consider non-symplectic methods here. Higher-order explicit symplectic integrators can be derived as well, leveraging higher-order compositions such as Yoshida splittings [Yoshida, 1990], but it was observed in [Duruisseaux et al., 2021] that these require a larger number of evaluations of the objective function and of its gradient at each step (7 for Yoshida's fourth-order splitting, and 19 for Yoshida's sixth-order splitting, for instance). As a consequence, the resulting algorithms would not be competitive in terms of computational time and number of gradient evaluations, since the other methods usually converge in a similar number of iterations but only require one gradient evaluation per iteration.*

Resulting Algorithms

PolyHTVI

$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + h\frac{p}{\mathring{p}}\mathfrak{q}_k^{1-\mathring{p}/p}$$
$$r_{k+1} = r_k - \frac{p^2}{\mathring{p}}Ch\mathfrak{q}_k^{2p-\mathring{p}/p}\nabla f(q_k)$$
$$q_{k+1} = q_k + \frac{p^2}{\mathring{p}}h\mathfrak{q}_k^{-p-\mathring{p}/p}r_{k+1}$$

PolyLTVI

$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + h\frac{p}{\mathring{p}}\mathfrak{q}_k^{1-\mathring{p}/p}$$
$$q_{k+1} = q_k + \frac{hp^3}{\mathring{p}^2\mathfrak{q}_k^{p-1+2\mathring{p}/p}}r_k - \frac{Ch^2p^4}{\mathring{p}^2}\mathfrak{q}_k^{p-2\mathring{p}/p}\nabla f(q_k)$$
$$r_{k+1} = \frac{\mathring{p}^2\mathfrak{q}_k^{p+\mathring{p}/p}}{hp^3\mathfrak{q}_{k+1}^{1-\mathring{p}/p}}(q_{k+1} - q_k)$$

PolySLC

$$r \leftarrow r - \frac{Cp^2}{2\mathring{p}}h\mathfrak{q}^{2p-\mathring{p}/p}\nabla f(q)$$
$$\mathfrak{q} \leftarrow \left(\mathfrak{q}^{\mathring{p}/p} + \frac{h}{2}\right)^{p/\mathring{p}}$$
$$q \leftarrow q + \frac{hp^2}{\mathring{p}\mathfrak{q}^{p+\mathring{p}/p}}r$$
$$\mathfrak{q} \leftarrow \left(\mathfrak{q}^{\mathring{p}/p} + \frac{h}{2}\right)^{p/\mathring{p}}$$
$$r \leftarrow r - \frac{Cp^2}{2\mathring{p}}h\mathfrak{q}^{2p-\mathring{p}/p}\nabla f(q)$$

PolySV

$$r_{k+\frac{1}{2}} = r_k - \frac{p^2}{2\mathring{p}}Ch\mathfrak{q}_k^{2p-\mathring{p}/p}\nabla f(q_k)$$
$$\text{Solve} \quad \mathfrak{q}_{k+1} = \mathfrak{q}_k + \frac{hp}{2\mathring{p}}\left(\mathfrak{q}_k^{1-\mathring{p}/p} + \mathfrak{q}_{k+1}^{1-\mathring{p}/p}\right)$$
$$q_{k+1} = q_k + \frac{hp^2}{2\mathring{p}}\left(\mathfrak{q}_k^{-p-\mathring{p}/p} + \mathfrak{q}_{k+1}^{-p-\mathring{p}/p}\right)r_{k+\frac{1}{2}}$$
$$r_{k+1} = r_{k+\frac{1}{2}} - \frac{p^2}{2\mathring{p}}Ch\mathfrak{q}_{k+1}^{2p-\mathring{p}/p}\nabla f(q_{k+1})$$

ExpoHTVI

$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + \frac{\eta}{\mathring{\eta}}h$$
$$r_{k+1} = r_k - \frac{\eta^2}{\mathring{\eta}}Che^{2\eta\mathfrak{q}_k}\nabla f(q_k)$$
$$q_{k+1} = q_k + \frac{\eta^2}{\mathring{\eta}}he^{-\eta\mathfrak{q}_k}r_{k+1}$$

ExpoLTVI

$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + \frac{\eta}{\mathring{\eta}}h$$
$$q_{k+1} = q_k + \frac{h\eta^3}{\mathring{\eta}^2}e^{-\eta\mathfrak{q}_k}r_k - \frac{Ch^2\eta^4}{\mathring{\eta}^2}e^{\eta\mathfrak{q}_k}\nabla f(q_k)$$
$$r_{k+1} = \frac{\mathring{\eta}^2}{h\eta^3}e^{\eta\mathfrak{q}_k}(q_{k+1} - q_k)$$

ExpoSLC

$$
\begin{aligned}
r &\leftarrow r - \frac{Ch\eta^2}{2\mathring{\eta}} e^{2\eta\mathfrak{q}} \nabla f(q) \\
\mathfrak{q} &\leftarrow \mathfrak{q} + \frac{h\eta}{2\mathring{\eta}} \\
q &\leftarrow q + \frac{h\eta^2}{\mathring{\eta} e^{\eta\mathfrak{q}}} r \\
\mathfrak{q} &\leftarrow \mathfrak{q} + \frac{h\eta}{2\mathring{\eta}} \\
r &\leftarrow r - \frac{Ch\eta^2}{2\mathring{\eta}} e^{2\eta\mathfrak{q}} \nabla f(q)
\end{aligned}
$$

ExpoSV

$$
\begin{aligned}
r_{k+\frac{1}{2}} &= r_k - \frac{\eta^2}{2\mathring{\eta}} Che^{2\eta\mathfrak{q}_k} \nabla f(q_k) \\
\mathfrak{q}_{k+1} &= \mathfrak{q}_k + \frac{\eta}{\mathring{\eta}} h \\
q_{k+1} &= q_k + \frac{h\eta^2}{2\mathring{\eta}} \left(e^{-\eta\mathfrak{q}_{k+1}} + e^{-\eta\mathfrak{q}_k} \right) r_{k+\frac{1}{2}} \\
r_{k+1} &= r_{k+\frac{1}{2}} - \frac{\eta^2}{2\mathring{\eta}} Che^{2\eta\mathfrak{q}_{k+1}} \nabla f(q_{k+1})
\end{aligned}
$$

ExpoToPolyHTVI

$$
\begin{aligned}
\mathfrak{q}_{k+1} &= \left(1 + \frac{\eta h}{p}\right) \mathfrak{q}_k \\
r_{k+1} &= r_k - \frac{\eta^2}{p} Ch\mathfrak{q}_k e^{2\eta\mathfrak{q}_k} \nabla f(q_k) \\
q_{k+1} &= q_k + \frac{h\eta^2}{p e^{\eta\mathfrak{q}_k}} \mathfrak{q}_k r_{k+1}
\end{aligned}
$$

ExpoToPolyLTVI

$$
\begin{aligned}
\mathfrak{q}_{k+1} &= \left(1 + \frac{\eta h}{p}\right) \mathfrak{q}_k, \\
q_{k+1} &= q_k + \frac{h\mathfrak{q}_k^2 \eta^3}{p^2 e^{\eta\mathfrak{q}_k}} r_k - \frac{Ch^2\eta^4}{p^2} \mathfrak{q}_k^2 e^{\eta\mathfrak{q}_k} \nabla f(q_k), \\
r_{k+1} &= \frac{p(p+\eta h)}{h\eta^3 \mathfrak{q}_k^2} e^{\eta\mathfrak{q}_k} (q_{k+1} - q_k).
\end{aligned}
$$

ExpoToPolySLC

$$
\begin{aligned}
r &\leftarrow r - \frac{Ch\mathfrak{q}\eta^2}{2p} e^{2\eta\mathfrak{q}} \nabla f(q) \\
\mathfrak{q} &\leftarrow \mathfrak{q} e^{\frac{\eta h}{2p}} \\
q &\leftarrow q + \frac{h\mathfrak{q}\eta^2}{p e^{\eta\mathfrak{q}}} r \\
\mathfrak{q} &\leftarrow \mathfrak{q} e^{\frac{\eta h}{2p}} \\
r &\leftarrow r - \frac{Ch\mathfrak{q}\eta^2}{2p} e^{2\eta\mathfrak{q}} \nabla f(q)
\end{aligned}
$$

ExpoToPolySV

$$
\begin{aligned}
r_{k+\frac{1}{2}} &= r_k - \frac{\eta^2}{2p} Ch\mathfrak{q}_k e^{2\eta\mathfrak{q}_k} \nabla f(q_k), \\
\mathfrak{q}_{k+1} &= \frac{2p + \eta h}{2p - \eta h} \mathfrak{q}_k, \\
q_{k+1} &= q_k + \frac{h\eta^2}{2p} \left(\mathfrak{q}_k e^{-\eta\mathfrak{q}_k} + \mathfrak{q}_{k+1} e^{-\eta\mathfrak{q}_{k+1}} \right) r_{k+\frac{1}{2}}, \\
r_{k+1} &= r_{k+\frac{1}{2}} - \frac{\eta^2}{2p} Ch\mathfrak{q}_{k+1} e^{2\eta\mathfrak{q}_{k+1}} \nabla f(q_{k+1}),
\end{aligned}
$$

PolyToExpoHTVI

$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + h\frac{p}{\eta}e^{-\frac{\eta}{p}\mathfrak{q}_k}$$
$$r_{k+1} = r_k - \frac{Chp^2}{\eta e^{\frac{\eta}{p}\mathfrak{q}_k}}\mathfrak{q}_k^{2p-1}\nabla f(q_k)$$
$$q_{k+1} = q_k + h\frac{p^2}{\eta\mathfrak{q}_k^{p+1}}e^{-\frac{\eta}{p}\mathfrak{q}_k}r_{k+1}$$

PolyToExpoLTVI

$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + h\frac{p}{\eta}e^{-\frac{\eta}{p}\mathfrak{q}_k}$$
$$q_{k+1} = q_k + \frac{hp^3}{\eta^2\mathfrak{q}_k^{p+1}e^{\frac{2\eta}{p}\mathfrak{q}_k}}r_k - \frac{Ch^2p^4}{\eta^2 e^{\frac{2\eta}{p}\mathfrak{q}_k}}\mathfrak{q}_k^{p-2}\nabla f(q_k)$$
$$r_{k+1} = \frac{\eta^2\mathfrak{q}_k^{p+1}}{hp^3}e^{\frac{\eta}{p}(\mathfrak{q}_{k+1}+\mathfrak{q}_k)}(q_{k+1}-q_k)$$

PolyToExpoSLC

$$r \leftarrow r - h\frac{Cp^2}{2\eta}\mathfrak{q}^{2p-1}e^{-\frac{\eta}{p}\mathfrak{q}}\nabla f(q)$$
$$\mathfrak{q} \leftarrow \frac{p}{\eta}\log\left(e^{\frac{\eta}{p}\mathfrak{q}} + \frac{h}{2}\right)$$
$$q \leftarrow q + \frac{hp^2}{\eta\mathfrak{q}^{p+1}}e^{-\frac{\eta}{p}\mathfrak{q}}r$$
$$\mathfrak{q} \leftarrow \frac{p}{\eta}\log\left(e^{\frac{\eta}{p}\mathfrak{q}} + \frac{h}{2}\right)$$
$$r \leftarrow r - h\frac{Cp^2}{2\eta}\mathfrak{q}^{2p-1}e^{-\frac{\eta}{p}\mathfrak{q}}\nabla f(q)$$

PolyToExpoSV

$$r_{k+\frac{1}{2}} = r_k - \frac{p^2}{2\eta}Ch\mathfrak{q}_k^{2p-1}e^{-\frac{\eta}{p}\mathfrak{q}_k}\nabla f(q_k)$$
$$\text{Solve}\quad \mathfrak{q}_{k+1} = \mathfrak{q}_k + \frac{hp}{2\eta}\left(e^{-\frac{\eta}{p}\mathfrak{q}_k} + e^{-\frac{\eta}{p}\mathfrak{q}_{k+1}}\right)$$
$$q_{k+1} = q_k + \frac{hp^2}{2\eta}\left(\mathfrak{q}_k^{-p-1}e^{-\frac{\eta}{p}\mathfrak{q}_k} + \mathfrak{q}_{k+1}^{-p-1}e^{-\frac{\eta}{p}\mathfrak{q}_{k+1}}\right)r_{k+\frac{1}{2}}$$
$$r_{k+1} = r_{k+\frac{1}{2}} - \frac{p^2}{2\eta}Ch\mathfrak{q}_{k+1}^{2p-1}e^{-\frac{\eta}{p}\mathfrak{q}_{k+1}}\nabla f(q_{k+1})$$

5.3.2 Problems of Interest

A subset C of $\mathbb{R}^d$ is **convex** if $\lambda x + (1-\lambda)y \in C$ for any $x, y \in C$ and $\lambda \in [0,1]$. A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is **convex** if its domain $\mathrm{dom}(f)$ is convex and for any $\lambda \in [0,1]$,

$$f\left(\lambda x + (1-\lambda)y\right) \;\le\; \lambda f(x) + (1-\lambda)f(y), \qquad \forall x, y \in \mathrm{dom}(f), \tag{5.38}$$

or equivalently if

$$f(y) \;\ge\; f(x) + \nabla f(x)^\top (y - x), \qquad \forall x, y \in \mathrm{dom}(f). \tag{5.39}$$

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is **strongly convex** if there exists $\mu > 0$ such that $f(x) - \mu\|x\|^2$ is convex, or equivalently if

$$f(y) \;\geq\; f(x) + \nabla f(x)^\top (y - x) + \mu\|y - x\|^2, \qquad \forall x, y \in \mathrm{dom}(f). \tag{5.40}$$

In our numerical experiments, we will use termination criteria of the form,

$$|f(x_k) - f(x_{k-1})| < \delta \quad \text{and} \quad \|\nabla f(x_k)\| < \delta, \tag{5.41}$$

for various values of the tolerance δ, and solve the following convex problems:

Problem 5.1. *Minimize the quartic polynomial*

$$f(x) = 1 + \left[(x-1)^\top \Sigma (x-1)\right]^2, \quad \textit{where } \Sigma_{ij} = 0.9^{|i-j|} \textit{ and } x \in \mathbb{R}^d. \tag{5.42}$$

This convex (not strongly convex) function achieves its global minimum at $x^* = (1, \ldots, 1)^\top$.

Problem 5.2. *Minimize the convex (not strongly convex) function*

$$f(x_1, x_2) = x_1 + x_2^2 - \ln(x_1 x_2), \tag{5.43}$$

which achieves its global minimum at $x^* = (1, \sqrt{2}/2)^\top$.

Problem 5.3. *Minimize the strongly convex function*

$$f(x_1, ..., x_d) = \sum_{k=1}^{d} x_k \log x_k. \tag{5.44}$$

This function, known as the negative entropy function, achieves its global minimum at $x^* = (e^{-1}, \ldots, e^{-1})^\top$.

Problem 5.4. *Minimize the ill-conditioned strongly convex function*

$$f(x_1, x_2, x_3) = 1 + 0.01x_1^2 + x_2^2 + 100x_3^2, \tag{5.45}$$

which achieves its global minimum at $x^* = (0, 0, 0)^\top$.

Problem 5.5 (**Linear Regression** or **Least Squares**)**.** *Given a matrix $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ and a vector $b \in \mathbb{R}^m$, consider the problem of finding a vector $x \in \mathbb{R}^n$ such that $\|Ax - b\|_2$ is minimized. The least squares problem has many applications in data-fitting and interpolation. It can be formulated as the minimization of*

$$f(x) = \frac{1}{2} x^\top A^\top A x - b^\top A x, \tag{5.46}$$

with gradient given by $\nabla f(x) = A^\top A x - A^\top b$. A vector $x \in \mathbb{R}^n$ is a solution of the least squares problem if and only if it satisfies the normal equation $A^\top A x = A^\top b$. Furthermore, the least squares problem has a unique solution, given by $x^* = (A^\top A)^{-1} A^\top b$, if and only if the matrix A has full rank [Trefethen and Bau, 1997].

There are also regularized versions of the least squares problem or linear regression [Boyd and Vandenberghe, 2004; Bertsekas, 2009], to penalize larger values of the vector x. A common form of regularization is Tikhonov regularization [Phillips, 1962; Tikhonov and Arsenin, 1977] (or ℓ^2 regularization), where we minimize the convex function

$$f(x) = \|Ax - b\|_2^2 + \lambda \|x\|_2^2, \tag{5.47}$$

for some $\lambda > 0$, which has a unique minimizer $x^* = (A^\top A + \lambda b)^{-1} A^\top b$.

Another regularized version is the ℓ^1 penalized linear regression (also known as the Lasso problem [Tibshirani, 1996]), where we minimize the convex (not strongly convex) function

$$f(x) = \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1. \tag{5.48}$$

Problem 5.6 (**Logistic Regression** for Binary Classification)**.** *Given a set of feature vectors $x_1, \ldots, x_m \in \mathbb{R}^n$ and associated labels $y_1, \ldots, y_m \in \{-1, 1\}$, we want to find a vector $w \in \mathbb{R}^n$ such that $\operatorname{sign}(w^\top x)$ is a good model for $y(x)$. This can be formulated as the problem of minimizing the convex (not strongly convex) function*

$$f(w) = \sum_{i=1}^{m} \log\left(1 + \exp\left(-y_i w^\top x_i\right)\right). \tag{5.49}$$

As for linear regression, there are also regularized versions of logistic regression, such as ℓ^1 and ℓ^2 regularized logistic regression:

$$f(w) = \sum_{i=1}^{m} \log\left(1 + \exp\left(-y_i w^\top x_i\right)\right) + \lambda \|x\|_1, \tag{5.50}$$

$$f(w) = \sum_{i=1}^{m} \log\left(1 + \exp\left(-y_i w^\top x_i\right)\right) + \lambda \|x\|_2^2. \tag{5.51}$$

Problem 5.7 (**Fermat–Weber Location Problem** [Boltyanski et al., 1999; Drezner and Hamacher, 2002; Beck and Teboulle, 2009])**.** *Given a set of points $y_1, \ldots, y_m \in \mathbb{R}^n$ and associated positive weights $w_1, \ldots, w_m \in \mathbb{R}$, we want to find the location $x \in \mathbb{R}^n$ whose sum of weighted distances from the points $y_1, \ldots, y_m$ is minimized. In other words, we wish to minimize the convex function*

$$f(x) = \sum_{j=1}^{m} w_j \|x - y_j\|. \tag{5.52}$$

The Fermat–Weber location problem is at the heart of Location Theory and has countless applications across many fields of science and engineering.

Remark 5.2. *A Tikhonov-type regularization can also be achieved by modifying the second-order differential equation instead of adding a penalty to the objective function (see [Jendoubi and May, 2010; Attouch and Czarnecki, 2017; Attouch and Chbani, 2018; Alecsa and László, 2021] for instance). The idea is to add an extra term $\epsilon(t)x(t)$ with $\epsilon(t) \to 0$ as $t \to \infty$ to the second-order differential equation of interest:*

$$\ddot{x}(t) + \alpha(t)\dot{x}(t) + \gamma(t)\nabla f(x(t)) + \epsilon(t)x(t) = 0. \tag{5.53}$$

This extra term forces the generated trajectory to converge to a solution of minimal norm. This type of modified differential equation can be generated from a variational framework via Lagrangians and Hamiltonians of the form

$$L(x, v, t) = \frac{1}{2}\alpha(t)\langle v, v\rangle + \epsilon(t)\langle v, x\rangle - \gamma(t)f(x), \tag{5.54}$$

$$H(x, p, t) = \frac{1}{2\alpha(t)}\langle p - \epsilon(t)x, p - \epsilon(t)x\rangle + \gamma(t)f(x), \tag{5.55}$$

whose Euler–Lagrange equation reads

$$\alpha(t)\ddot{x}(t) + \dot{\alpha}(t)\dot{x}(t) + \gamma(t)\nabla f(x(t)) + \dot{\epsilon}(t)x(t) = 0. \tag{5.56}$$

5.4 Controlling the Oscillatory Behavior

The Bregman Euler–Lagrange equation (5.7) can be written in the form

$$\ddot{x}(t) + d(t)\dot{x}(t) + b(t)\nabla f(x(t)) = 0. \tag{5.57}$$

The introduction of momentum in the Bregman dynamical system causes the solution to this ordinary differential equation to overshoot frequently in its path towards the minimizer of the objective function f, and as a result the continuous-time solution can be highly oscillatory. Therefore, this differential equation can be thought of as modeling a nonlinear oscillator with damping, and the convergence of the function f to its minimum value is not monotone along Bregman trajectories. This is similar to what was observed for the limiting continuous differential equation for Nesterov's accelerated gradient method [Su et al., 2016; Muehlebach and Jordan, 2019] and for most momentum methods. These oscillations are problematic since they can significantly slow down optimization algorithms that are derived from the discretization of these Bregman differential equations. Indeed, to resolve the fast oscillations of the differential equation, the time-step in the discretization has to be reduced sufficiently, which can considerably increase the number of iterations and gradient evaluations needed to achieve convergence. If the time-step is not taken small enough, the momentum in the algorithms can lead to large overshoots which can result in divergence. It would therefore be desirable to have a mechanism to neutralize these oscillations. Fortunately, there are ways to reduce the effect of these oscillations, which we will discuss in the remainder of this section.

5.4.1 Momentum Restarting

Momentum in the optimization algorithms causes the solution to the Bregman Euler–Lagrange equation to overshoot frequently on its path towards the minimizer of f. One strategy to control these overshoots and reduce the effect of the resulting oscillations is to use restarting or **momentum restarting** schemes, previously explored in [Powell, 1977; Dai et al., 2004; O'donoghue and Candès, 2015; Giselsson and Boyd, 2015; Su et al., 2016; Fercoq and Qu, 2016; Donghwan and Fessler, 2018; Fercoq and Qu, 2019; Roulet and d'Aspremont, 2020; Renegar and Grimmer, 2022].

We will consider three different momentum restarting schemes:

- **Function Scheme**: Restart momentum whenever

$$f(q_k) > f(q_{k-1}),$$

that is, whenever the function evaluation at the new update moves away from its minimum value, to try to avoid wasting iterations in a bad direction.

- **Gradient Scheme**: Restart momentum whenever

$$\nabla f(q_k)(q_k - q_{k-1}) > 0$$

that is, whenever momentum seems to take the new updates in a bad direction, measured using the gradient at that point.

- **Velocity Scheme**: Restart momentum whenever

$$\|q_{k+1} - q_k\| < \|q_k - q_{k-1}\|$$

that is, whenever the norm of the (discrete version of the) velocity $\|\dot{q}\|$ starts decreasing, to try to maintain a high velocity along the trajectory.

Note that the quantities needed to implement these restarting schemes are already calculated in the standard versions of the optimization algorithms, and thus there is a negligible difference in the computational costs of each iteration in the restarted and non-restarted schemes. We can also require a minimum number of iterations between momentum restarts, as in [Su et al., 2016], to avoid having consecutive restarts that are too close to each other. In practice however, it did not seem to really improve the computational efficiency of the algorithm and could sometimes negatively impact the overall performance. For simplicity, we will not impose a minimum number of iterations between consecutive restarts.

In our first numerical experiment, we compared the performance of the standard algorithms to their restarted versions on three different problems for fixed values of all the parameters except the time-step h. Figure 5.1 shows the resulting error plots after tuning the value of h optimally.

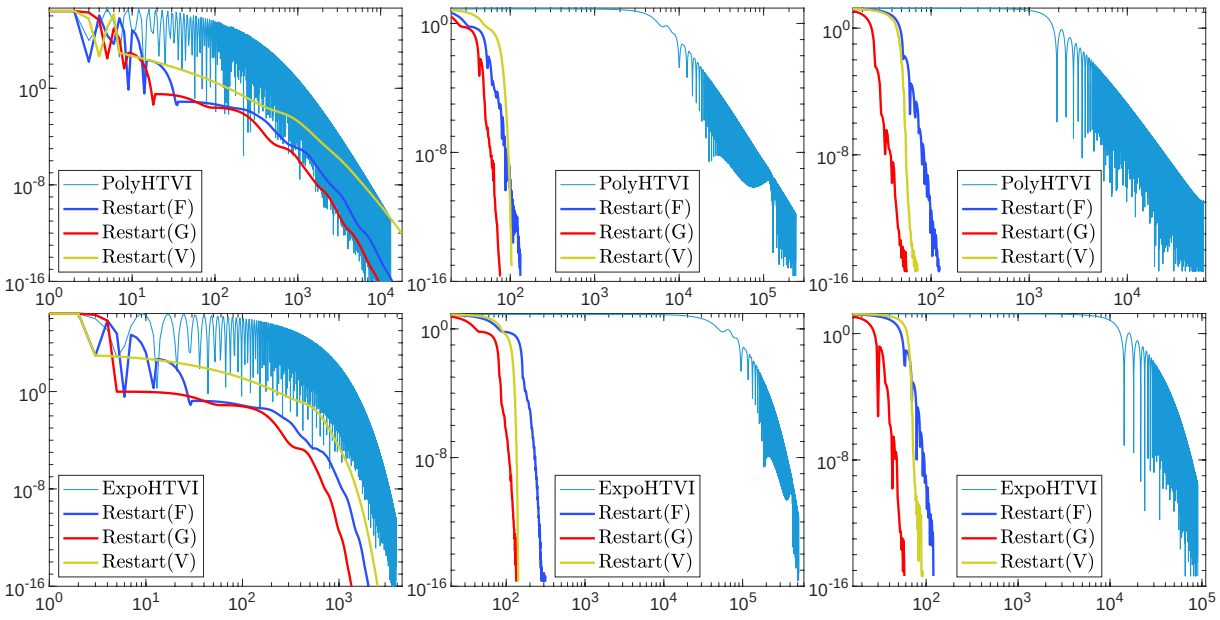


Figure 5.1: Error vs. Iterations number as the standard PolyHTVI and ExpoHTVI algorithms and their restarted versions (Function (F), Gradient (G) and Velocity (V)) are applied to Problem 5.1 (left), Problem 5.2 (middle), and Problem 5.3 (right).

We can clearly see that the restarted versions of the algorithms are much less oscillatory, and as a result they can sometimes allow for much larger time-steps leading to significantly faster algorithms, as is the case for Problems 5.2 and 5.3. Note however that Problem 5.1 is a special instance where larger time-steps cannot be taken in the algorithms with a momentum restarting scheme, despite their non-oscillatory nature. It should be noted nevertheless that although the use of momentum restarting may not always lead to significant improvements in computational efficiency, it does not penalize computational efficiency either.

We have performed additional experiments to obtain a better idea of the benefits of momentum restarting in terms of computational efficiency, robustness and stability. More precisely, we solved optimization problems using the different versions of the algorithms on a 100×100 grid with logarithmic spacing in the parameter (C, h)-plane, and recorded the number of iterations required to achieve certain convergence criteria. Figures 5.2, 5.3, 5.4, and 5.5 display the results as filled contour plots (where the absence of color indicates either divergence or failure of the algorithm to converge in less than 10^6 iterations). Table 5.1 displays the number of iterations required to converge by each version of the algorithms with its optimal (C, h) pair on the 100×100 logarithmically-spaced grid.

Figure 5.2 confirms the earlier observations that restarting can significantly reduce the number of iterations needed to converge, and we can also see that the restarted versions of the algorithm are more robust, since the regions of fast convergence are larger than for the standard algorithm. As a result, it is easier to tune the restarted algorithms to achieve fast convergence. Note as well from Figure 5.3 that a restarting scheme can significantly improve the stability of the algorithms. Indeed, we can that as the convergence criteria are made stricter going from Figure 5.2 to Figure 5.3, the regions of fast convergence have not shrunk as dramatically for the restarted algorithms as for the standard version. Given a converging (C, h) pair for a restarted algorithm in Figure 5.2, the restarted algorithm usually remains convergent for that (C, h) pair with the stricter criteria in Figure 5.2 with a slightly increased number of iterations required. This is not true for the standard algorithm where the increase in number of iterations is much more significant, and there is a larger region of initially convergent (C, h) pairs where the standard algorithm diverges when the stricter convergence criteria is imposed.

As observed earlier in Figure 5.1, we see from Figure 5.4 that momentum restarting does not lead to significant improvements in computational efficiency for Problem 5.1, but also does not penalize computational efficiency in that case. From Figure 5.4, we see that this observation extends to robustness and stability. since all the different versions of the algorithm share similar convergence regions given the same parameter values and convergence criteria.

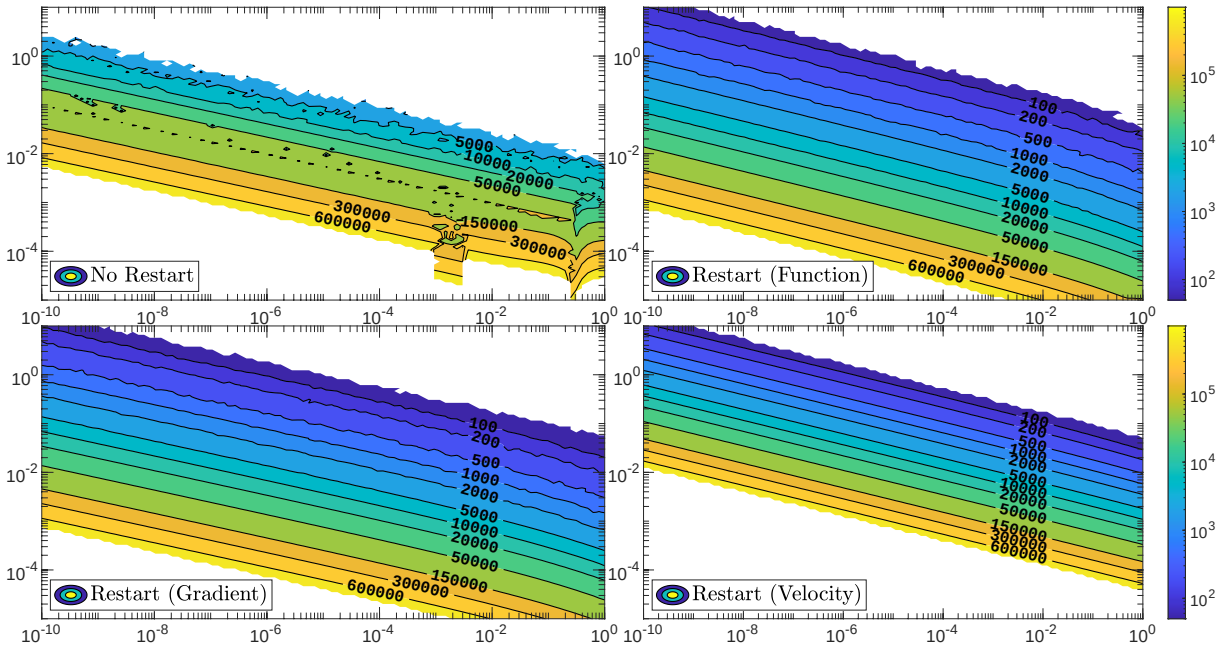


Figure 5.2: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-5}$ in the (C, h)-plane, for $p = \mathring{p} = 4$ PolyHTVI applied to Problem 5.2.

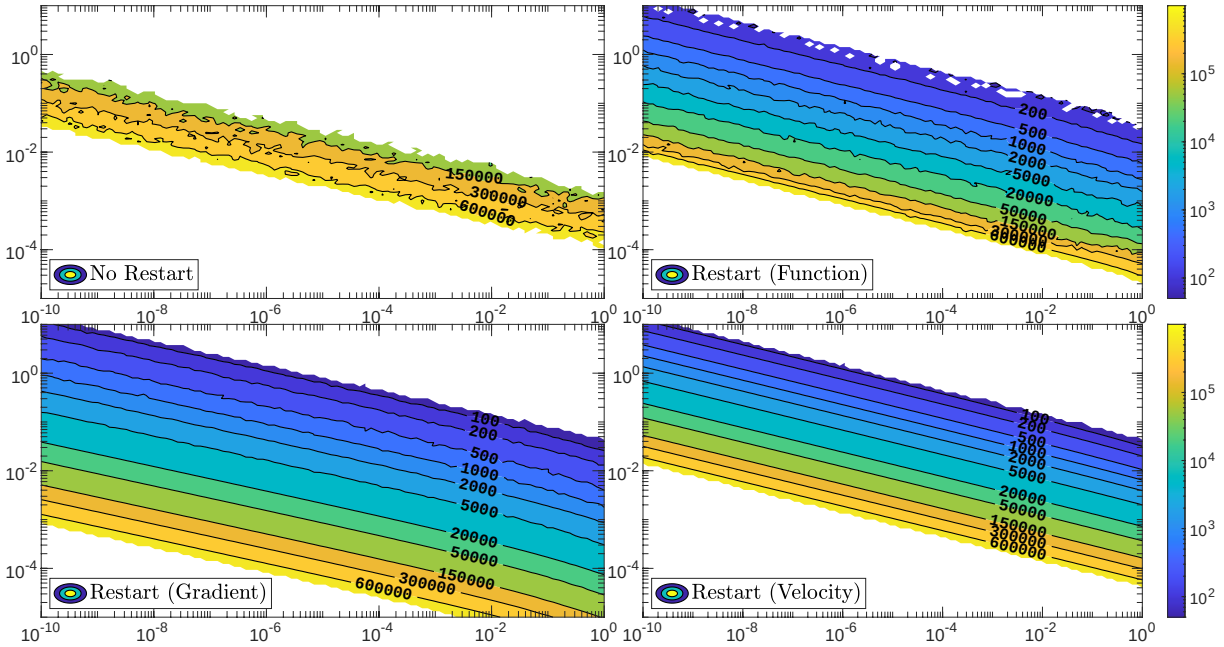


Figure 5.3: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-8}$ in the (C, h)-plane, for $p = \mathring{p} = 4$ PolyHTVI applied to Problem 5.2.

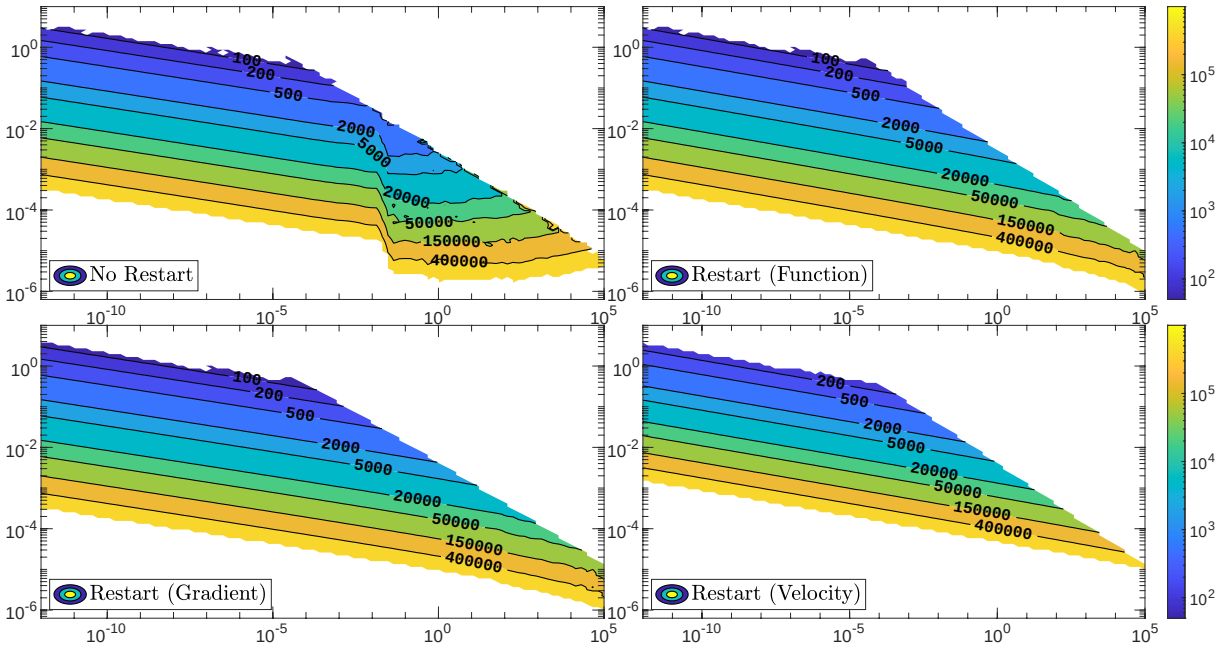


Figure 5.4: Contour plot of the number of iterations required to achieve convergence in the (C, h)-plane, for the $p = \mathring{p} = 8$ PolyHTVI algorithm applied to Problem 5.1.

All the observations made so far also extend to the other families of Bregman dynamics and other algorithms, as can be seen in Figure 5.5 for the ExpoSLC algorithm for instance, where momentum restarting leads to significant gains in computational efficiency, robustness and stability for Problem 5.2. Table 5.1 provides some additional data supporting the significant gain in efficiency that can be achieved using momentum restarting.

Overall, all the numerical experiments conducted in this section unequivocally support the use of momentum restarting in the algorithms for accelerated optimization, and it can be seen from Figures 5.2, 5.3, 5.4, and 5.5 and Table 5.1 that the gradient-based restarting scheme consistently outperforms the other two restarting schemes in terms of computational efficiency, robustness and stability. Unless stated otherwise, we will now always use momentum restarting based on the gradient scheme in the remainder of this chapter, without explicitly stating it every time.

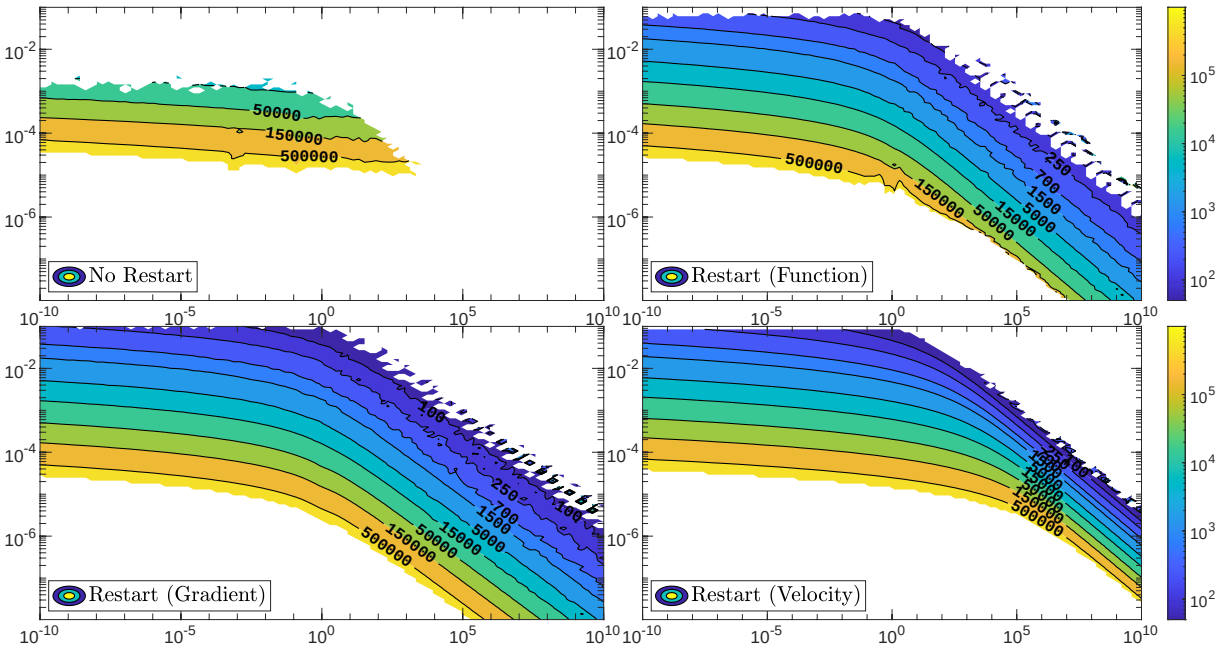


Figure 5.5: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-5}$ in the (C, h)-plane, for $\eta = \mathring{\eta} = 1$ ExpoSLC applied to Problem 5.2.

Table 5.1: Comparison of the fastest convergence achieved by the standard algorithms and the restarting schemes on various problems with different tolerances δ (displayed in terms of number of iterations required to achieve the termination criteria).

Algorithm	**Problem**	δ	**No Restart**	**Function Scheme**	**Gradient Scheme**	**Velocity Scheme**
PolyHTVI	Problem 5.1	10^{-12}	52	52	52	108
PolyHTVI	Problem 5.2	10^{-5}	621	39	23	34
PolyHTVI	Problem 5.2	10^{-8}	15994	80	51	57
PolyHTVI	Problem 5.3	10^{-8}	4121	60	11	16
PolyHTVI	Problem 5.4	10^{-8}	14723	60	12	14
PolyHTVI	Problem 5.5	10^{-5}	3917	104	33	38
ExpoSLC	Problem 5.1	10^{-12}	75	68	64	155
ExpoSLC	Problem 5.2	10^{-5}	3929	50	20	21
ExpoSLC	Problem 5.2	10^{-8}	204598	91	27	32
ExpoSLC	Problem 5.3	10^{-8}	17081	66	15	18
ExpoSLC	Problem 5.4	10^{-8}	58028	72	10	12
ExpoSLC	Problem 5.5	10^{-5}	21440	54	27	41

5.4.2 The Effect of the Parameter C

The parameter C in the polynomial and exponential subfamilies of Bregman dynamics can sometimes provide a simple way to control the oscillatory behavior of the second-order differential equation. From the point of view of perturbation theory, the polynomial and exponential Bregman Euler–Lagrange equations (5.11) and (5.16),

$$\ddot{q}(t) + \frac{p+1}{t}\dot{q}(t) + Cp^2 t^{p-2}\nabla f(q(t)) = 0, \quad \text{and} \quad \ddot{q}(t) + \eta\dot{q} + C\eta^2 e^{\eta t}\nabla f(q(t)) = 0, \tag{5.58}$$

can be thought of as perturbations of the simpler differential equations,

$$\ddot{u}(t) + \frac{p+1}{t}\dot{u}(t) = 0, \qquad \text{and} \qquad \ddot{v}(t) + \eta\dot{v} = 0. \tag{5.59}$$

The solutions to these two unperturbed equations are given by

$$u(t) = \left(k_1 t^{-p} + k_2\right)\mathbf{1}, \qquad \text{and} \qquad v(t) = \left(k_3 e^{-\eta t} + k_4\right)\mathbf{1}, \tag{5.60}$$

for some constants k_1, k_2, k_3, k_4 depending on the initial conditions. These unperturbed solutions are non-oscillatory, and converge monotonically to a constant vector at the respective rates of $\mathcal{O}(t^{-p})$ and $\mathcal{O}(e^{-\eta t})$. We can thus think of the terms $Cp^2t^{p-2}\nabla f(q(t))$ and $C\eta^2 e^{\eta t}\nabla f(q(t))$ as perturbations steering the dynamical system towards the minimizer of the objective function f, in an oscillatory fashion. The parameter C, which appears in front of these two perturbation terms, should therefore be chosen, in theory, to be small enough to control the oscillations but also large enough to guide the dynamical system towards the minimizer of the objective function. The situation is similar in the ExpoToPoly and PolyToExpo subfamilies of Bregman dynamics.

This perturbation theoretic point of view and the numerical results which will be presented in this section suggest that the parameter C can play an important role reducing the effect of oscillations and improving the performance of optimization algorithms. The role of C has not been sufficiently explored so far in this dissertation and in the literature exploiting the variational framework for accelerated optimization (in [Wibisono et al., 2016; Jordan, 2018; Betancourt et al., 2018; Campos et al., 2021] for instance), and the resulting dynamical systems were highly-oscillatory and thus required smaller time-steps for their discretizations. As a consequence, the resulting optimization algorithms might not be as competitive as they could have been. Note as well that the limiting continuous differential

equation for Nesterov's Accelerated Gradient introduced in [Su et al., 2016] can be thought of as the $p = 2$ polynomial Bregman dynamics with $C = 1/4$, which results in the highly oscillatory behavior observed in the continuous dynamics associated to most objective functions, and in the numerous discretizations of these dynamics which can be found in the literature. This observation also extends to the Riemannian manifold generalization of this variational framework for accelerated optimization [Duruisseaux and Leok, 2022d], where the constant C might not have been optimally tuned in practice (in Chapter 4 and in [Tao and Ohsawa, 2020; Lee et al., 2021] for instance).

As a first example, Figures 5.6 and 5.7 display the changes in the polynomial and exponential Bregman dynamics for Problem 5.1 as the parameter C is decreased. The oscillations are clearly neutralized in the continuous solutions as C decreases. Although the convergence happens later in time for lower values of C, this is usually not an issue since the neutralization of the oscillations allows for larger time-steps when discretizing, as can be seen in Figure 5.8. This could also be seen for instance in the 'No Restart' contour plots presented in Figures 5.2, 5.3, 5.4, and 5.5, where lower values of C allowed for larger time-steps h. Unfortunately, this behavior as the value of C is decreased does not seem to be universal, as can be seen from Figures 5.9 and 5.10 for Problem 5.4.

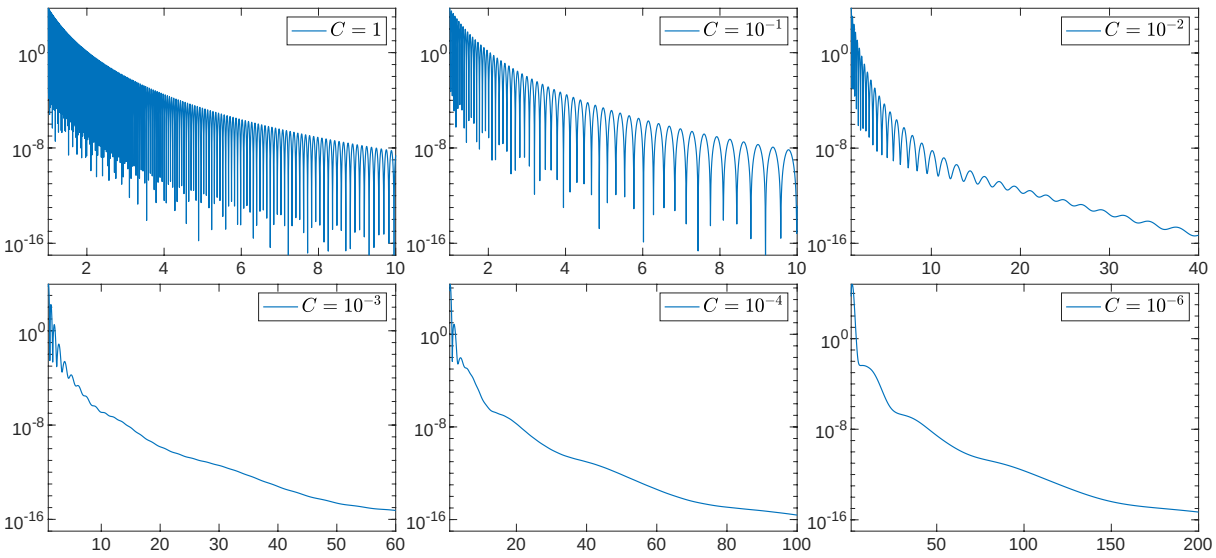


Figure 5.6: Error as a function of time t along the $p = \mathring{p} = 6$ polynomial Bregman dynamics for Problem 5.1, with different values of the constant C.

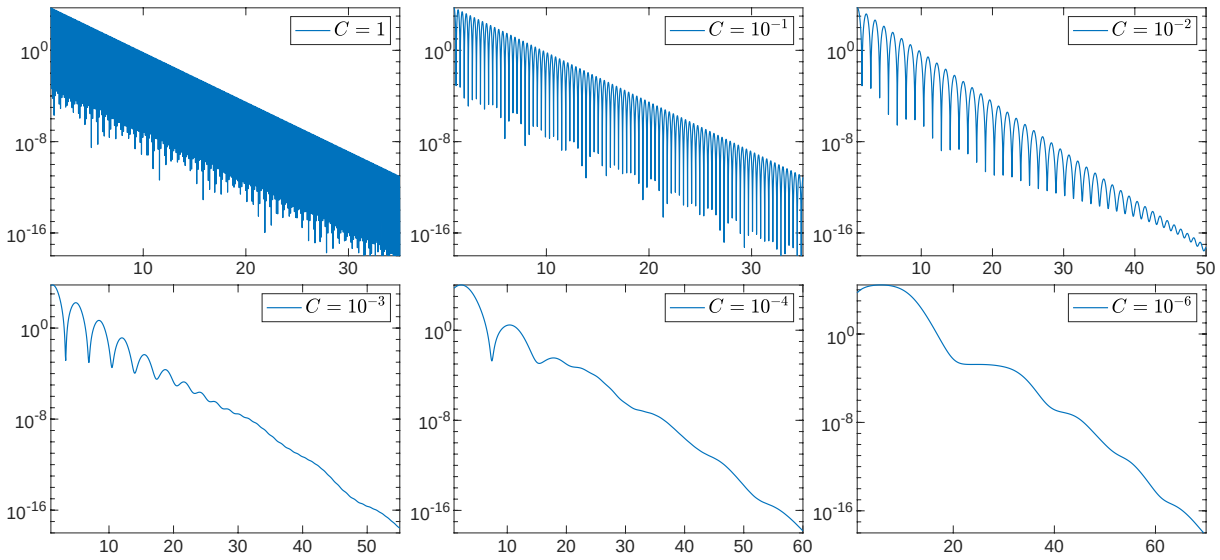


Figure 5.7: Error as a function of time t along the $\eta = \mathring{\eta} = 0.5$ exponential Bregman dynamics for Problem 5.1, with different values of the constant C.

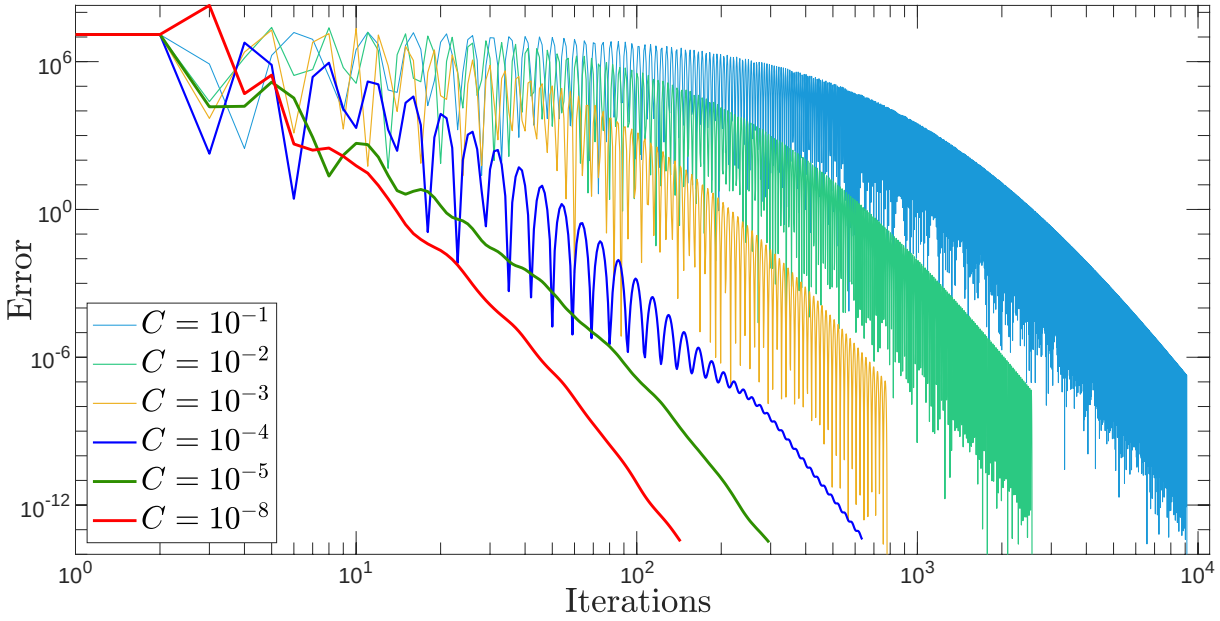


Figure 5.8: Discretization of the polynomial Bregman dynamics using PolyHTVI with different values of C for Problem 5.1 (without momentum restarting).

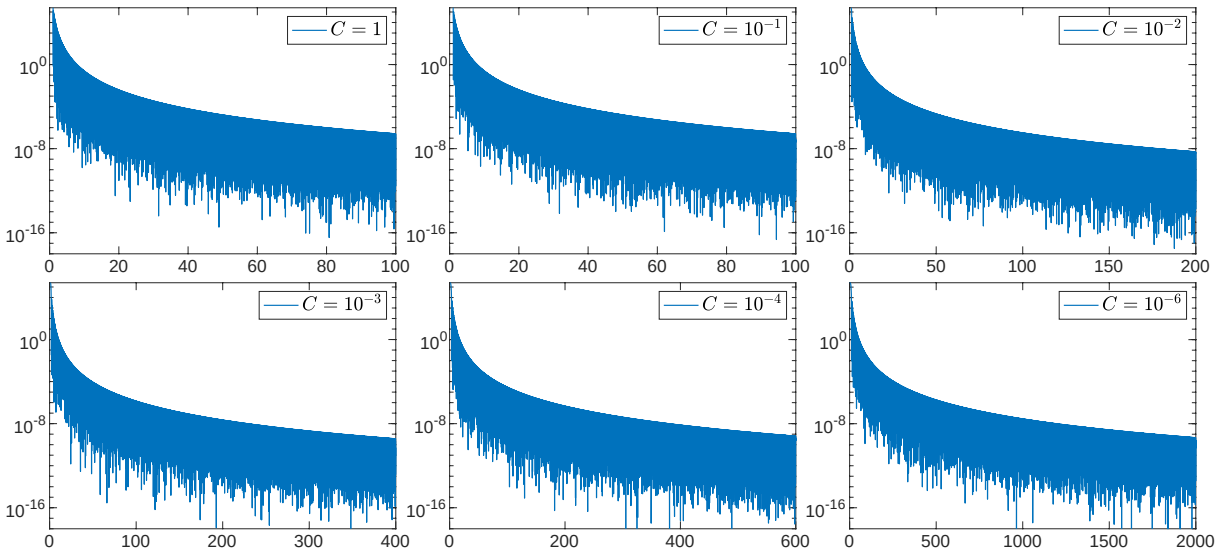


Figure 5.9: Error as a function of time t along the $p = \mathring{p} = 6$ polynomial Bregman dynamics for Problem 5.4, with different values of the constant C.

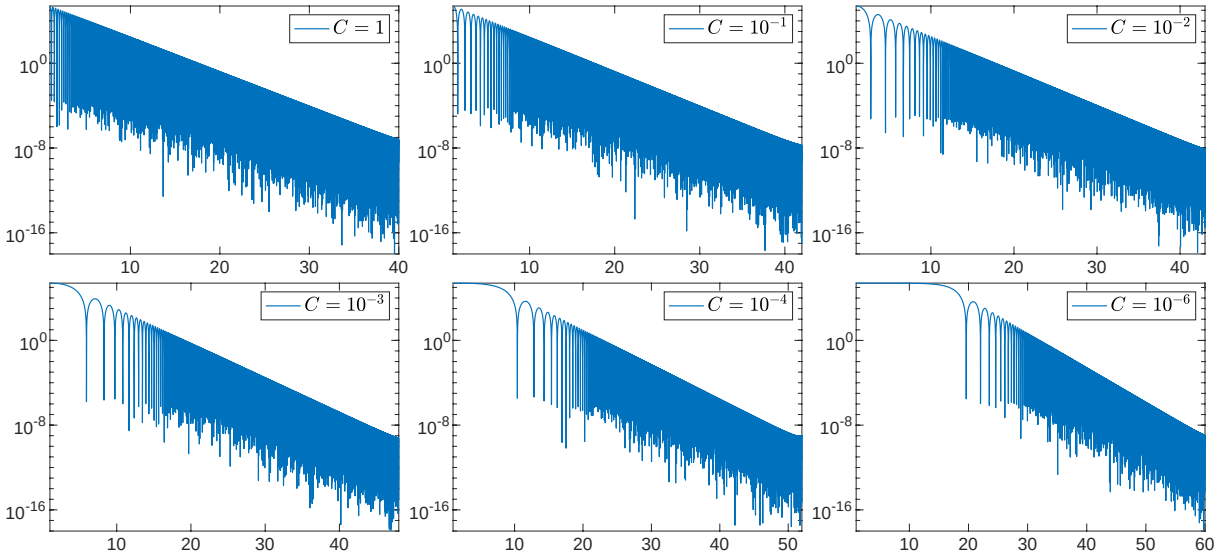


Figure 5.10: Error as a function of time t along the $\eta = \mathring{\eta} = 0.5$ exponential Bregman dynamics for Problem 5.4, with different values of the constant C.

We will now try to obtain a better understanding of the dependence on C of the computational efficiency of the optimization algorithms, and of how a good choice of parameter C depends on the other variables.

Preliminary experiments showed that the convergence regions are very similar for the different algorithms within the same adaptive family of Bregman dynamics, so we will only use the HTVI algorithms here but the results extend to the other families of algorithms. We will return to the comparison of the different geometric integrators later in Section 5.6.

Let us first investigate whether the regions of optimal convergence to the minimizer in the (C, h)-plane are problem-dependent. Figures 5.11 and 5.12 display the convergence regions obtained when using the HTVI algorithm for the polynomial and exponential Bregman dynamics on four different objective functions. Unfortunately, these numerical results show that the regions of optimal convergence are problem-dependent and as a result we will not be able to find a single value of C which will achieve almost-optimal performance on all problems.

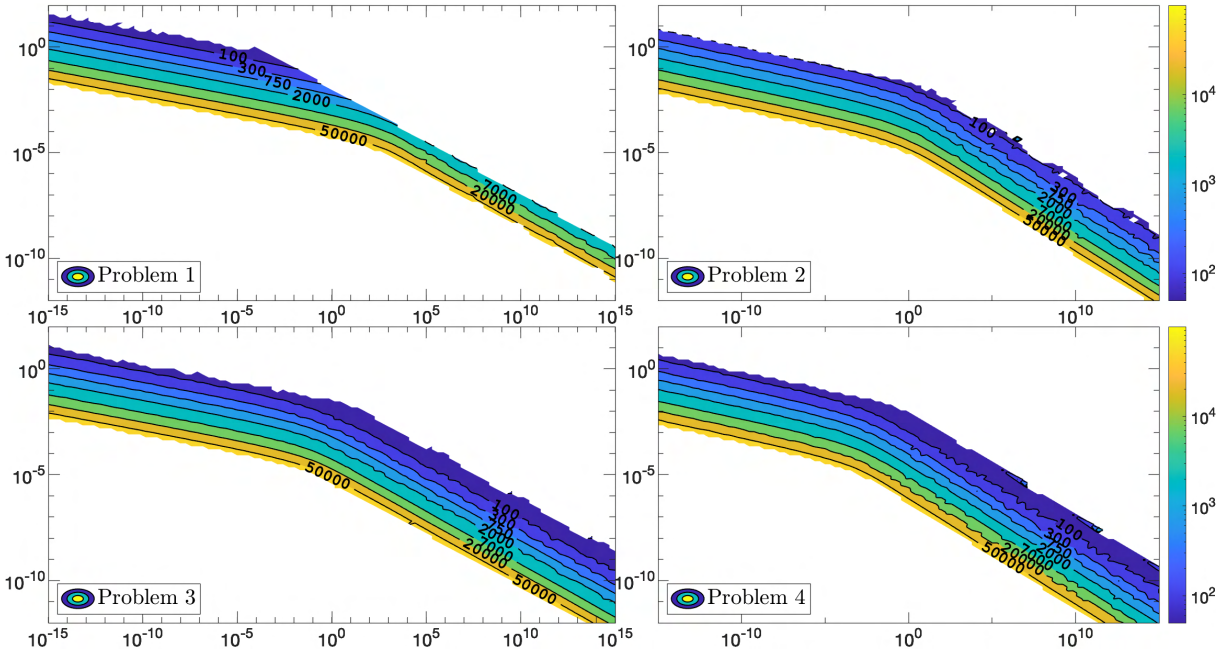


Figure 5.11: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-6}$ in the (C, h)-plane, for the $p = \mathring{p} = 6$ PolyHTVI algorithm applied to Problems 5.1, 5.2, 5.3, 5.4.

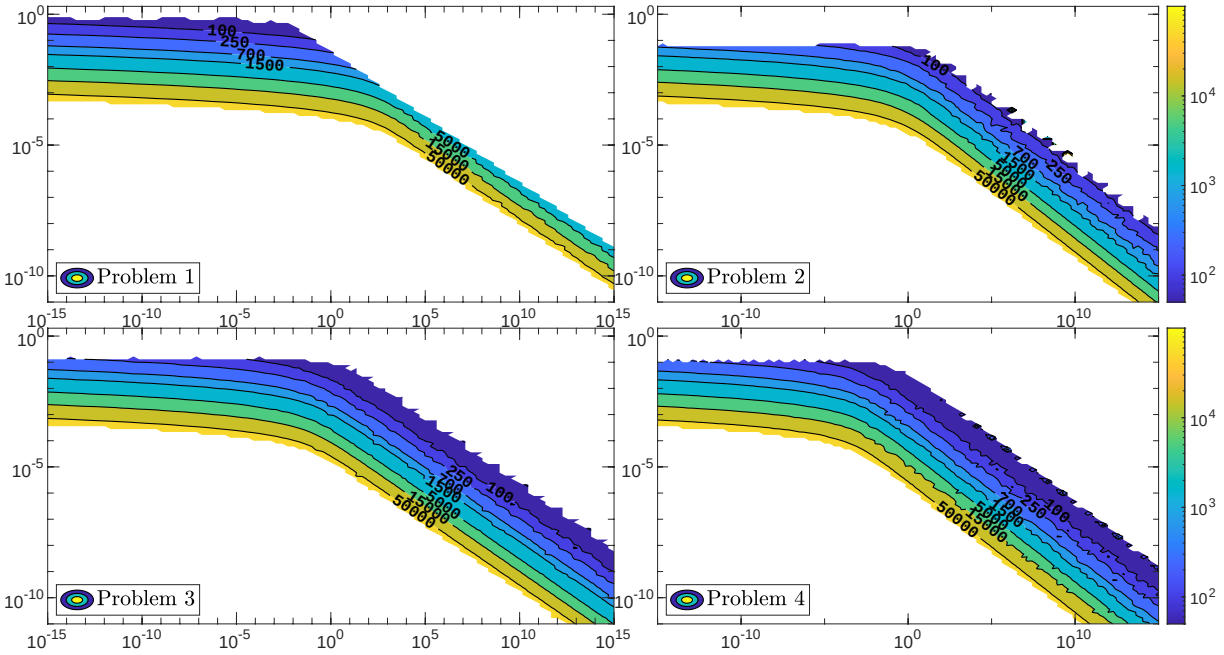


Figure 5.12: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-6}$ in the (C, h)-plane, for the $\eta = \mathring{\eta} = 6$ ExpoHTVI algorithm applied to Problems 5.1, 5.2, 5.3, 5.4.

However, from Figures 5.13, 5.14 and 5.15, we can see that for fixed values of $p, \mathring{p}, \eta, \mathring{\eta}$, the convergence regions in the (C, h)-plane are left almost unchanged as the dimension of the problem is increased from $d = 3$ to $d = 100$, although the numbers of iterations required increase slightly with d. This observation can improve significantly the process of tuning the optimization algorithm for high-dimensional problems by first tuning the algorithm on a similar low-dimensional problem, which could be particularly helpful for certain machine learning applications.

Note that all the observations made in this section extend to the ExpoToPoly and PolyToExpo subfamilies of time-adaptive Bregman dynamics.

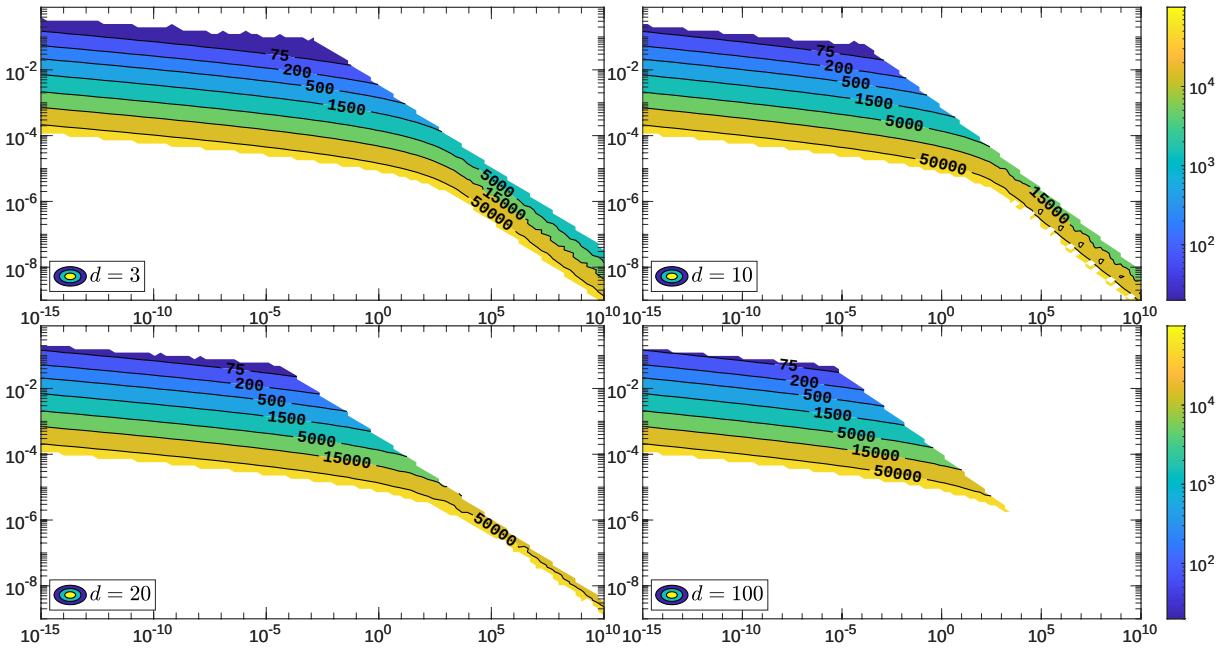


Figure 5.13: Number of iterations required to achieve $\delta = 10^{-6}$ convergence in the (C, h)-plane, for $p = 6, \mathring{p} = 2$ PolyHTVI applied to Problem 5.1 with different dimensions d.

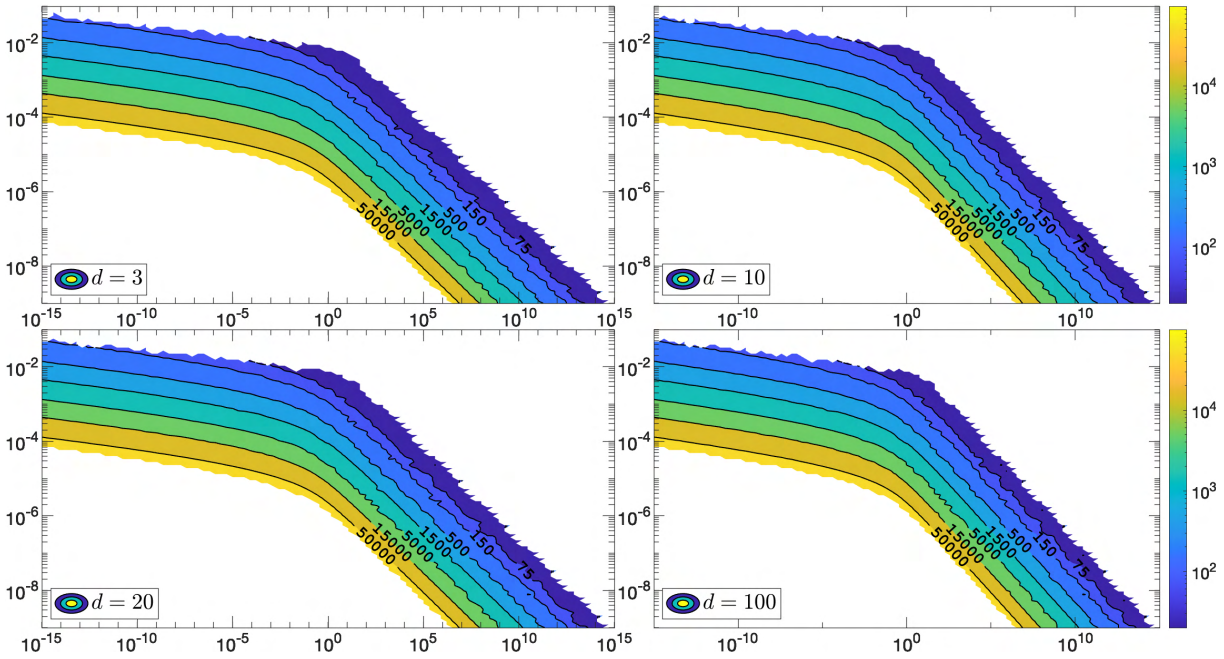


Figure 5.14: Number of iterations required to achieve $\delta = 10^{-6}$) convergence in the (C, h)-plane, for $p = 6, \mathring{p} = 2$ PolyHTVI applied to Problem 5.3 with different dimensions d.

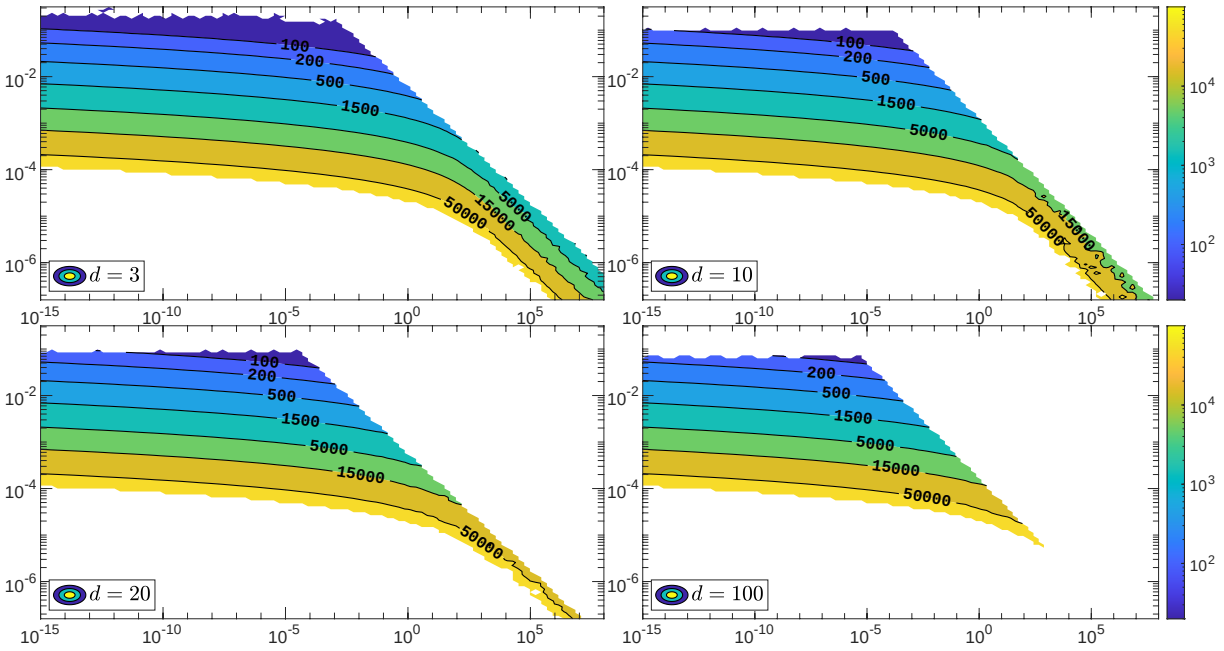


Figure 5.15: Contour plot of the number of iterations required to achieve convergence ($\delta = 10^{-6}$) in the (C, h)-plane, for the $\eta = 2, \mathring{\eta} = 1$ ExpoHTVI algorithm applied to Problem 5.1 with different dimensions d.

5.4.3 Other Approaches to Control Oscillations

There are other possible approaches to control the oscillations in second-order nonlinear differential equations. One such method is Hessian-driven damping [Alvarez et al., 2002; Attouch et al., 2020, 2021, 2022], where the idea is to add a damping term which involves the Hessian of the objective function, $\beta(t)\nabla^2 f(x(t))\dot{x}(t)$, to the differential equation of interest:

$$\ddot{x}(t) + \gamma(t)\dot{x}(t) + \beta(t)\nabla^2 f(x(t))\dot{x}(t) + b(t)\nabla f(x(t)) = 0. \tag{5.61}$$

The addition of this Hessian-driven damping term to the differential equation appears to neutralize the undesirable oscillations in the continuous solution to this second-order differential equation. Furthermore, it was shown using Lyapunov analysis that

under suitable assumptions, solutions to the modified equation not only achieved a similar convergence rate to the minimizer as the solutions to the original equation, but also benefited from additional convergence properties for the norm of the gradient ∇f. First-order optimization algorithms were also derived by discretizing the modified second-order differential equation, after rewriting $\nabla^2 f(x(t))\dot{x}(t)$ as $\frac{d}{dt}\nabla f(x(t))$. Unfortunately, we cannot derive a simple variational formulation for this modified second-order differential equation, so we cannot easily incorporate Hessian-driven damping into our framework which relies on geometric numerical integration of Lagrangian or Hamiltonian systems.

Another possible approach to control oscillations consists in simplifying the Bregman dynamics using local approximations. For instance, one could integrate local linearizations of Hamilton's equations, or use a local quadratic model for the objective function, or start from local quadratic Hamiltonian approximations to the Bregman Hamiltonian. We will not consider these methods here because they can suffer from additional numerical stability issues coming from the approximations at play, and it can be very challenging to design a symplectic integrator which preserves the nice properties of the dynamics across all the different local approximations.

A different approach consists in designing a symplectic integrator which can travel faster along the oscillations via larger time-steps. This may be achievable using Spectral or Galerkin variational integrators [Marsden and West, 2001; Leok and Zhang, 2011; Leok and Shingel, 2012b; Hall, 2015], which rely on a choice of basis functions that span a good approximation space for the Bregman dynamics (for instance, simulations of the polynomial Bregman dynamics suggest that the error usually follows a trajectory which can be well-approximated using functions of the form $t^{-\gamma}\cos(\alpha t^{\beta})$ or $t^{-\gamma}\sin(\alpha t^{\beta})$, where γ is the decay rate, α tunes the frequency of oscillations, and $\beta \in (0,1)$ characterizes the slowing down of the oscillation frequency). Due to the oscillatory nature of the dynamical system, it might also be advantageous to use Filon-type [Filon, 1930] or Levin-type [Levin, 1982, 1996] quadrature rules in the construction of the integrators, since they are designed specifically for highly oscillatory integrals (see [Deaño et al., 2018]).

Yet another possibility involves averaging techniques [Verhulst, 1996; Sanders et al., 2007; Farr, 2009; Schmitt and Leok, 2017]. The extended Bregman Hamiltonians or Lagrangians can be split as

$$H(\bar{q},\bar{r}) = H^A(\bar{q},\bar{r}) + CH^B(\bar{q}), \tag{5.62}$$

or

$$L(\bar{q},\bar{q}') = L^A(\bar{q},\bar{q}') + CL^B(\bar{q}), \tag{5.63}$$

where the A-component is dominating and can be solved exactly (or efficiently approximated with high accuracy), and the B-component generates small perturbations affecting the overall dynamics. One can then hope to integrate the dominating dynamics very accurately with larger time-steps and incorporate the influence of the small perturbations by averaging them out. Unfortunately, although this approach seemed to neutralize the oscillations in the solution in practice, it did not allow the use of larger time-steps, and the resulting algorithm was actually less competitive and robust because of the implicit nature of the update for the momentum r.

5.5 The Use of Time-Adaptivity in the Momentum Restarted Algorithms

We will first investigate how the optimization algorithms behave as the values of the parameters $\mathring{p}$ and $\mathring{\eta}$ are varied. In Chapters 3 and 4 and in [Duruisseaux et al., 2021; Duruisseaux and Leok, 2022a,c,d, 2023a], numerical experiments with the polynomial Bregman dynamics suggested that time-adaptivity (i.e. using $\mathring{p} \neq p$) could result in significantly faster optimization algorithms due to the exponentially growing time-steps that they use (instead of constant time-steps for $\mathring{p} = p$). These numerical experiments were however carried with the standard versions of the algorithms and without a careful tuning of the parameter C.

New numerical experiments carried in this chapter suggest that the introduction of momentum restarting schemes in the optimization algorithms enables significantly faster optimization and seems to remove the advantages of the time-adaptive formulation. Indeed, the contour plots in (C, h)-space presented in Figures 5.16, 5.17, 5.18, 5.19, and 5.20, 5.21 show that the performance and robustness of the PolyHTVI and ExpoHTVI algorithms with momentum restarting is almost unaffected by the introduction of time-adaptivity, regardless of which of Problems 5.1, 5.2, 5.3, 5.4 they are applied to. This is confirmed by the contour plots in $(\mathring{p}, h)$-space and $(\mathring{\eta}, h)$-space presented in Figures 5.22 and 5.23 where we can see that for fixed values of p or η, the value of $\mathring{p}$ or $\mathring{\eta}$ has very little effect on the performance of the optimization algorithms.

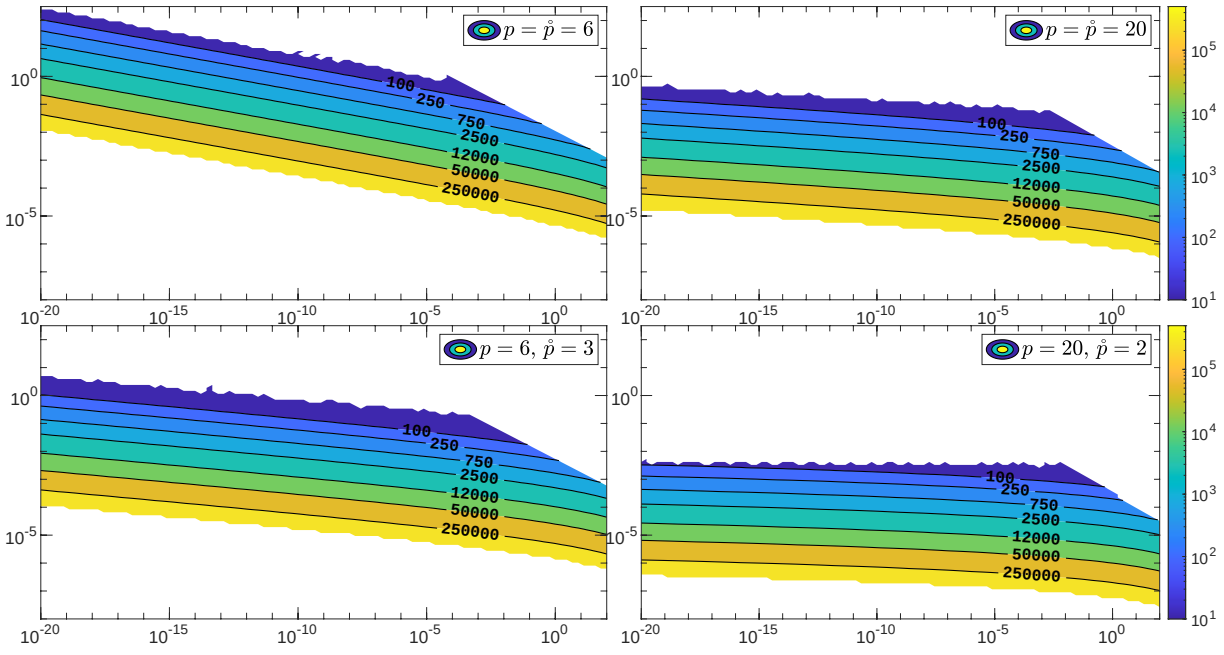


Figure 5.16: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-6}$ in the (C, h)-plane, for the PolyHTVI algorithm applied to Problem 5.1 with different values of p and $\mathring{p}$.

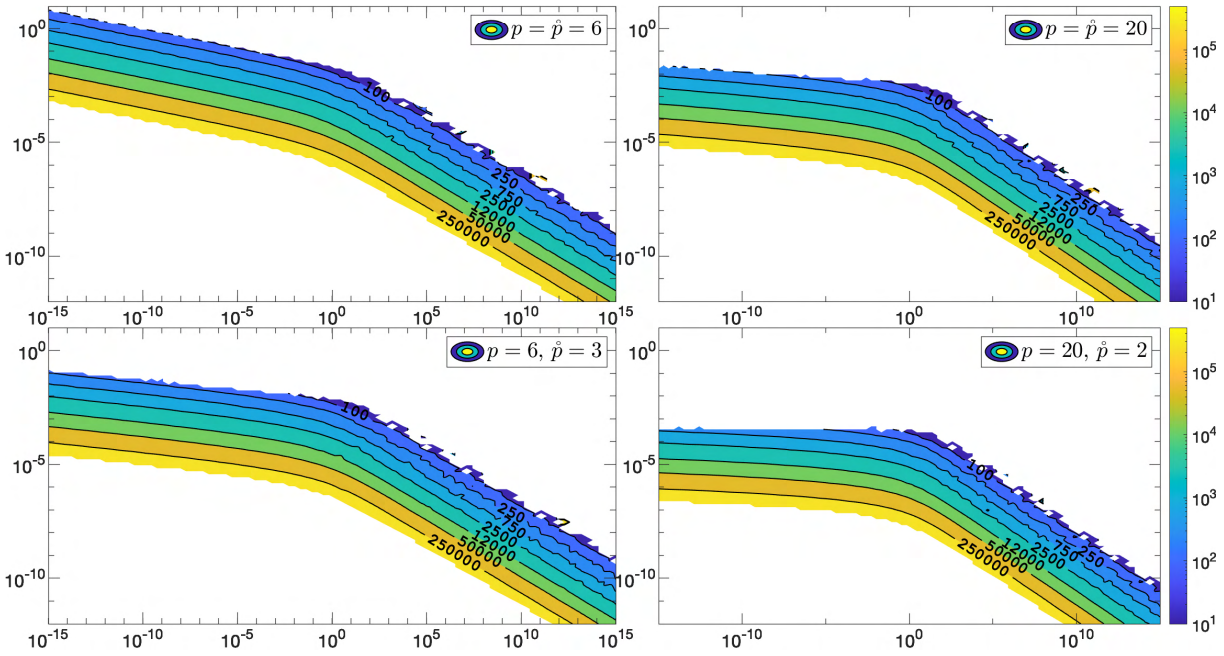


Figure 5.17: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-6}$ in the (C, h)-plane, for PolyHTVI appliedto Problem 5.2.

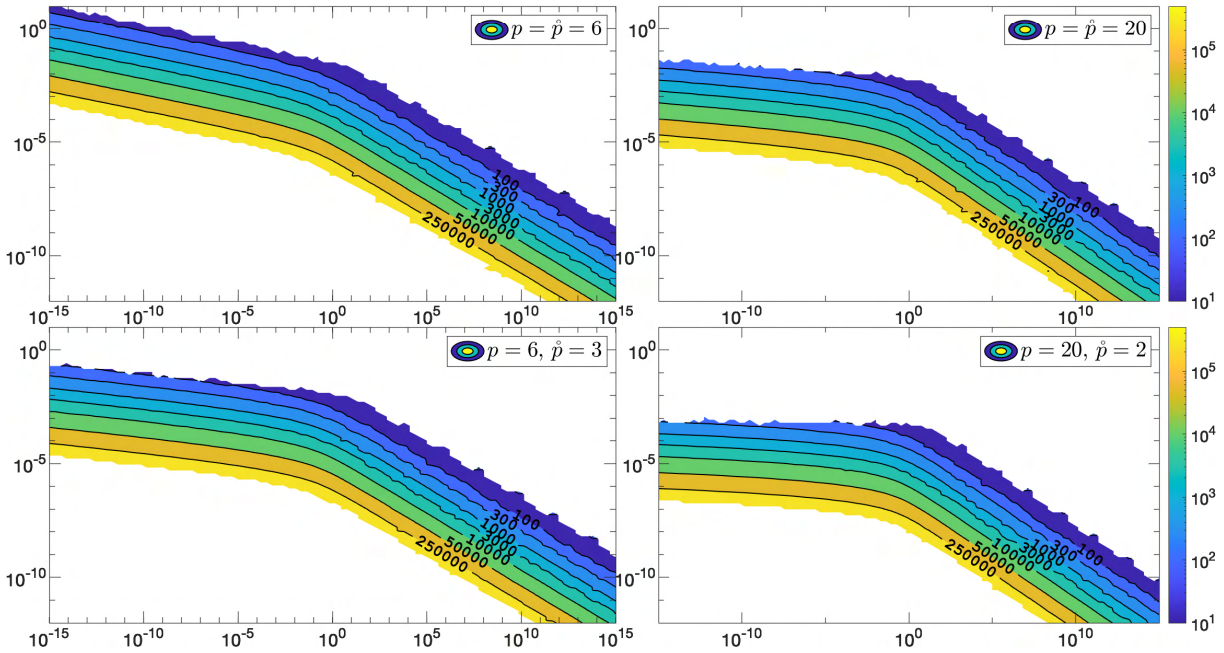


Figure 5.18: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-6}$ in the (C, h)-plane, for PolyHTVI applied to Problem 5.3.

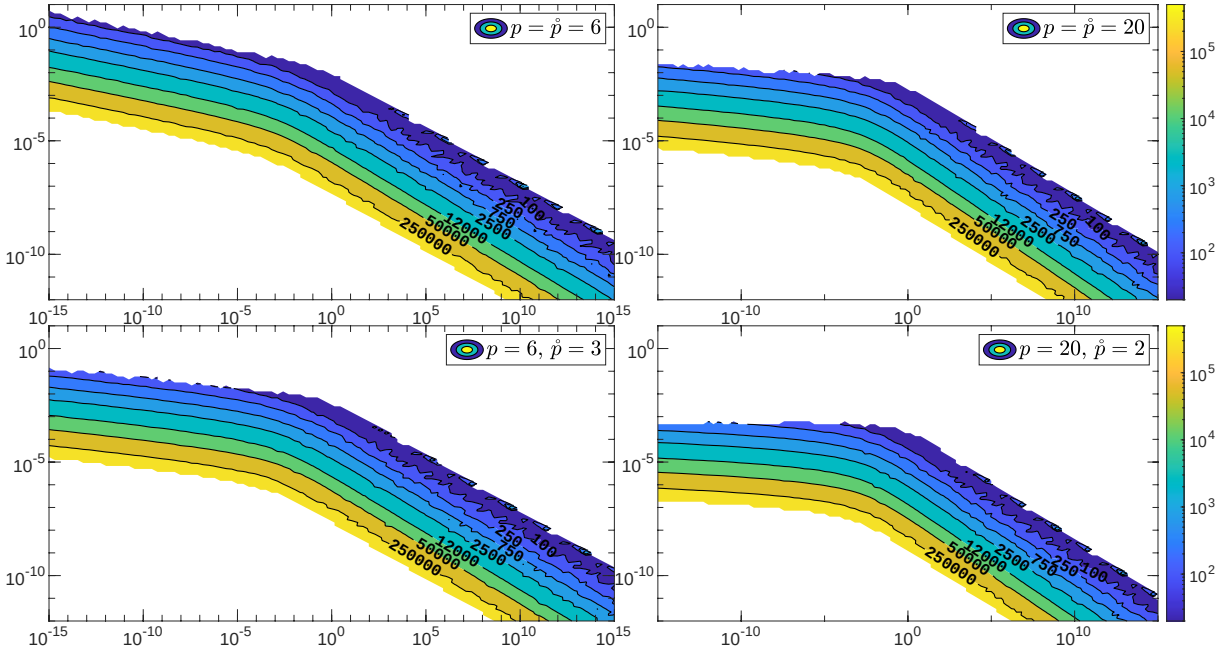


Figure 5.19: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-8}$ in the (C, h)-plane, for PolyHTVI applied to Problem 5.4.

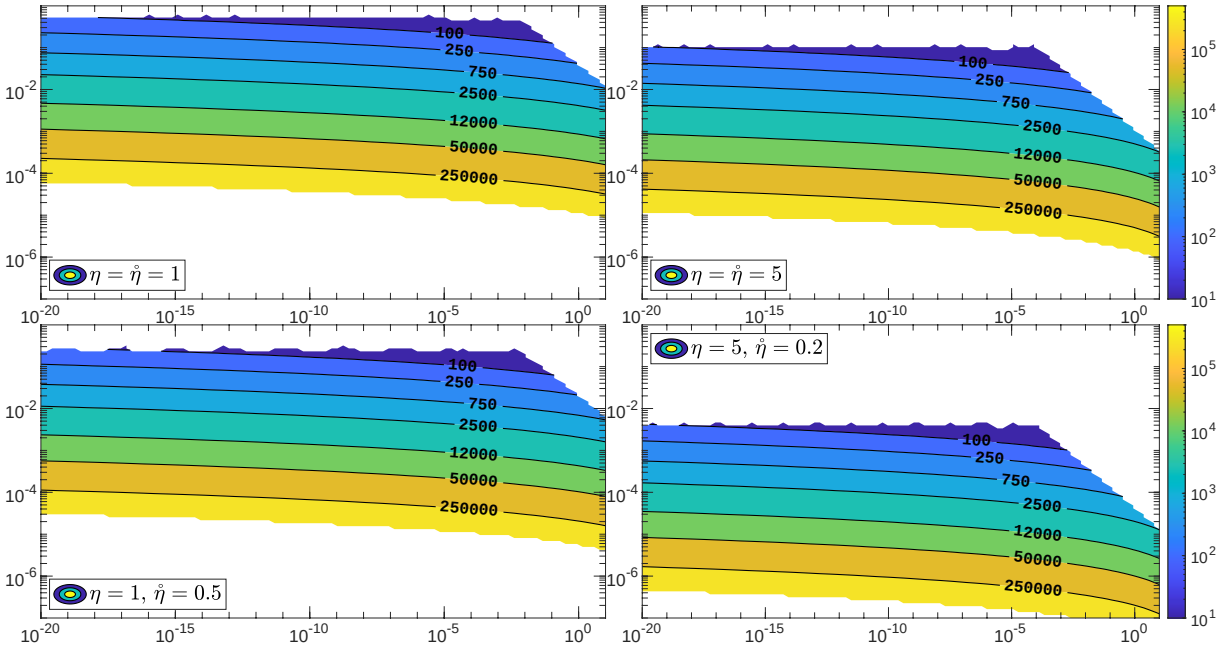


Figure 5.20: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-7}$ in the (C, h)-plane, for ExpoHTVI applied to Problem 5.1.

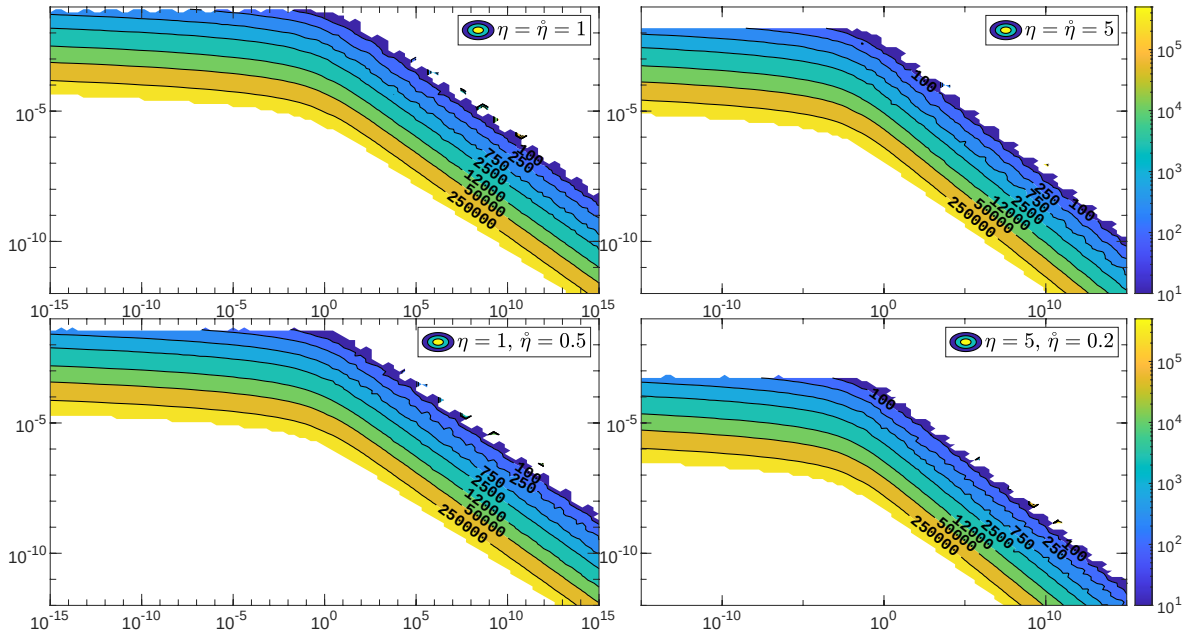


Figure 5.21: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-5}$ in the (C, h)-plane, for ExpoHTVI applied to Problem 5.2.

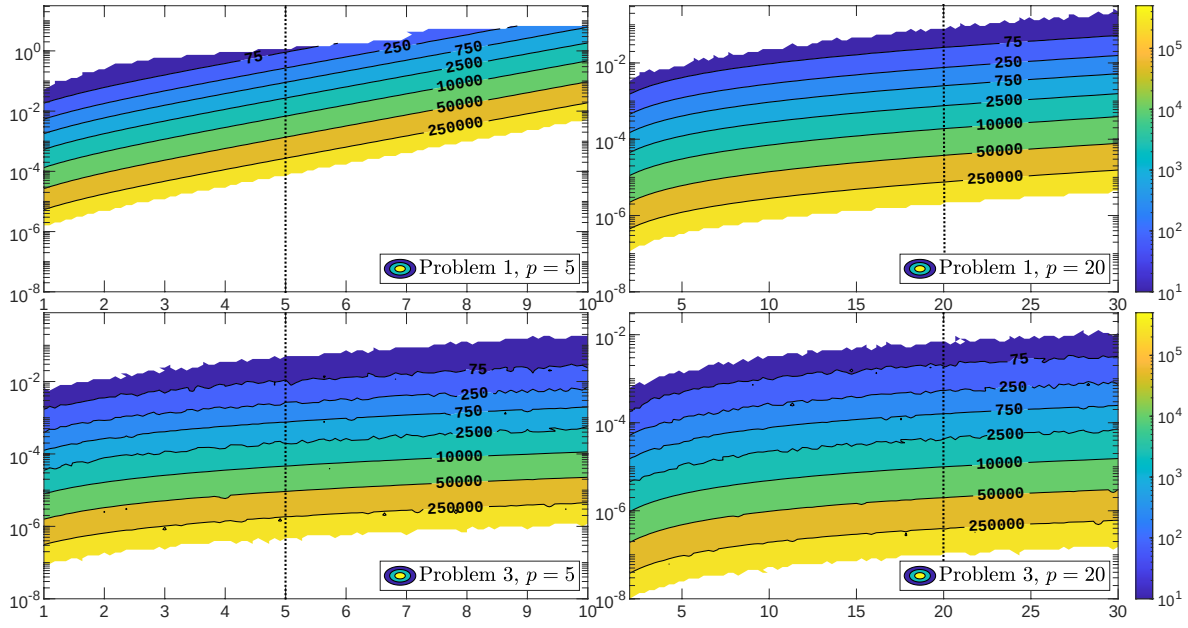


Figure 5.22: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-6}$ in the $(\mathring{p}, h)$-plane, for PolyHTVI applied to Problems 5.1 (with $C = 10^{-5}$) and 5.3 (with $C = 1$). The dotted line represents the non-adaptive algorithm $p = \mathring{p}$.

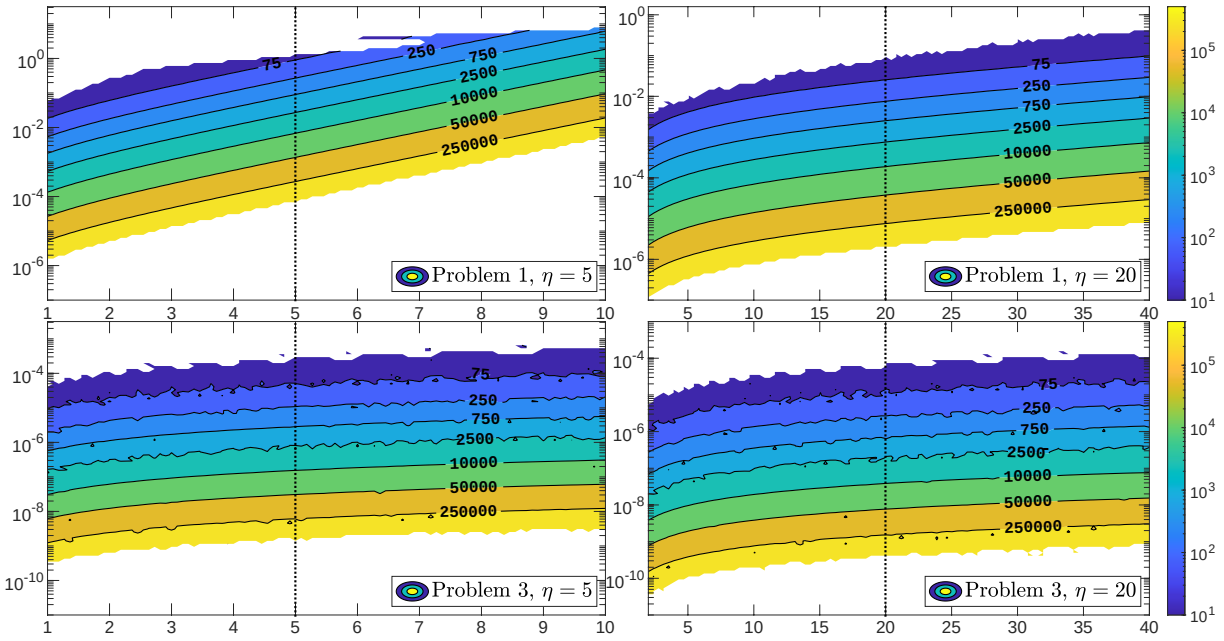


Figure 5.23: Contour plot of the number of iterations required to achieve convergence with $\delta = 10^{-6}$ in the $(\mathring{\eta}, h)$-plane, for ExpoHTVI applied to Problems 5.1 (with $C = 10^{-5}$) and 5.3 (with $C = 10^{5}$). The dotted line represents the non-adaptive algorithm $\eta = \mathring{\eta}$.

Overall, the use of time-adaptivity allows for a larger family of algorithms from which one might be able to extract a more efficient algorithm than without time-adaptivity. However, our numerical experiments suggest that with momentum restarting, the benefits time-adaptivity may provide are very limited and are not worth the computational effort of tuning one additional parameter $\mathring{p}$ or $\mathring{\eta}$. For this reason, we will now discard time-adaptivity, and focus on the non-adaptive approaches. More precisely, we will not consider the ExpoToPoly and PolyToExpo Bregman subfamilies anymore, and will only focus on the $p = \mathring{p}$ polynomial and $\eta = \mathring{\eta}$ exponential Bregman subfamilies.

5.6 Comparison of Integrators

Without time-adaptivity, the optimization algorithms derived in Section 5.3.1 can be simplified into the following optimization algorithms which are now all explicit:

PolyHTVI

$$r_{k+1} = r_k - Chp\mathfrak{q}_k^{2p-1}\nabla f(q_k)$$
$$q_{k+1} = q_k + hp\mathfrak{q}_k^{-p-1}r_{k+1}$$
$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + h$$

PolyLTVI

$$q_{k+1} = q_k + hp\mathfrak{q}_k^{-p-1}r_k - Ch^2p^2\mathfrak{q}_k^{p-2}\nabla f(q_k)$$
$$r_{k+1} = \frac{\mathfrak{q}_k^{p+1}}{hp}(q_{k+1} - q_k)$$
$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + h$$

PolySLC

$$r \leftarrow r - \frac{1}{2}Chp\mathfrak{q}^{2p-1}\nabla f(q)$$
$$\mathfrak{q} \leftarrow \mathfrak{q} + \frac{h}{2}$$
$$q \leftarrow q + hp\mathfrak{q}^{-p-1}r$$
$$\mathfrak{q} \leftarrow \mathfrak{q} + \frac{h}{2}$$
$$r \leftarrow r - \frac{1}{2}Chp\mathfrak{q}^{2p-1}\nabla f(q)$$

PolySV

$$r_{k+\frac{1}{2}} = r_k - \frac{1}{2}Chp\mathfrak{q}_k^{2p-1}\nabla f(q_k)$$
$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + h$$
$$q_{k+1} = q_k + \frac{h}{2}p\left(\mathfrak{q}_k^{-p-1} + \mathfrak{q}_{k+1}^{-p-1}\right)r_{k+\frac{1}{2}}$$
$$r_{k+1} = r_{k+\frac{1}{2}} - \frac{1}{2}Chp\mathfrak{q}_{k+1}^{2p-1}\nabla f(q_{k+1})$$

ExpoHTVI

$$r_{k+1} = r_k - C\eta he^{2\eta\mathfrak{q}_k}\nabla f(q_k)$$
$$q_{k+1} = q_k + \eta he^{-\eta\mathfrak{q}_k}r_{k+1}$$
$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + h$$

ExpoLTVI

$$q_{k+1} = q_k + h\eta e^{-\eta\mathfrak{q}_k}r_k - C\eta^2h^2e^{\eta\mathfrak{q}_k}\nabla f(q_k)$$
$$r_{k+1} = \frac{e^{\eta\mathfrak{q}_k}}{\eta h}(q_{k+1} - q_k)$$
$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + h$$

ExpoSLC

$$r \leftarrow r - \frac{1}{2}C\eta he^{2\eta\mathfrak{q}}\nabla f(q)$$
$$\mathfrak{q} \leftarrow \mathfrak{q} + \frac{h}{2}$$
$$q \leftarrow q + \eta he^{-\eta\mathfrak{q}}r$$
$$\mathfrak{q} \leftarrow \mathfrak{q} + \frac{h}{2}$$
$$r \leftarrow r - \frac{1}{2}C\eta he^{2\eta\mathfrak{q}}\nabla f(q)$$

ExpoSV

$$r_{k+\frac{1}{2}} = r_k - \frac{1}{2}C\eta he^{2\eta\mathfrak{q}_k}\nabla f(q_k)$$
$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + h$$
$$q_{k+1} = q_k + \frac{1}{2}\eta h\left(e^{-\eta\mathfrak{q}_{k+1}} + e^{-\eta\mathfrak{q}_k}\right)r_{k+\frac{1}{2}}$$
$$r_{k+1} = r_{k+\frac{1}{2}} - \frac{1}{2}C\eta he^{2\eta\mathfrak{q}_{k+1}}\nabla f(q_{k+1})$$

It should be noted that the HTVI and LTVI algorithms are now equivalent, and will be referred to as LTVI from now on (since the construction of LTVIs extends to Riemannian manifolds, while that of HTVIs does not). Note that we are still using momentum restarting based on the gradient scheme in all our numerical experiments.

Regions of convergence in the (C, h) and (p, h) planes for the different algorithms were computed based on 100×100 grids of points and are presented in Figures 5.24, 5.25, and 5.26. We can see that the regions of fast convergence both in the (C, h)-plane and (p, h)-plane for the different algorithms are almost identical, regardless of the termination criteria. It seems that the three algorithms perform in a very similar way in terms of computational efficiency, robustness, and stability.

We pushed the numerical experimentation further by solving Problems 5.1, 5.2, 5.3, and 5.4 on a 3-dimensional grid of 100^3 points in (C, p, h)-space. The results are displayed in Figures 5.27, 5.28, 5.29, 5.30.

The top barplots investigate the overall robustness and efficiency of the algorithms. They display the percentages of time each algorithm met the convergence criteria under certain numbers of iterations. For instance, the first bar in the top plot of Figure 5.27 shows that SV converged in < 50 iterations roughly 5% of the time, in < 100 iterations close to 10% of the time, and so on.

Each bar in the middle barplots compares the performance of two algorithms for a specific problem, by displaying the percentages of times they outperformed each other. For instance, the last bar in the middle plot of Figure 5.27 shows that the SV algorithm outperformed its LTVI counterpart about 21% of the time, while the LTVI algorithm outperformed the SV algorithm roughly 11% of the time.

The bottom barplots quantify the gain in efficiency of each algorithm versus the others, by displaying the speedups observed in terms of number of iterations required. For instance, the last bar in the bottom plot of Figure 5.27 shows that LTVI, when compared to SV, achieves a speedup $> 1.5\times$ roughly 4.5% of the time, $> 2\times$ roughly 3.5% of the time, etc... Note that for each bar, we only considered triplets (C, h, p) for which both algorithms converged within 10000 iterations.

Let us first focus on the polynomial Bregman family, based on the numerical results displayed in Figures 5.27 and 5.28.

The top plots confirm the earlier observation that the 3 algorithms have very similar regions of fast convergence, and are thus comparable in terms of robustness.

The middle and bottom plots indicate that SLC outperforms SV more often than vice-versa, although both scenarios occur rather rarely. Although LTVI seems to outperform SLC/SV roughly as regularly as vice-versa, the speedups LTVI allows when compared to SLC/SV are not as significant and frequent as the slowdowns it entails.

It should also be noted that as the termination criteria are made stricter, the differences and significant speedups between the methods become much rarer (although this is partially due to the fact that smaller tolerances require more iterations and we stopped iterating after 10000 iterations).

Overall, the three different algorithms perform very similarly, but the numerical results presented here suggest that SLC might be a slightly better choice within the polynomial Bregman family.

Let us now focus on the exponential Bregman family, based on the numerical results displayed in Figures 5.29 and 5.30. As for the polynomial family, the 3 algorithms perform very similarly in terms of robustness, and the differences in performance between the algorithms become less significant as the convergence criteria are made stricter. SLC and SV perform almost identically on all problems, regardless of the convergence criteria. Now, it seems that LTVI outperforms SLC/SV slightly more often than vice-versa with the more relaxed tolerances, but with less significant speedups, and as the convergence criteria is made stricter, SLC/SV algorithms seem to outperform LTVI.

Overall, the three different algorithms perform very similarly, but the numerical results presented here suggest that SLC/SV might be the slightly better choices for the exponential Bregman family.

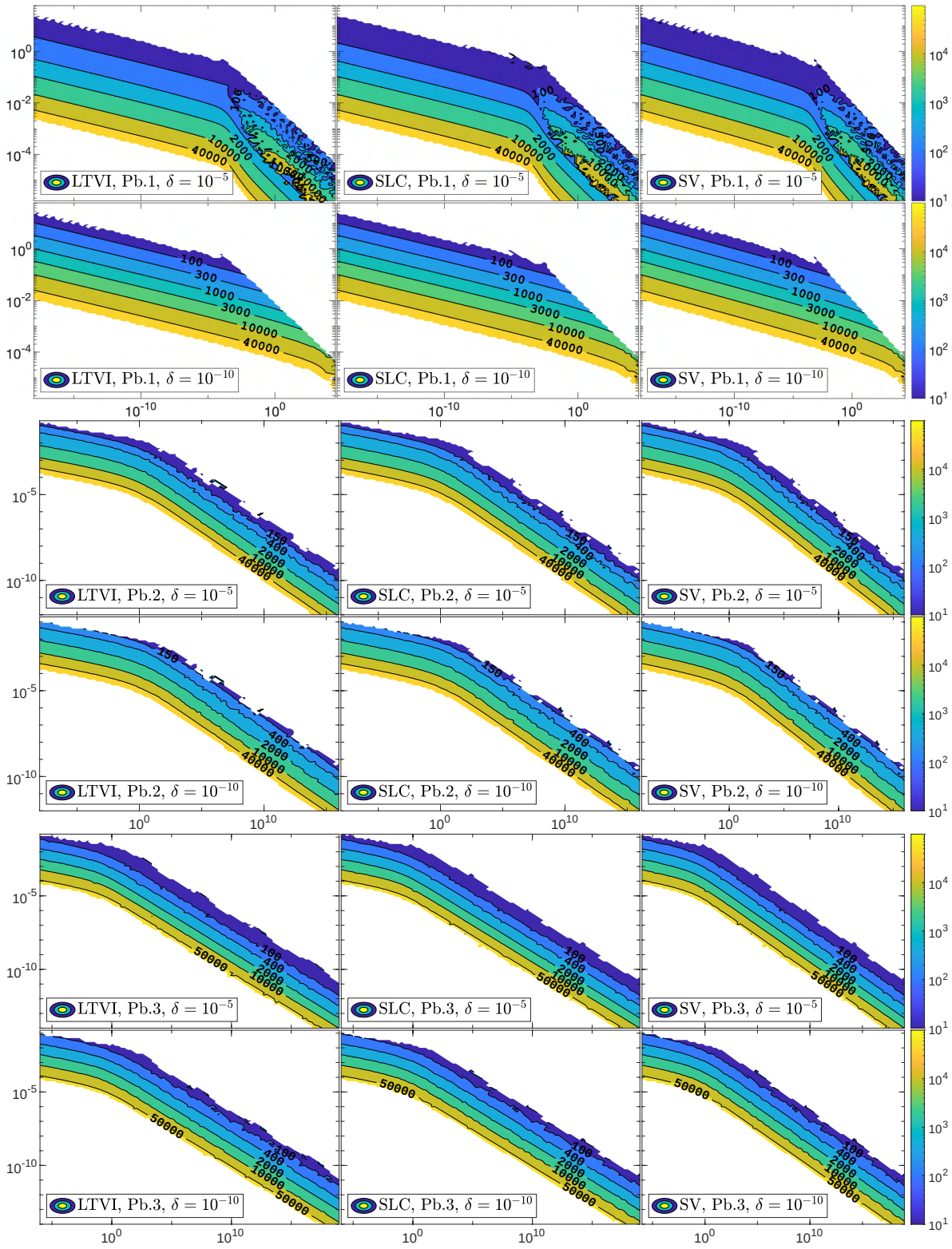


Figure 5.24: Convergence regions in the (C, h)-plane for the PolyLTVI, PolySLC and PolySV algorithms applied to Problems 5.1, 5.2, 5.3 with $p = 8$.

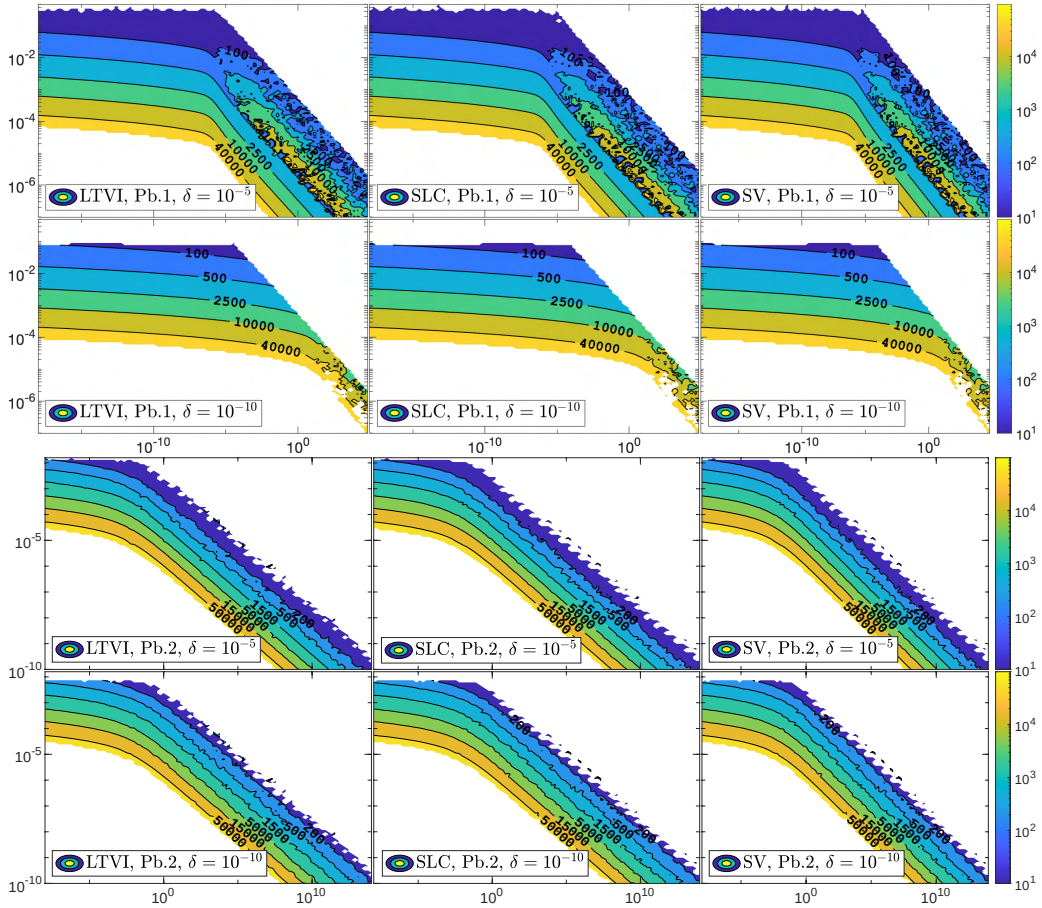


Figure 5.25: Convergence regions in the (C, h)-plane for the ExpoLTVI, ExpoSLC and ExpoSV algorithms applied to Problems 5.1, 5.2 with $\eta = 6$.

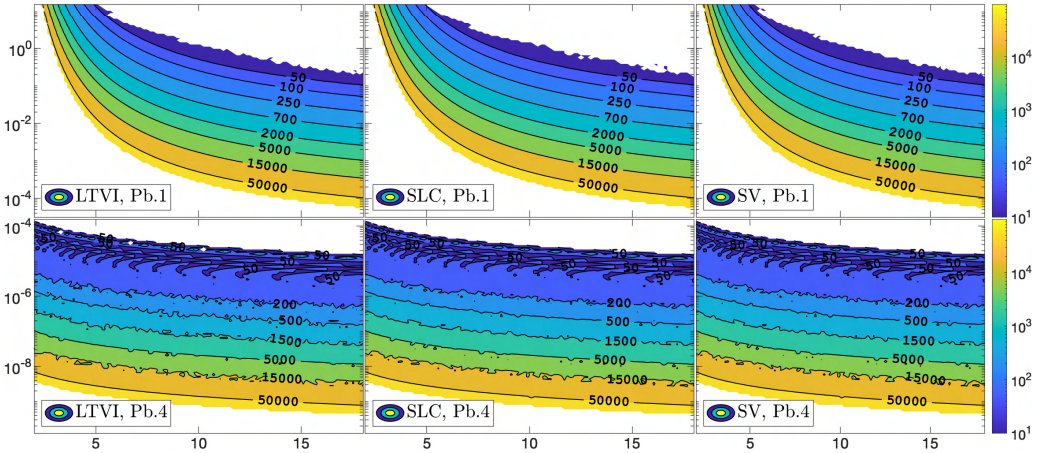


Figure 5.26: Convergence regions in the (p, h)-plane for the PolyLTVI, PolySLC and PolySV algorithms applied to Problems 5.1, 5.4.

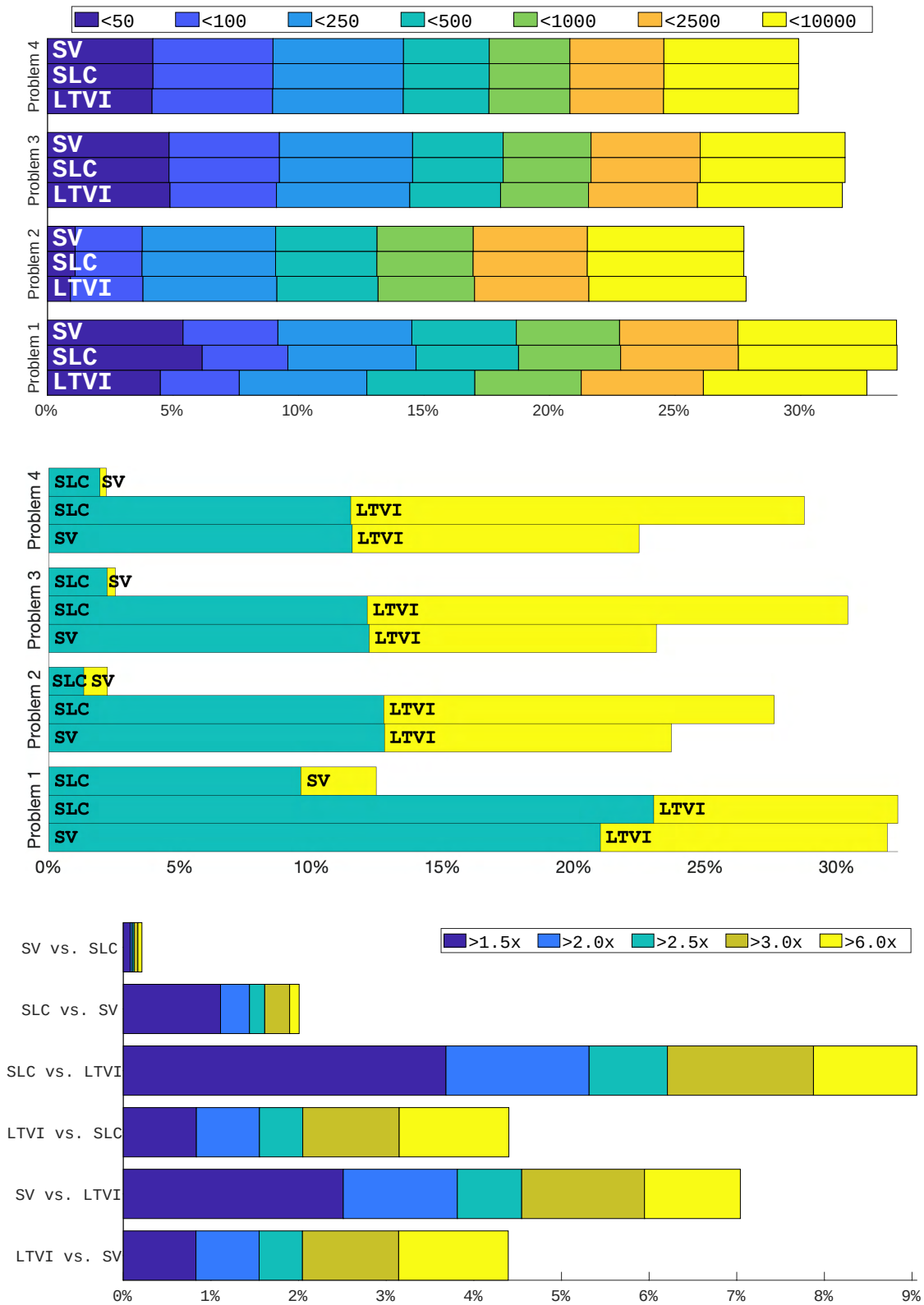


Figure 5.27: Results for the polynomial Bregman family with $\delta = 10^{-5}$.

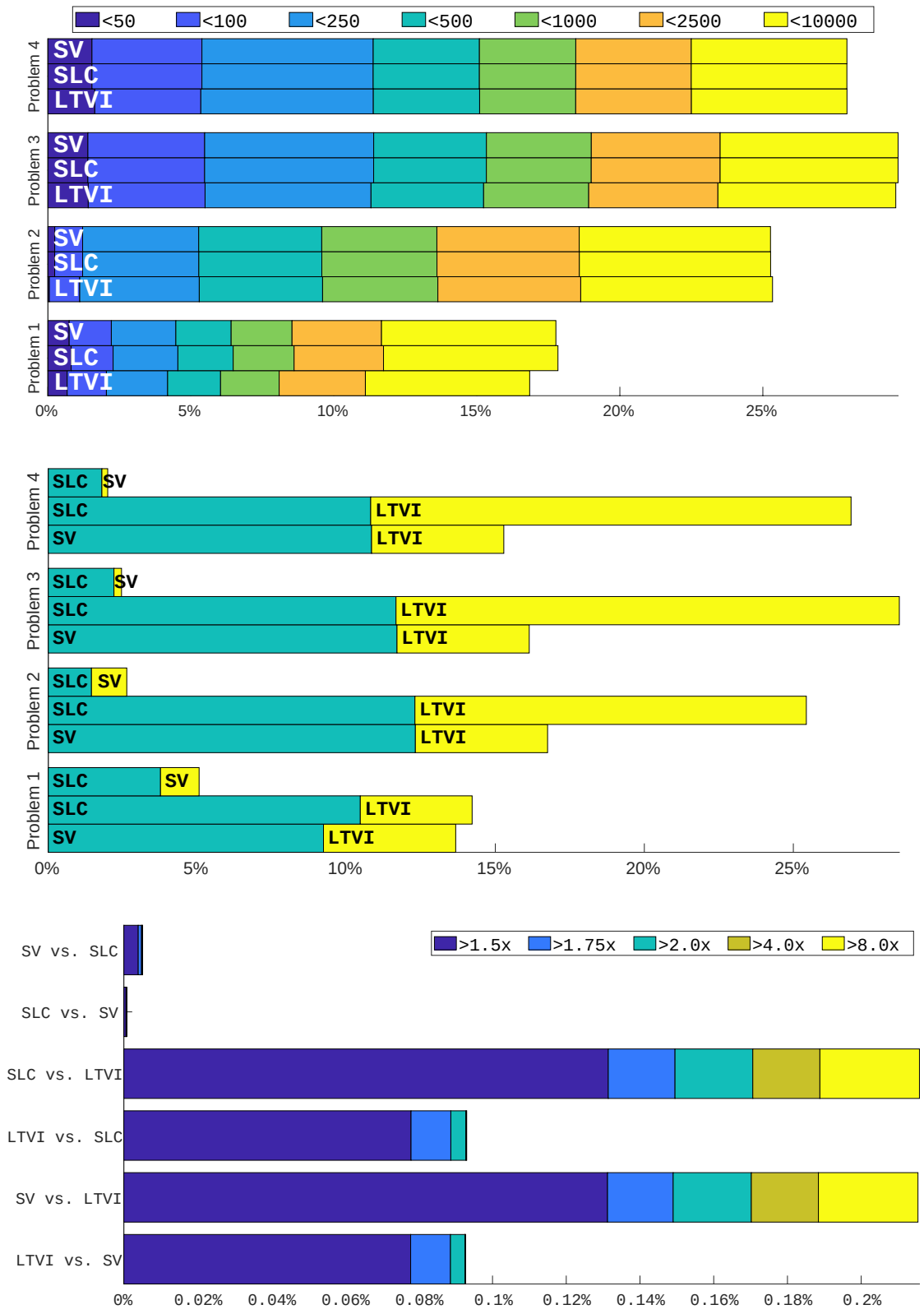


Figure 5.28: Results for the polynomial Bregman family with $\delta = 10^{-10}$.

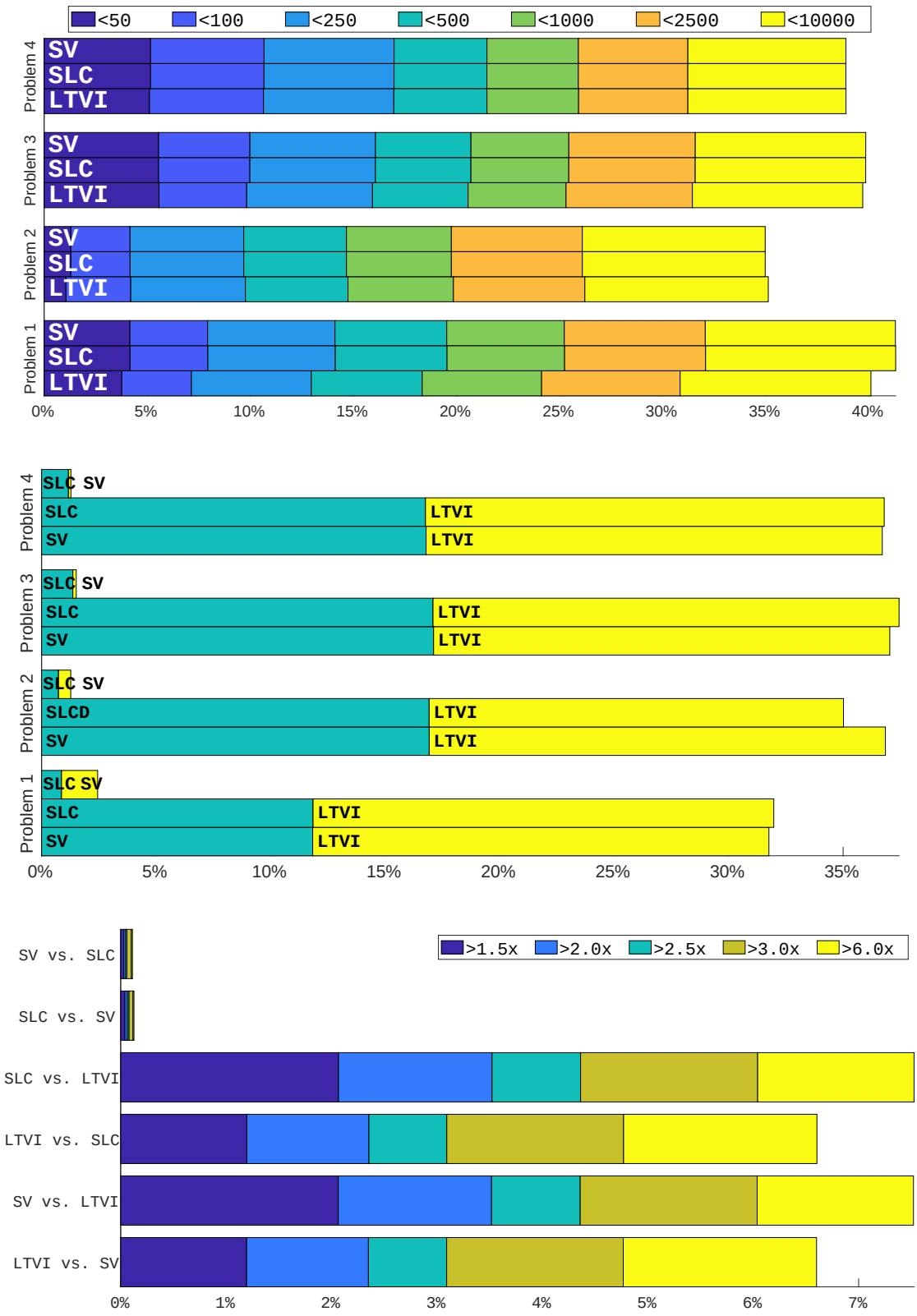


Figure 5.29: Results for the exponential Bregman family with $\delta = 10^{-5}$.

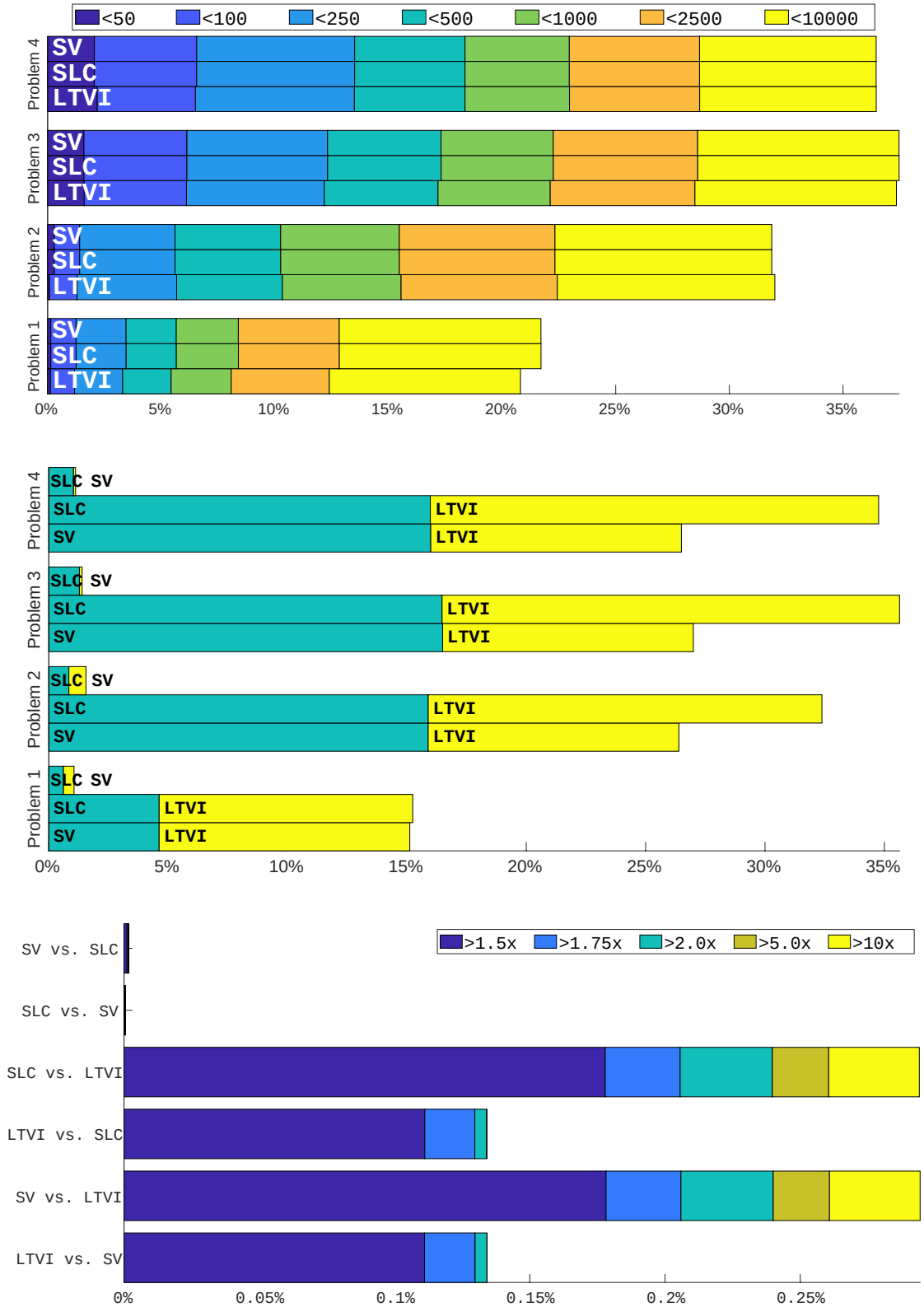


Figure 5.30: Results for the exponential Bregman family with $\delta = 10^{-10}$.

To conclude this section, all the algorithms seem to perform very well and with very small discrepancies, but if we had to choose an algorithm to integrate the Bregman dynamics, it seems that the SLC algorithms with momentum restarting are the slightly better choices. Algorithms 6, 7 show more detailed pseudocodes for the SLC algorithms:

Algorithm 6: **S**ymmetric **L**eapfrog **C**omposition of Component Dynamics for **Poly**nomial Bregman dynamics, with Momentum **R**estarting (**PolySLC-R**)

Input: An objective function $f : \mathbb{R}^d \to \mathbb{R}$. An initial guess $q \in \mathbb{R}^d$. Parameters $C, h, p > 0$.

$\mathfrak{q} \leftarrow 1, \quad G \leftarrow \nabla f(q), \quad r \leftarrow -\frac{1}{2} Chp\mathfrak{q}^{2p-1} G$

while *convergence criteria are not met* **do**

- $\Delta q \leftarrow hp\left(\mathfrak{q} + \frac{h}{2}\right)^{-p-1} r$
- $q \leftarrow q + \Delta q$
- $G \leftarrow \nabla f(q)$
- **if** $G^\intercal \Delta q > 0$ **then** restart momentum: $r \leftarrow 0$
- $\mathfrak{q} \leftarrow \mathfrak{q} + h$
- $r \leftarrow r - Chp\mathfrak{q}^{2p-1} G$

Algorithm 7: **S**ymmetric **L**eapfrog **C**omposition of Component Dynamics for **Expo**nential Bregman dynamics with Momentum **R**estarting (**ExpoSLC-R**)

Input: An objective function $f : \mathbb{R}^d \to \mathbb{R}$. An initial guess $q \in \mathbb{R}^d$. Parameters $C, h, \eta > 0$.

$\mathfrak{q} \leftarrow 1, \quad G \leftarrow \nabla f(q), \quad r \leftarrow -\frac{1}{2} C\eta h e^{2\eta\mathfrak{q}} G$

while *convergence criteria are not met* **do**

- $\Delta q \leftarrow \eta h e^{-\eta\left(\mathfrak{q} + \frac{h}{2}\right)} r$
- $q \leftarrow q + \Delta q$
- $G \leftarrow \nabla f(q)$
- **if** $G^\intercal \Delta q > 0$ **then** restart momentum: $r \leftarrow 0$
- $\mathfrak{q} \leftarrow \mathfrak{q} + h$
- $r \leftarrow r - C\eta h e^{2\eta\mathfrak{q}} G$

5.7 Tuning the Algorithms

We will now investigate how the PolySLC-R and ExpoSLC-R algorithms perform as the parameters C, h, p, η are varied, and try to reduce the number of parameters needing tuning in practice.

5.7.1 Tuning PolySLC-R

We solved Problems 5.1, 5.2, 5.3, 5.4 and two distinct randomly generated instances of Problem 5.5 using PolySLC-R on a 3-dimensional grid of $500 \times 153 \times 500$ points in (C, p, h)-space (logarithmically-spaced in C between 10^{-12} and 10^{12}, logarithmically-spaced in h between 10^{-6} and 10^{3}, and linearly-spaced in p between 2 and 40), and recorded the number of iterations needed to achieve convergence with $\delta = 10^{-10}$. Figure 5.31 displays the number of (C, h) pairs for which convergence was achieved under 200 and 50 iterations for each value of p. We can see that the value of p does not seem to significantly affect the number of (C, h) pairs that exhibit fast convergence, once it is taken to be sufficiently large, so tuning the parameter p carefully might not be very helpful and necessary. For numerical stability reasons, which will be discussed in Section 5.8, it is preferable to use lower values of p, so we will set $p = 6$ since this is a small value of p which performed very well in Figure 5.31.

We solved the same problems using PolySLC-R with $p = 6$ on a 2-dimensional grid of 200×10000 logarithmically-spaced points in (C, h)-space (C between 10^{-15} and 10^{15}, h between 10^{-8} and 10^{4}). The results, presented in Figure 5.32, confirm the observation made in Section 5.4.2 that there is no universally optimal value of C. However, $C = 0.1$ is an intermediate value which seems to work well for most problems, so we will set it as the default value, but it might need some tuning in practice. A similar experiment was conducted for 10^5 logarithmically-spaced values of h using PolySLC-R with $p = 6$ and $C = 0.1$, and Figure 5.33 shows that $h = 0.01$ could be a good default value.

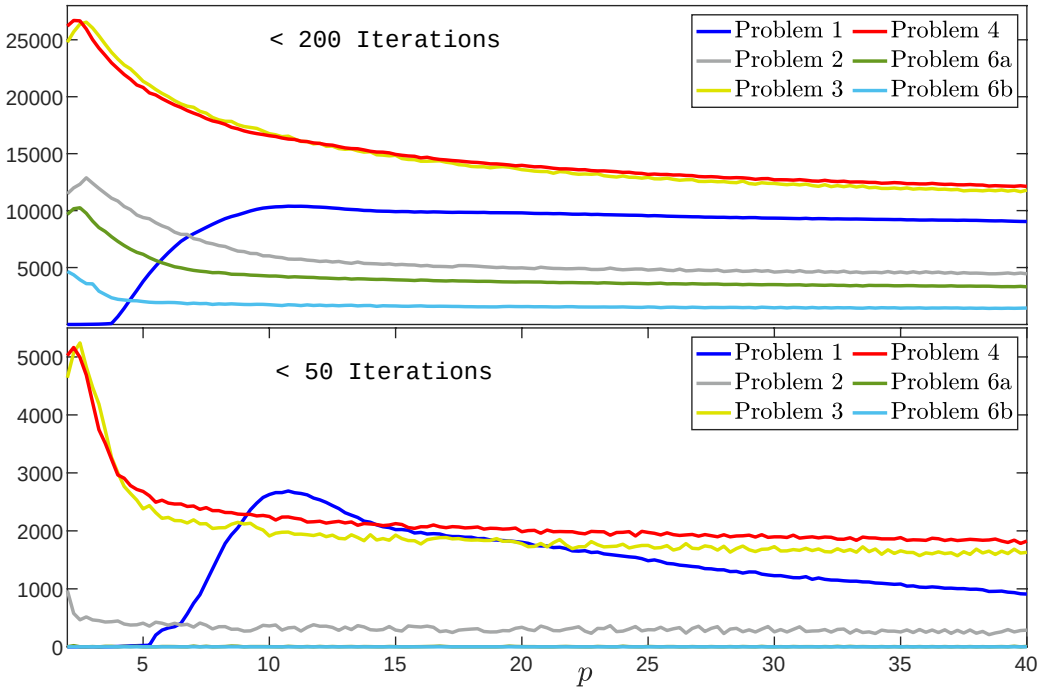


Figure 5.31: Number of (C, h) pairs (out of 500^2) for which convergence was achieved under 200 and 50 iterations using PolySLC-R, as the value of p is varied.

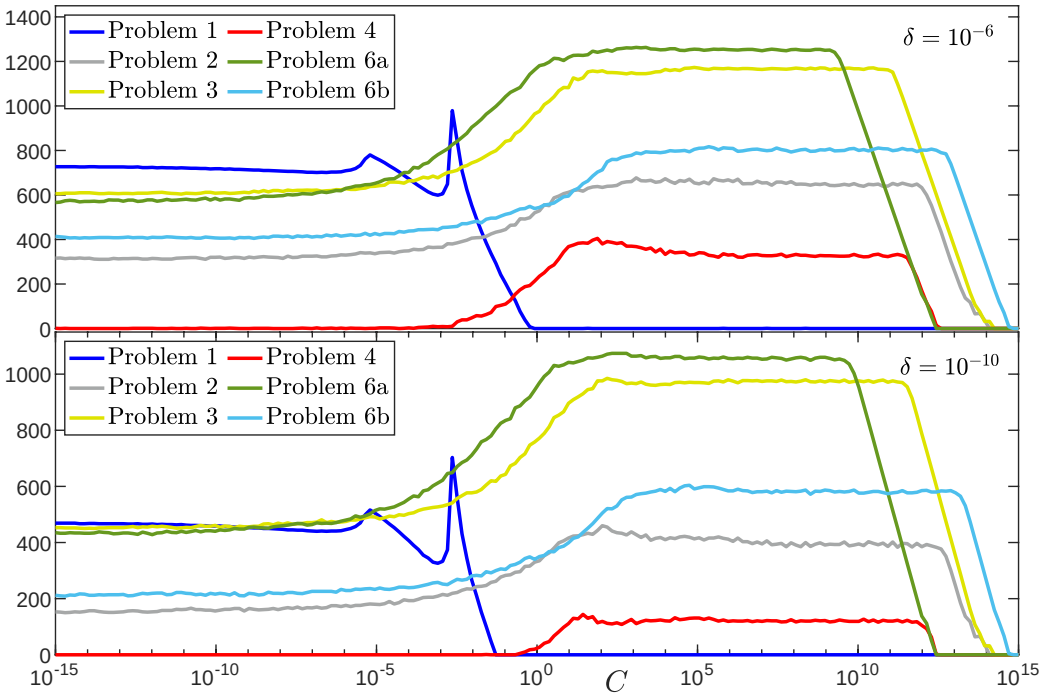


Figure 5.32: Number of values of h (out of 10^4) for which convergence was achieved under 200 iterations using PolySLC-R with $p = 6$, as the value of C is varied.

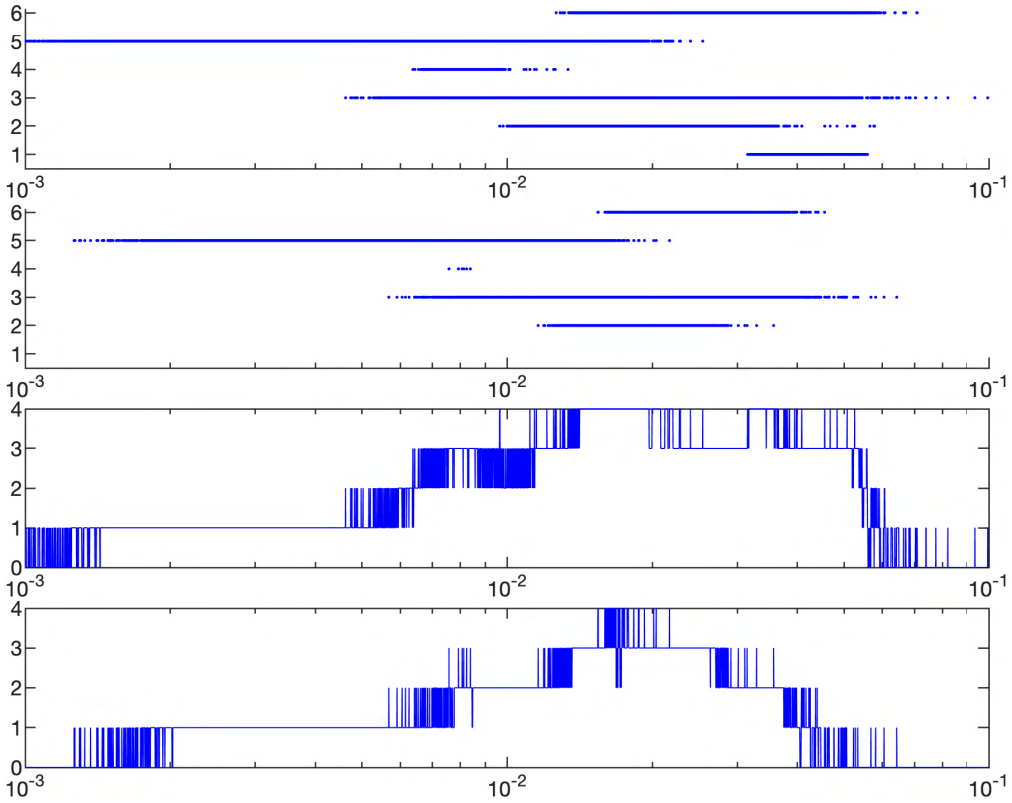


Figure 5.33: The top two plots display the values of h for which PolySLC-R ($p = 6$, $C = 0.1$) converged in < 200 iterations for each of the 6 problems considered. The bottom two plots display the number of problems (out of 6) that were solved in < 200 iterations.

5.7.2 Tuning ExpoSLC-R

We solved Problems 5.1, 5.2, 5.3, 5.4 and 2 distinct randomly generated instances of Problem 5.5 using ExpoSLC-R on a 3-dimensional grid of logarithmically-spaced $500 \times 100 \times 500$ points in (C, η, h)-space (C between 10^{-12} and 10^{12}, h between 10^{-6} and 10^{3}, η between 10^{-5} and 10^{2}), and recorded the number of iterations needed to achieve convergence with $\delta = 10^{-10}$. Figure 5.34 displays the number of (C, h) pairs for which convergence was achieved under 200 and 50 iterations for each value of η. Just like the value of p in the polynomial Bregman algorithm, the value of η does not seem to significantly affect the number of (C, h) pairs of fast convergence, as long as it falls between 0.001 and 10. Therefore, tuning the parameter η carefully might not be very helpful and necessary, so we will fix it to $\eta = 0.01$.

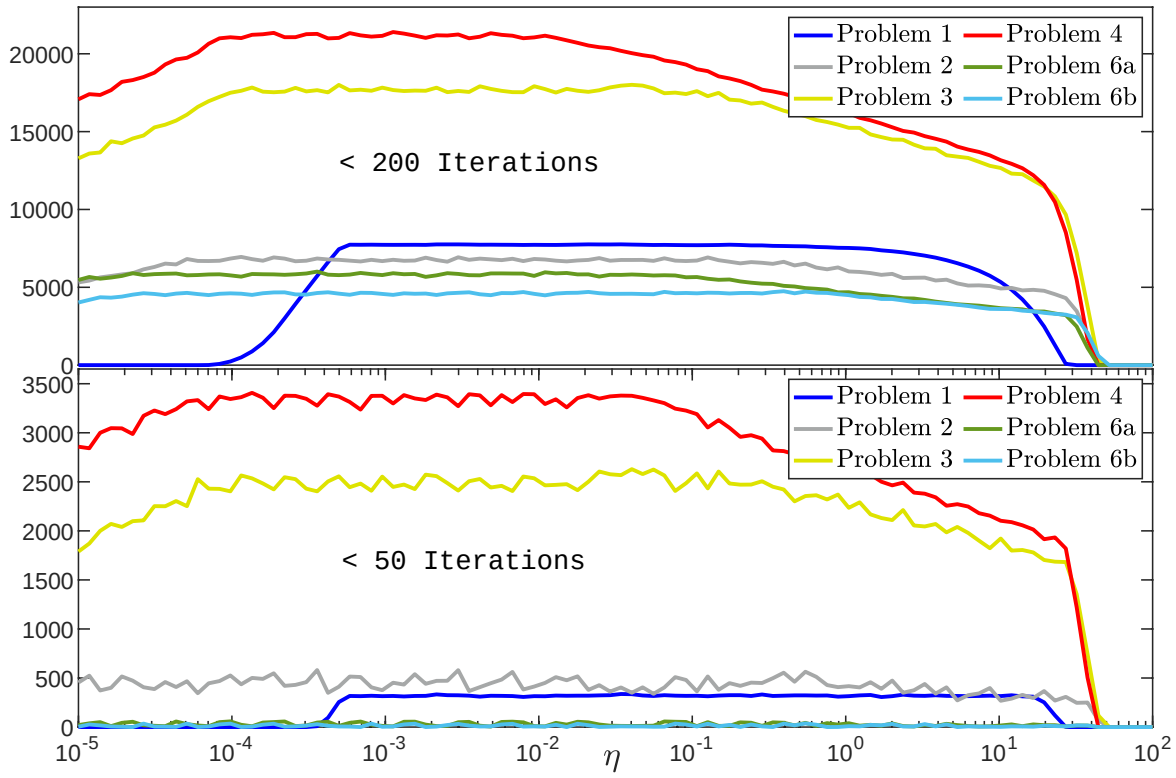


Figure 5.34: Number of (C, h) pairs (out of 500^2) for which convergence was achieved under 200 and 50 iterations using ExpoSLC-R, as the value of η is varied.

We then solved Problems 5.1, 5.2, 5.3, 5.4 and two distinct randomly generated instances of Problem 5.5 using ExpoSLC-R with $\eta = 0.01$ on a 2-dimensional grid of 200×10000 logarithmically-spaced points in (C, h)-space (C between 10^{-15} and 10^{15}, h between 10^{-8} and 10^{4}). The results, presented in Figure 5.35 confirm the observation made in Section 5.4.2 that there is no value of C which is universally optimal. However, $C = 1$ seems to work well for most problems, so we will set it as the default value, but it might need some tuning in practice.

A similar experiment was conducted for 10^5 logarithmically-spaced values of h using PolySLC-R with $\eta = 0.01$ and $C = 1$, and Figure 5.36 shows that $h = 4$ seems to perform well on most problems considered here.

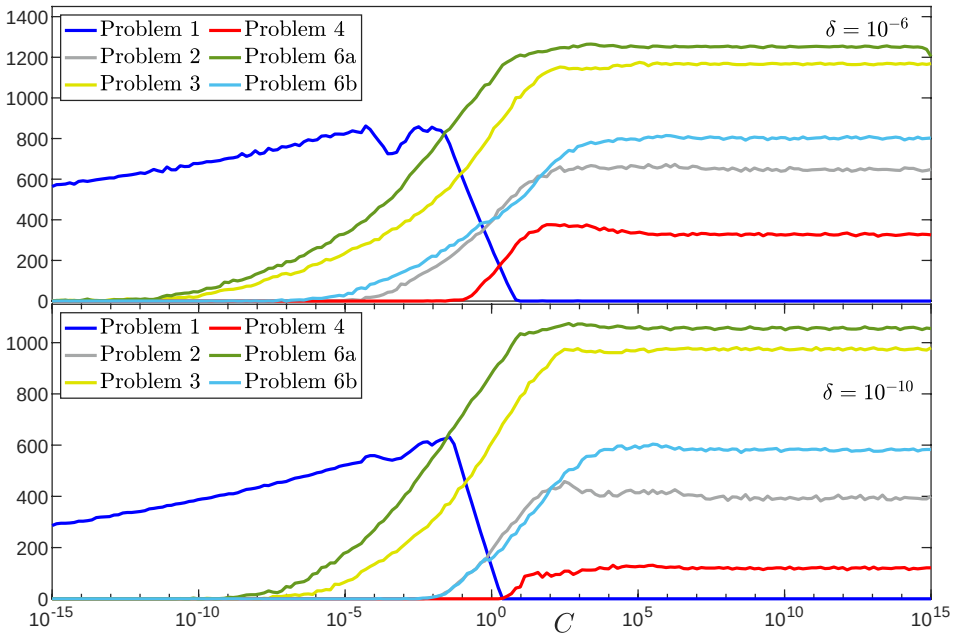


Figure 5.35: Number of values of h (out of 10^4) for which convergence was achieved under 200 iterations using ExpoSLC-R with $\eta = 0.01$, as the value of C is varied.

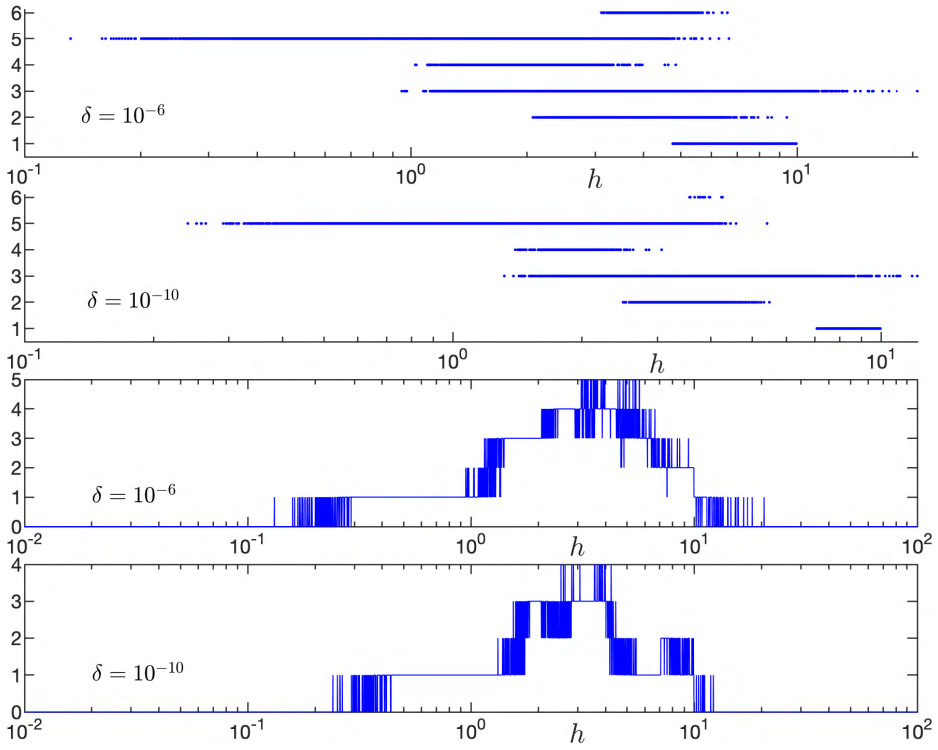


Figure 5.36: The top two plots display the values of h for which ExpoSLC-R ($\eta = 0.01$, $C = 1$) converged in < 200 iterations for each of the 6 problems considered. The bottom two plots display the number of problems (out of 6) that were solved in < 200 iterations.

5.8 Temporal Looping Improves Numerical Stability

There is an important caveat to the promising performance observed for the optimization algorithms constructed in this chapter. The evolution of the variables $q, \mathfrak{q}$ and r associated with the exponential Poincaré Hamiltonian,

$$\bar{H}^{\eta}(\bar{q}, \bar{r}) = \frac{\eta}{2e^{\eta \mathfrak{q}}}\langle r, r\rangle + C\eta e^{2\eta \mathfrak{q}} f(q) + \mathfrak{r}, \tag{5.64}$$

is guided by Hamilton's equations,

$$\dot{q} = \eta e^{-\eta \mathfrak{q}} r, \qquad \dot{r} = -C\eta e^{2\eta \mathfrak{q}} \nabla f(q), \qquad \dot{\mathfrak{q}} = 1. \tag{5.65}$$

From these equations of motion, we can see that the time variable $\mathfrak{q}$ grows linearly without bound, and as a result quantities like $e^{\eta \mathfrak{q}}$ grow exponentially without bound. In practice, this will cause numerical instability due to numerical precision issues, since the unbounded growth of certain terms cannot be balanced by the decay to 0 of other quantities which is limited by machine precision. More precisely, looking at the updates of Algorithm 7,

$$r \leftarrow r - C\eta h e^{2\eta \mathfrak{q}} \nabla f(q), \qquad \mathfrak{q} \leftarrow \mathfrak{q} + h, \qquad q \leftarrow q + \Delta q = q + \eta h e^{-\eta\left(\mathfrak{q} + \frac{h}{2}\right)} r, \tag{5.66}$$

we have at every iteration that

$$\Delta q \;\leftarrow\; A(\Delta q)_{previous} + B e^{\eta \mathfrak{q}} \nabla f(q), \tag{5.67}$$

for some constants A and B. The decay to 0 of $\nabla f(q)$ is limited by machine precision, while $e^{\eta \mathfrak{q}}$ grows without bound, and eventually the quantity $Be^{\eta \mathfrak{q}} \nabla f(q)$ becomes large again and the position variable q moves away from the equilibrium it found near its optimal value. Something analogous happens for the Bregman polynomial subfamily, except that the growing exponential term is replaced by an unbounded growing polynomial term.

This numerical instability phenomenon is illustrated in Figure 5.37 which displays the evolution of the error $|f(x_k) - f(x^*)|$ when the PolySLC-R and ExpoSLC-R algorithms are applied to Problem 5.2. We can see that both algorithms first achieve convergence to machine precision (after 134 and 154 iterations, respectively), stay at the minimizer for a few hundred iterations, and finally are expelled away from the minimizer due to numerical instability (after 314 and 399 iterations, respectively).

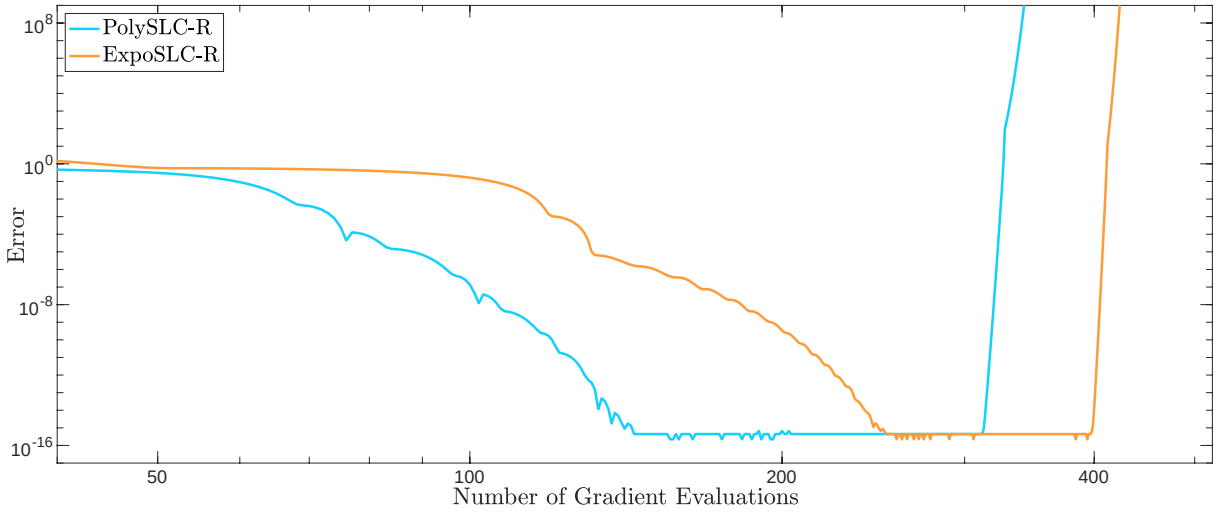


Figure 5.37: The PolySLC-R and ExpoSLC-R algorithms applied to Problem 5.2.

In all our numerical experiments so far, the algorithms stopped when they reached a desired convergence criterion, and as a consequence we did not observe this numerical instability phenomenon as it happens only after convergence is achieved. However, in practice, optimization algorithms are very often terminated after a specified number of iterations instead of a specified tolerance being achieved. Therefore, we need a strategy to avoid this numerical instability phenomenon.

Since the numerical instability results from the limitation imposed by machine precision on accurately representing the decay to 0 of $\nabla f(q)$ while $e^{\eta \mathfrak{q}}$ grows without bound in the exponential subfamily, it is natural to try to avoid this phenomenon by restricting the growth of the term $e^{\eta \mathfrak{q}}$, that is, by restricting the growth of $\mathfrak{q}$ (and similarly in the polynomial case). One possibility is to reset time whenever a certain numerical instability criterion is met, via $\mathfrak{q} \leftarrow \beta \mathfrak{q}$ for some $\beta \in (0, 1)$. A larger β is preferable to keep enough momentum in case convergence to the minimizer was only suboptimal when the numerical instability criterion was met, or if the algorithm is used in an online fashion or with a stochastic or mini-batch approach. It is also preferable to avoid values of β very close to 1, since the algorithm would then always remain close to numerical instability, and could possibly become unstable if the criterion is not chosen very carefully. In practice, taking β between 0.6 and 0.95 works well, by ensuring that a reasonable amount of momentum is kept while avoiding the numerical instability region. Alternatively, one could reset time

via $\mathfrak{q} \leftarrow \mathfrak{q} - \nu h$ for some $\nu > 1$. A smaller ν is preferable to retain momentum, while ν should not be too close to 1 to avoid numerical instability. In practice, we will reset time via $\mathfrak{q} \leftarrow \max(\epsilon, \beta\mathfrak{q})$ or $\mathfrak{q} \leftarrow \max(\epsilon, \mathfrak{q} - \nu h)$, where ϵ is a small positive number, to avoid very small or negative values of time $\mathfrak{q}$. This phenomenon where the time variable $\mathfrak{q}$ is stuck in a loop by resetting $\mathfrak{q} \leftarrow \beta\mathfrak{q}$ or $\mathfrak{q} \leftarrow \mathfrak{q} - \nu h$ whenever numerical instability is near will be referred to as **Temporal Looping**.

Improving the ExpoSLC-R algorithm via temporal looping yields Algorithm 8, which will be referred to as ExpoSLC-RTL:

Algorithm 8: Symmetric **L**eapfrog **C**omposition for **Expo**nential Bregman dynamics, with Momentum **R**estarting and **T**emporal **L**ooping (**ExpoSLC-RTL**)

Input: An objective function $f : \mathbb{R}^d \to \mathbb{R}$. An initial guess $q \in \mathbb{R}^d$.
Parameters $C, h, p > 0$, $\beta \in (0, 1)$ or $\nu > 1$.

$\epsilon \leftarrow 0.001, \quad \mathfrak{q} \leftarrow 1, \quad G \leftarrow \nabla f(q), \quad r \leftarrow -\frac{1}{2} C\eta h e^{2\eta\mathfrak{q}} G$
while *convergence criteria are not met* **do**
  $\Delta q \leftarrow \eta h e^{-\eta(\mathfrak{q} + \frac{h}{2})} r$
  $q \leftarrow q + \Delta q$
  $G \leftarrow \nabla f(q)$
  if $G^\top \Delta q > 0$ **then** restart momentum: $r \leftarrow 0$
  if *numerical instability criterion is met* **then** $\mathfrak{q} \leftarrow \max(\epsilon, \beta\mathfrak{q})$ or $\mathfrak{q} \leftarrow \max(\epsilon, \mathfrak{q} - \nu h)$
  $\mathfrak{q} \leftarrow \mathfrak{q} + h$
  $r \leftarrow r - C\eta h e^{2\eta\mathfrak{q}} G$

In our numerical experiments, we have chosen the following numerical instability criterion to reset time in ExpoSLC-RTL:

$$Ch^2\eta^2 e^{\eta\mathfrak{q}} \|G\| > e^{-\eta h} \|\Delta q\|. \tag{5.68}$$

This numerical instability criterion roughly ensures that $|B| e^{\eta\mathfrak{q}} \|\nabla f(q)\| < |A| \|(\Delta q)_{previous}\|$ in equation (5.67), so that the new position update is not significantly larger in norm than the previous position update.

Improving the PolySLC-R algorithm in a similar way via temporal looping yields the following algorithm:

Algorithm 9: **S**ymmetric **L**eapfrog **C**omposition for **Poly**nomial Bregman dynamics, with Momentum **R**estarting and **T**emporal **L**ooping (**PolySLC-RTL**)

Input: An objective function $f : \mathbb{R}^d \to \mathbb{R}$. An initial guess $q \in \mathbb{R}^d$.
Parameters $C, h, p > 0$, $\beta \in (0,1)$ or $\nu > 1$.

$\epsilon \leftarrow 0.001, \quad \mathfrak{q} \leftarrow 1, \quad G \leftarrow \nabla f(q), \quad r \leftarrow -\frac{1}{2} Chp\mathfrak{q}^{2p-1} G$
while *convergence criteria are not met* **do**
  $\Delta q \leftarrow hp\left(\mathfrak{q} + \frac{h}{2}\right)^{-p-1} r$
  $q \leftarrow q + \Delta q$
  $G \leftarrow \nabla f(q)$
  if $G^\top \Delta q > 0$ **then** restart momentum: $r \leftarrow 0$
  if *numerical instability criterion is met* **then** $\mathfrak{q} \leftarrow \max(\epsilon, \beta\mathfrak{q})$ or $\mathfrak{q} \leftarrow \max(\epsilon, \mathfrak{q} - \nu h)$
  $\mathfrak{q} \leftarrow \mathfrak{q} + h$
  $r \leftarrow r - Chp\mathfrak{q}^{2p-1} G$

In our numerical experiments, we have chosen the numerical instability criterion

$$Ch^2 p^2 (\mathfrak{q} + h)^{p+1} \|G\| > \mathfrak{q} \|\Delta q\|, \tag{5.69}$$

which roughly ensures that the new position update is not significantly larger than the previous one.

Figure 5.38 shows that adding temporal looping in the polynomial and exponential Bregman optimization algorithms takes care of the numerical instability issue.

It can be seen from Figures 5.39 and 5.40 that adding temporal looping, with the $\mathfrak{q} \leftarrow \max(\epsilon, \beta\mathfrak{q})$ scheme or $\mathfrak{q} \leftarrow \max(\epsilon, \mathfrak{q} - \nu h)$ scheme, does not negatively affect the performance of the algorithms, although the algorithms with temporal looping might sometimes require a larger number of iterations to achieve convergence for a fixed (C, h)-pair. Indeed the regions of fast convergence might be shifted slightly, but remained at least as large if not larger when using temporal looping.

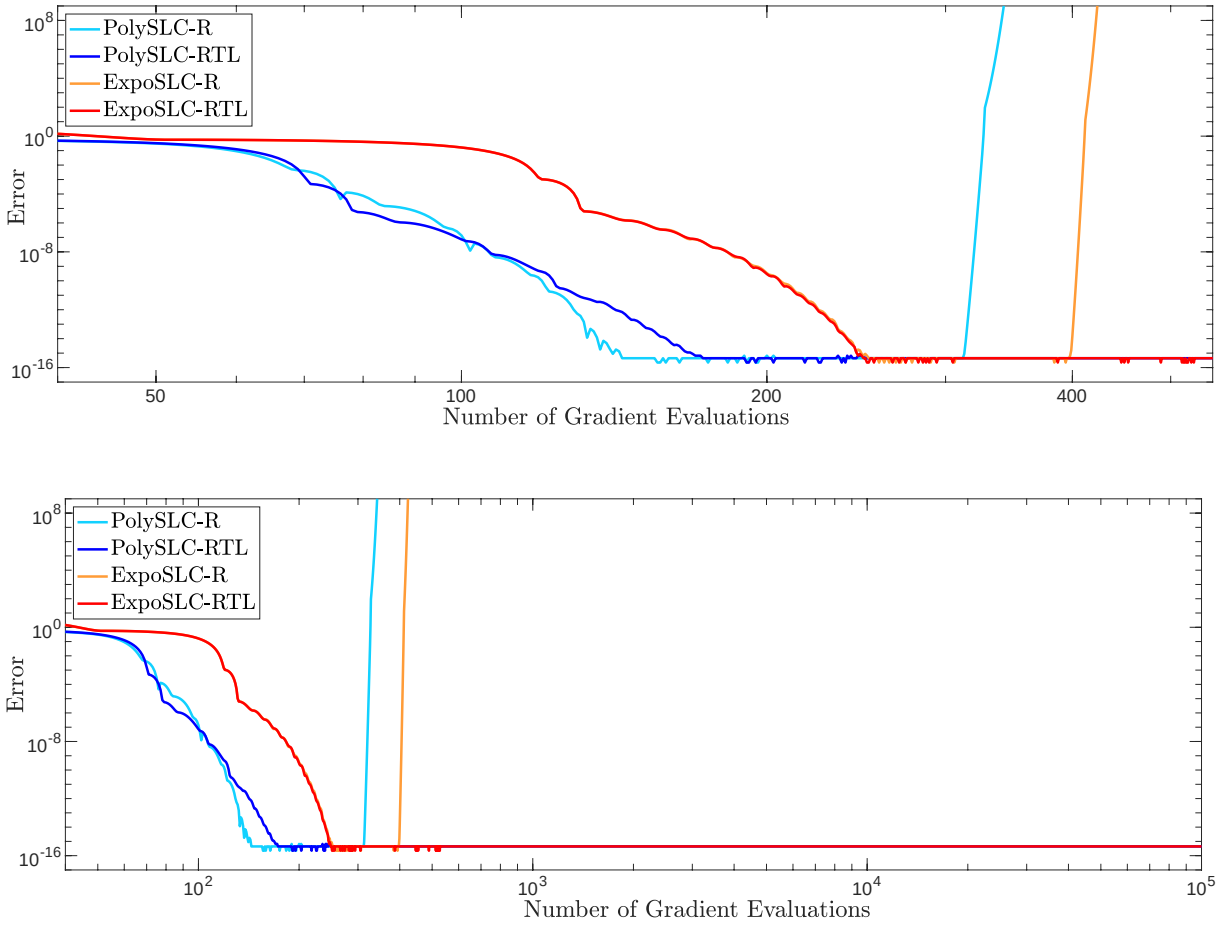


Figure 5.38: The effect of temporal looping, when added to the PolySLC-R and ExpoSLC-R algorithms, and applied to Problem 5.2.

Overall, we have seen that temporal looping can be very helpful to deal with post-convergence numerical instability, and that it does not affect negatively the initial performance of the algorithm. Note that temporal looping could be improved further by tuning the numerical instability parameters β or ν, or by designing a better suited criterion to detect upcoming numerical instability.

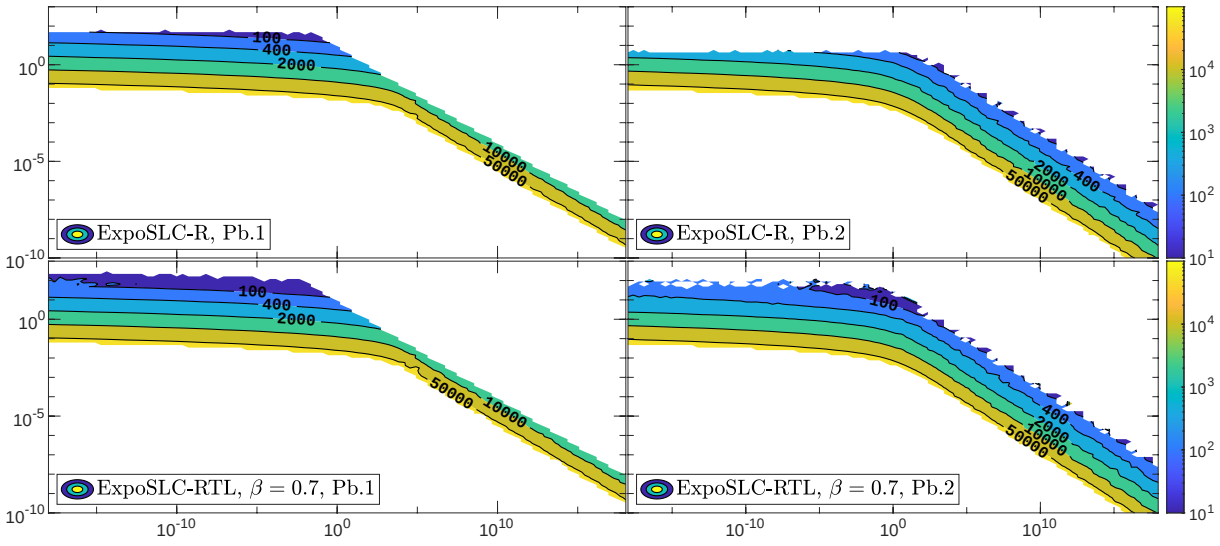


Figure 5.39: Contour plot of the number of iterations required to achieve convergence ($\delta = 10^{-8}$) in the (C, h)-plane, for the ExpoSLC-R and ExpoSLC-RTL algorithms, when applied with $\eta = 0.01$ to Problems 5.1 and 5.2.

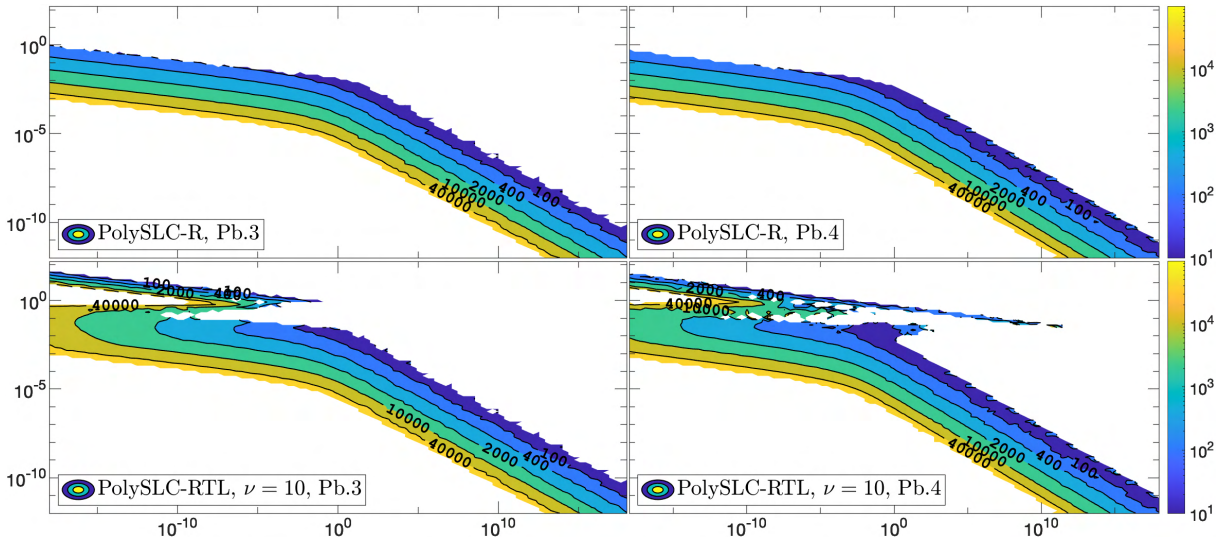


Figure 5.40: Contour plot of the number of iterations required to achieve convergence ($\delta = 10^{-8}$) in the (C, h)-plane, for the PolySLC-R and PolySLC-RTL algorithms, when applied with $p = 10$ to Problems 5.3 and 5.4.

5.9 Testing for Machine Learning Applications

We now test our algorithms on more challenging optimization problems for machine learning with a variety of model architectures, loss functions to minimize, and applications. For reference, we will also solve these optimization problems using gradient descent and the most commonly used optimizer in machine learning, Adam [Kingma and Ba, 2014]:

ADAM

$$m_{k+1} = \beta_1 m_k + (1-\beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1-\beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$\tilde{m} = (1-\beta_1^{k+1})^{-1} m_{k+1}, \quad \tilde{v} = (1-\beta_2^{k+1})^{-1} v_{k+1}$$

$$x_{k+1} = x_k - h(\sqrt{\tilde{v}} + \epsilon)^{-1}\tilde{m}$$

Here, $u \odot v$ denotes the elementwise multiplication of the vectors u and v. The variable ϵ present in the updates of the Adam algorithm is there to avoid numerical instability associated with division by 0 (with default value $\epsilon = 10^{-8}$ in [Kingma and Ba, 2014] and PyTorch). The three parameters of Adam are β_1, β_2 used for computing running averages of gradients, and the learning rate h (with default values $\beta_1 = 0.9$, $\beta_2 = 0.999$, $h = 0.001$ in [Kingma and Ba, 2014], PyTorch and TensorFlow).

The ExpoSLC-RTL and PolySLC-RTL algorithms have been implemented under the more evocative names **eBrAVO** and **pBrAVO** (**Br**egman **A**ccelerated **V**ariational **O**ptimizer) in such a way that they can be used within the TensorFlow and PyTorch frameworks. These algorithms are available at github.com/vduruiss/AccOpt_via_GNI, and can be called in a similar way as the Adam algorithm in TensorFlow and PyTorch.
In TensorFlow:

```
optimizer = tf.keras.optimizers.Adam(learning_rate = 0.001)
optimizer = BrAVO_tf.eBravo(learning_rate = 1)
```

In PyTorch:

```
optimizer = torch.optim.Adam(model.parameters(), lr = 0.01)
optimizer = BrAVO_torch.eBravo(model.parameters(), lr = 1)
```

We have first tested the performance of our BrAVO algorithms with automatic differentiation on instances of the Binary Classification Problem 5.6 and the Fermat–Weber Location Problem 5.7. Figure 5.41 shows the evolution of the loss function (5.49) when trying to formulate a model separating blue and red regions of 2-dimensional space using a line based on the displayed 500 randomly generated points. Figure 5.42 shows the evolution of the loss function (5.52) when trying to solve the Fermat–Weber Location Problem 5.7 with 5000 randomly generated vectors in $\mathbb{R}^{1000}$ and 5000 randomly generated corresponding scalar weights.

We can see from Figures 5.41 and 5.42 that our algorithms solve the binary classification and location problems with an accuracy and efficiency comparable to those of the Adam and standard gradient descent (SGD) algorithms implemented in TensorFlow.

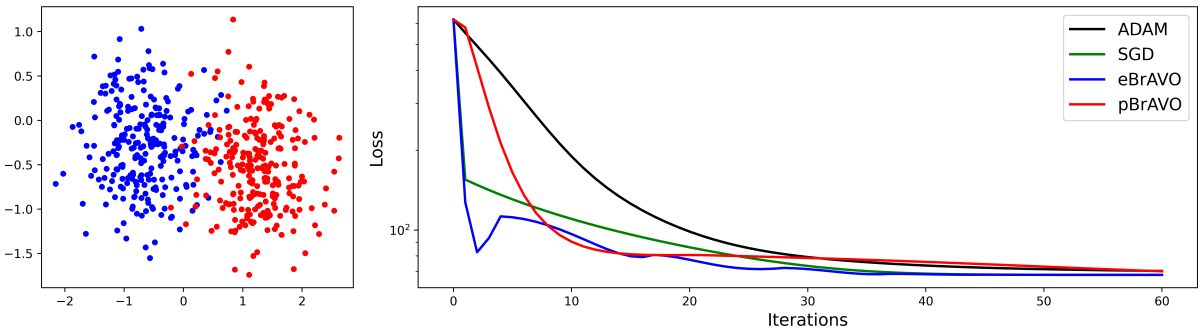


Figure 5.41: Comparison of algorithms applied to a Binary Classification Problem 5.6.

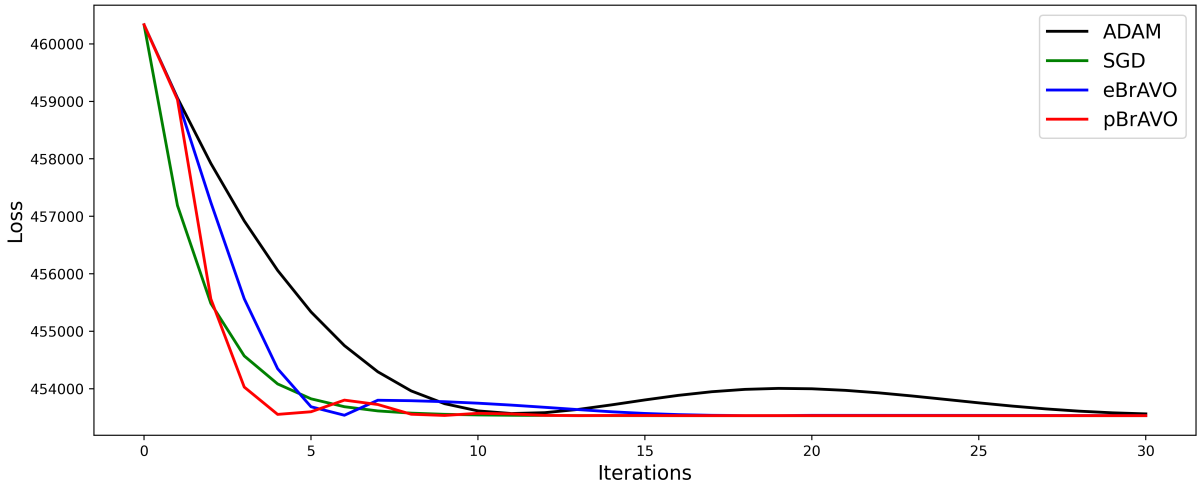


Figure 5.42: Comparison of algorithms applied to a Location Problem 5.7.

Next, we have tested our algorithm on the popular multi-label image classification problem based on the Fashion–MNIST dataset [Xiao et al., 2017]:

'Fashion–MNIST is a dataset of Zalando's article images consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28×28 grayscale image, associated with a label from 10 classes (t-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, ankle boot)'.

We use `nn.CrossEntropyLoss()` as the loss function, and the following neural network architecture in PyTorch as our classification model:

```
Layer (type)      Output Shape          Parameters
=================================================================
dense (Dense)     [-1, 784]             0
dense_1 (Dense)   [-1, 64]              50,240
dense_2 (Dense)   [-1, 64]              0
dense_3 (Dense)   [-1, 64]              4,160
=================================================================
Total Number of Parameters: 55,050
```

Figure 5.43 shows that the BrAVO algorithms achieve comparable accuracy and efficiency on the Fashion–MNIST classification problem as the Adam and gradient descent (SGD) algorithms.

Note that the momentum restarting scheme and the temporal looping strategy are essential to the good behavior of the algorithms. Indeed, we can see from Figure 5.44 that without them, the algorithms eventually lose convergence due to numerical instability. Note as well that these strategies can also allow for larger time-steps which usually translates into faster convergence.

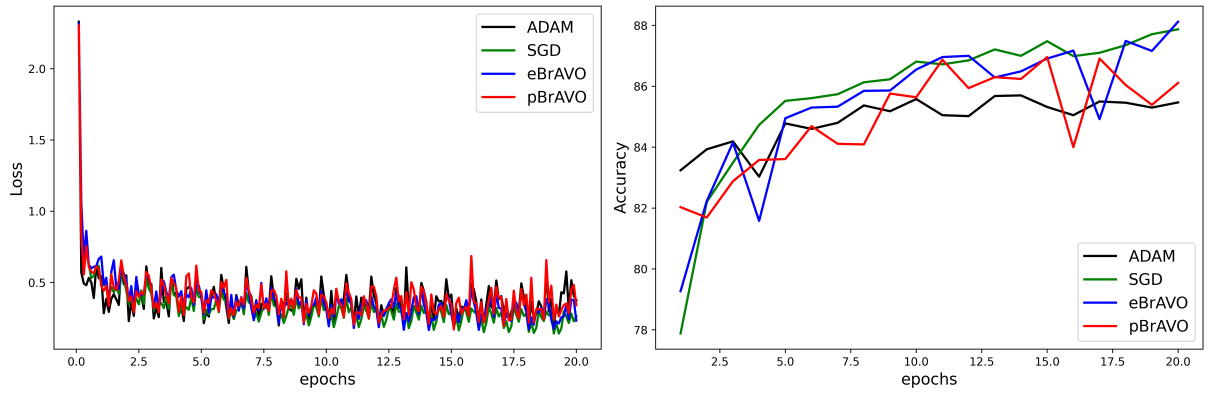


Figure 5.43: Evolution of the loss function and accuracy (in %) of Adam, standard gradient descent (SGD) and the BrAVO algorithms, when applied to the Fashion–MNIST multi-label classification problem.

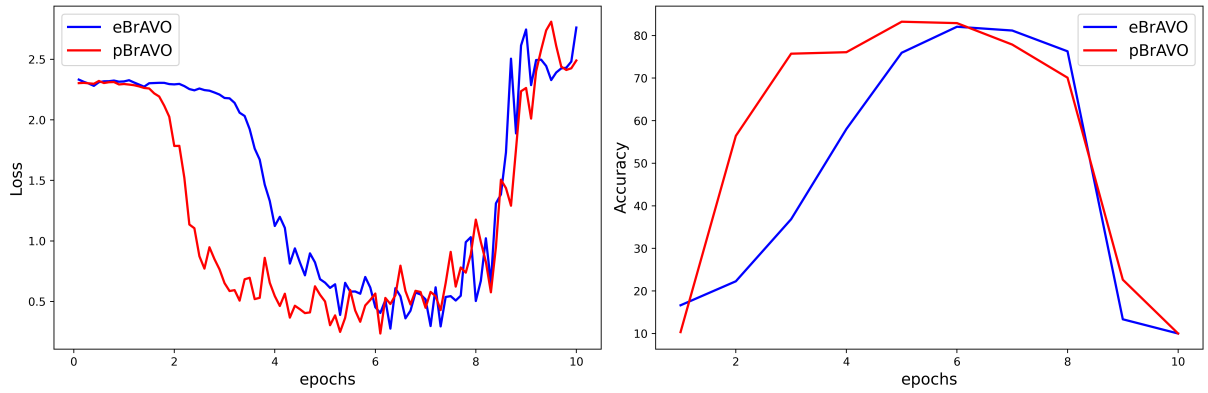


Figure 5.44: The convergence and loss of convergence for the BrAVO algorithms when implemented without momentum restarting and temporal looping, when applied to the Fashion–MNIST multi-label classification problem.

We tested our algorithm on another popular multi-label image classification problem based on the CIFAR-10 dataset [Krizhevsky, 2009]: *'The CIFAR-10 dataset consists of 60000 32×32 color images in 10 mutually exclusive classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck), with 6000 images per class'.*

We use `nn.CrossEntropyLoss()` as the loss function and a small Convolutional Neural Network in PyTorch similar to the LeNet-5 architecture from [Lecun et al., 1998]:

```
Layer (type)       Output Shape          Parameters
================================================================
Conv2d-1        [-1, 6, 28, 28]       456
Conv2d-2        [-1, 16, 10, 10]      2,416
Linear-3        [-1, 120]             48,120
Linear-4        [-1, 84]           10,164
Linear-5        [-1, 10]           850
================================================================
Total Number of Parameters: 62,006
```

The results are displayed in Figure 5.45.

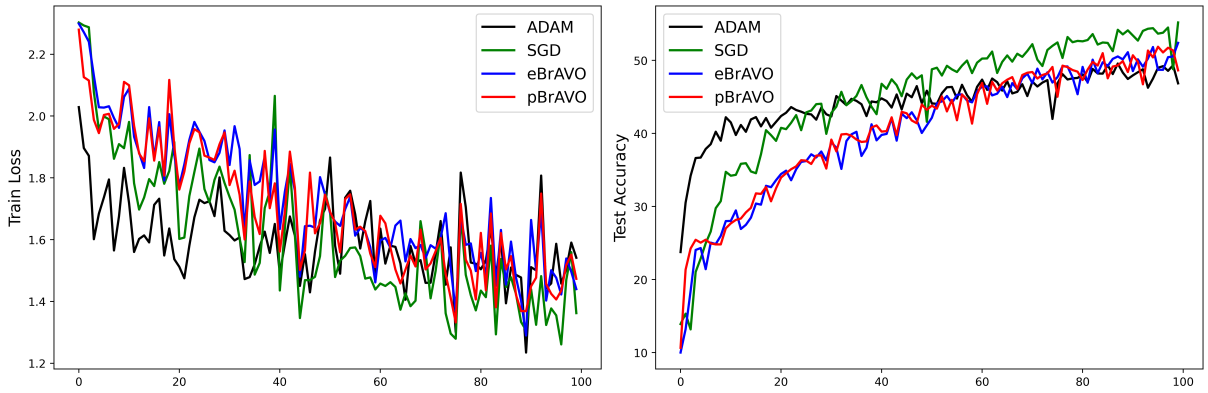


Figure 5.45: Evolution over 20 epochs of the loss function and accuracy of various algorithms when applied to the CIFAR–10 multi-label image classification problem.

Let us now consider the Natural Language Processing problem of constructing a multi-label text classifier which can provide suggestions for the most appropriate subject areas for arXiv papers based on their abstracts. The code and architecture used are based on the Keras tutorial [Paul and Rakshit, 2020]. An arXiv paper can belong to multiple categories, so the prediction task can be divided into a series of multiple binary classification problems, and we can use the Binary Cross Entropy loss. We will use the following neural network architecture:

```
model = keras.Sequential()
model.add(layers.Dense(units = 256, activation = 'relu'))
model.add(layers.Dense(units = 256, activation = 'relu'))
model.add(layers.Dense(units = lookup.vocabulary_size(), activation='sigmoid'))
```

The evolution of the Binary Cross Entropy loss on the training and validation sets is displayed in Figure 5.46. Although the Adam optimizer achieves the smallest loss on the training dataset, the resulting optimized model does not outperform the models generated by the other optimizers on the validation dataset. Its validation loss actually worsens as the epoch number increases (unlike for the other algorithms) which indicates that the optimized model might be suffering from overfitting.

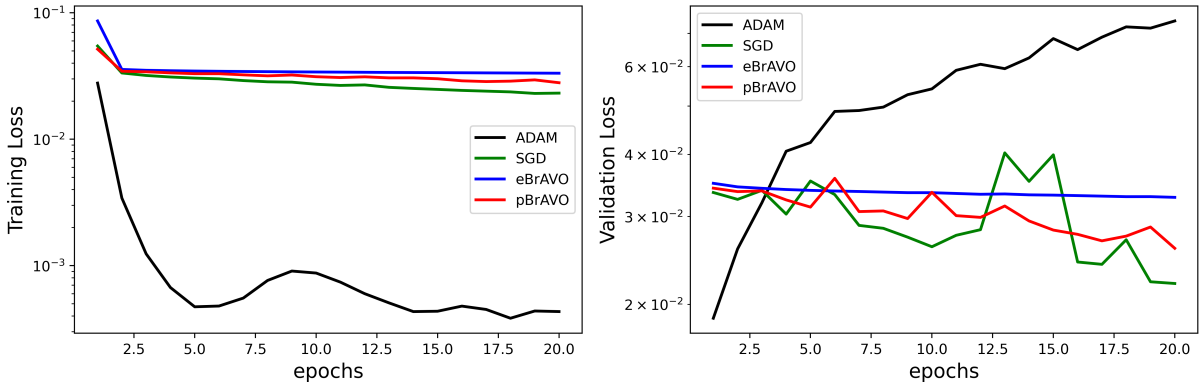


Figure 5.46: Evolution of the Binary Cross Entropy loss function on training and validation datasets for several algorithms, when applied to the Natural Language Processing problem of multi-label text classification of arXiv papers.

Next, we consider timeseries forecasting for weather prediction, based on the Keras tutorial [Attri et al., 2020]. We use the Mean Squared Error as the loss function and the following Long Short-Term Memory (LSTM) model (with 5,153 parameters):

```
inputs = layers.Input(shape=(inputs.shape[1], inputs.shape[2]))
lstm_out = layers.LSTM(32)(inputs)
outputs = layers.Dense(1)(lstm_out)
model = keras.Model(inputs=inputs, outputs=outputs)
```

The evolution of the mean squared error on the training and validation sets is displayed in Figure 5.47. We can see that the four different algorithms generate similar losses on the training and validation datasets.

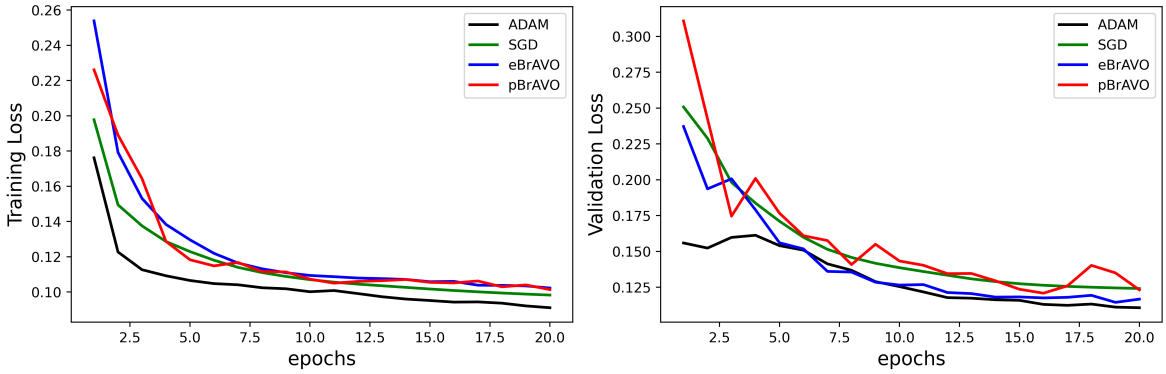


Figure 5.47: Evolution of the Mean Squared Error on training and validation datasets for several algorithms, when used to optimize a LSTM model for timeseries forecasting for weather prediction.

Then, we solved the following data fitting problem: given 500 data points from a noisy version of the function $10x|\cos 2x| + 10\exp(-\sin x)$ on the interval $[-2, 2]$, we wish to obtain a model which fits these data points as well as possible. We used the following neural network architecture (with 4,355 parameters) and loss function in TensorFlow:

```
model = keras.Sequential()
model.add(layers.Dense(units = 1, activation = 'linear', input_shape=[1]))
model.add(layers.Dense(units = 64, activation = 'relu'))
model.add(layers.Dense(units = 64, activation = 'relu'))
model.add(layers.Dense(units = 1, activation = 'linear'))
model.compile(loss='mse', optimizer=optimizer)
```

The results of this numerical experiment are displayed in Figures 5.48 and 5.49. We can see from Figure 5.48 that all the algorithms achieve very small mean squared error, and this observation is confirmed by Figure 5.49 which shows that the resulting models, plotted as blue curves, fit the green data points very well.

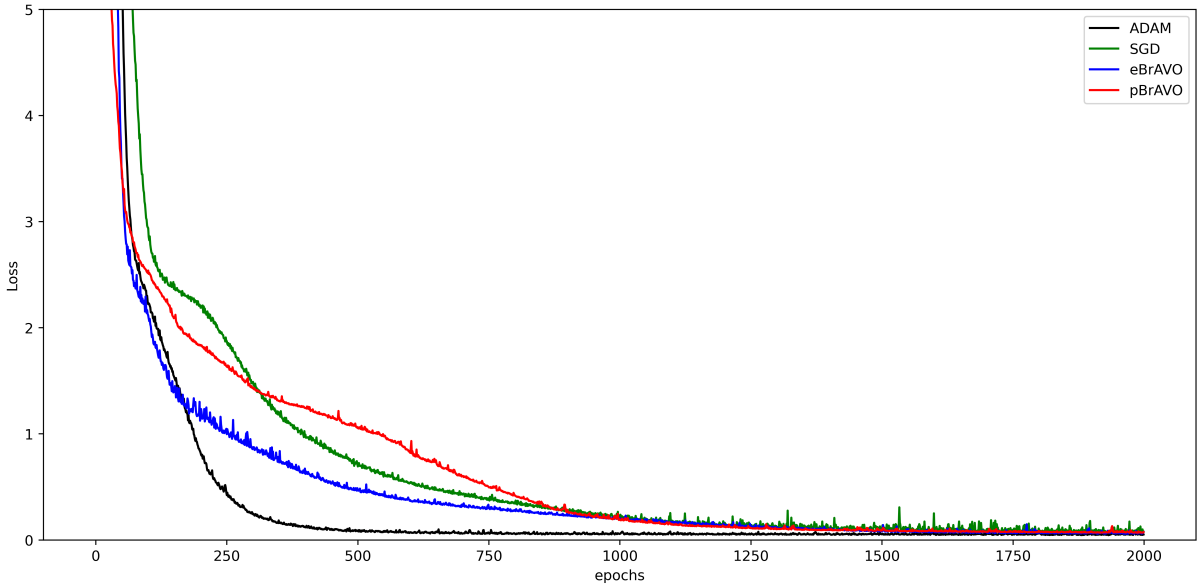


Figure 5.48: Evolution of the mean squared error for various algorithms, when applied to the problem of fitting a model to a set of 500 data points.

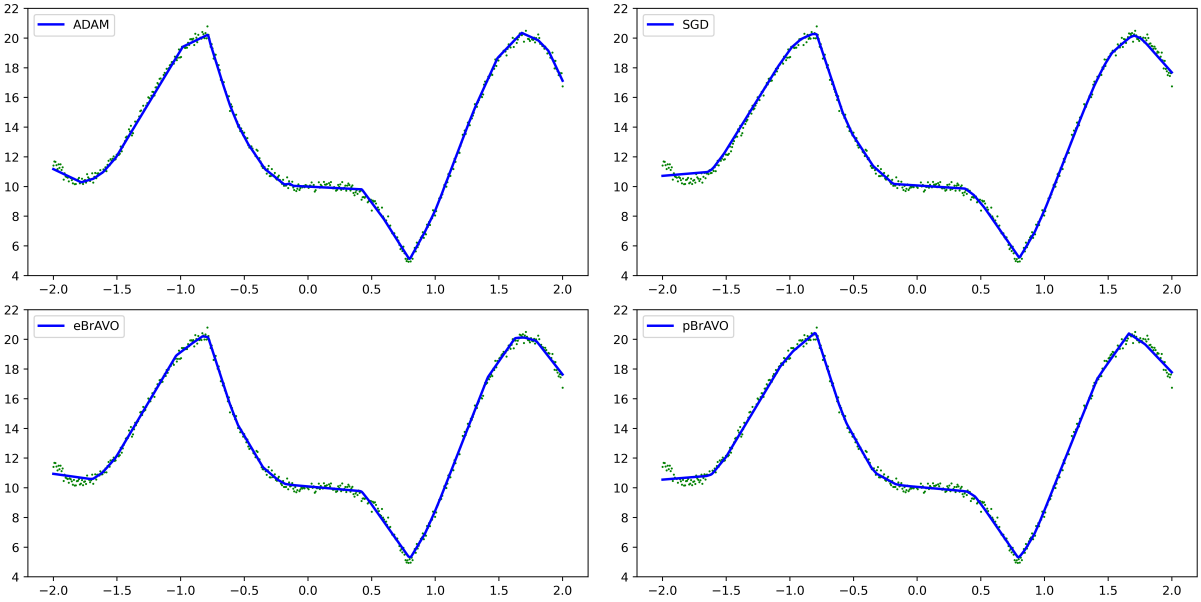


Figure 5.49: Models obtained after 2000 epochs using various algorithms to fit the 500 data points displayed in green.

Finally, we test our algorithms for dynamics learning and control. We consider the Hamiltonian-based neural ODE network from [Duong and Atanasov, 2021] (with 231,310 parameters), inspired by [Greydanus et al., 2019; Zhong et al., 2019], for dynamics learning and control on the rotation group SO(3), applied to a fully-actuated pendulum with dynamics given by

$$\ddot{\varphi} = -15 \sin \varphi + 3u, \tag{5.70}$$

where φ is the angle of the pendulum with respect to its vertically downward position and u is a scalar control input. The data is collected from an OpenAI Gym environment, provided by [Zhong et al., 2019].

We can see from Figure 5.50 that Adam and the BrAVO algorithms can achieve good training and test losses on this system identification problem, while we were unable to tune SGD to obtain a similar performance.

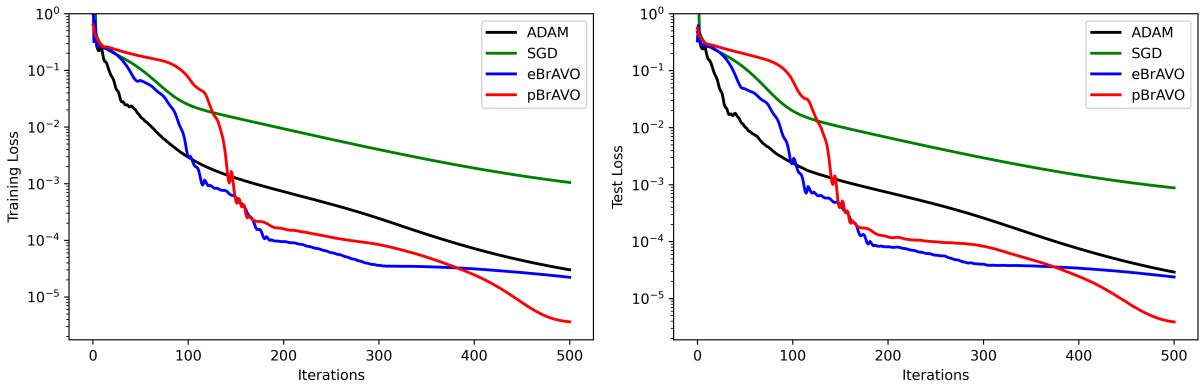


Figure 5.50: Evolution of the training and test losses for various algorithms, when learning the 231,310 parameters of a neural ODE network for dynamics learning.

Overall, we have demonstrated that the BrAVO algorithms can be used conveniently within the PyTorch and TensorFlow frameworks, and that they can perform very well on more challenging optimization problems arising in machine learning applications, with a variety of model architectures, loss functions, and applications. We reiterate that this was the main purpose of this section, and that it is not our intention to make a very careful computational comparison of the BrAVO algorithms with other optimization algorithms that are commonly used by the machine learning community.

A very careful computational comparison of optimization algorithms for machine learning applications is a much more ambitious goal which is beyond the scope of this dissertation. Such a comparison would be more meaningful once the current rudimentary implementation of the BrAVO algorithms within the PyTorch and TensorFlow frameworks has been highly-optimized, to take advantage of hardware architectures and highly-optimized PyTorch/TensorFlow operations. Aside from the quality of the implementation, other practical aspects of the algorithm could be investigated and improved further before carrying a careful comparison, for instance by looking into ways to boost the performance of the temporal looping technique or of the momentum restarting scheme.

An important advantage of our optimization methods is that they are derived by discretizing continuous-time dynamical systems. We might therefore be able to derive theoretical results about the algorithm by considering the associated continuous-time dynamical system and the discretization used. Furthermore, by considering the associated continuous-time dynamical system, we may be able to leverage numerous results from the theory of differential equations, dynamical systems, and geometric numerical integration. As a first example, in Section 5.4.2, we exploited perturbation theory for continuous-time dynamical systems to gain insight into the effect of the parameter C on the performance of the algorithms, which enabled us to improve tuning. As a second example, numerous ideas from the continuous-time theory of dynamical systems have been exploited in [Alvarez et al., 2002; Attouch et al., 2020, 2021, 2022] and in particular the notions of dissipation, of viscous and Hessian-driven damping, and of inertia, in second-order differential equations. As a last example, the notion of momentum itself is better understood as a physical property of a continuous-time dynamical system, and we can also gain a lot of insight into the mechanism allowing the accelerated convergence towards the minimizer by considering these dynamical systems. There might be many other ways in which the performance of our optimization algorithms can be improved by leveraging the associated continuous-time dynamical system.

5.10 Remark for Riemannian Optimization

Preliminary experiments suggest that the practical considerations discussed in this chapter also apply to the framework for accelerated optimization on Riemannian manifolds presented in Chapter 4. The Riemannian optimization algorithms presented in Chapter 4 respond in a similar way to changes in their parameters and can therefore be tuned just like their normed vector space analogues. The computational benefits of momentum restarting schemes and temporal looping uncovered in Sections 5.4.1 and 5.8 extend to the Riemannian manifold setting, although the search for good momentum restarting criteria and robust numerical instability criteria becomes more challenging and most likely manifold-dependent. These research directions could be explored further and more carefully to improve the computational efficiency and stability of symplectic accelerated optimization algorithms on Riemannian manifolds.

Conclusion and Future Directions

In this chapter, we have discussed practical considerations which can significantly boost the computational performance and simplify the tuning of symplectic accelerated optimization algorithms that are constructed by integrating Lagrangian and Hamiltonian systems coming from the variational framework for optimization that was introduced in [Wibisono et al., 2016].

We have showed in particular that momentum restarting schemes can lead to a significant gain in computational efficiency and robustness by reducing the undesirable effect of oscillations, and that a temporal looping strategy helps to avoid instability issues caused by numerical precision, and does not impair the computational performance of the algorithms in general. We also observed that time-adaptivity and the choice of symplectic integrator hardly make a difference once a momentum restarting scheme is incorporated in the optimization algorithms. This observation, along with other numerical experiments designed to study the effects of the different parameters, has provided insights that allowed to inform and ease the tuning process by simplifying the algorithms and by reducing the number of parameters to tune.

Overall, we have designed symplectic accelerated optimization algorithms whose computational efficiency and stability have been improved using temporal looping and momentum restarting, and which are now more user-friendly. We tested these algorithms on machine learning optimization problems with numerous different model architectures, loss functions, and applications, and saw that they can achieve results with accuracy and computational efficiency that are comparable to those of Adam.

Preliminary experiments with the algorithms from Chapter 4 suggest that the practical considerations and the benefits of momentum restarting and temporal looping discussed in this chapter extend to the Riemannian manifold setting. This could be explored further, and could lead to symplectic accelerated optimization algorithms on Riemannian manifolds with improved computational efficiency and stability.

The temporal looping technique could also be improved by carefully designing different numerical instability criteria. Instead of temporal looping strategies, one could also try to implement different popular techniques in machine learning such as decaying learning rates via a learning rate scheduler, or to progressively increase the batch size [Smith et al., 2018]. The current implementation of the algorithms within the PyTorch and TensorFlow frameworks is rather rudimentary, and can certainly be improved to reduce computational time by taking advantage of hardware architectures and highly-optimized PyTorch/TensorFlow operations. With the same objective in mind, one could also replace the gradient scheme for momentum restarting by the function scheme if the latter can be implemented more efficiently.

Once the algorithms have been improved further, possibly leveraging the theory of continuous-time dynamical systems, and once the implementation of the algorithms has been highly-optimized, it would be very interesting to perform a very careful computational comparison with other popular algorithms on many different types of problems to see whether the BrAVO algorithms can outperform the state-of-the-art algorithms on certain classes of machine learning problems.

➢ Chapter 5 contains original material from

① "Practical Perspectives on Symplectic Accelerated Optimization" by V. Duruisseaux and M. Leok. *Optimization Methods and Software*, Vol.38, Issue 6, pages 1230-1268, 2023

The dissertation author was the primary investigator and author of this paper.

Part II Conclusion

Most artificial intelligence and machine learning algorithms rely heavily on the capability of finding optimizers of high-dimensional objective functions. Improvements in the efficiency and reliability of optimization algorithms can therefore lead directly to huge savings in computational and energy resources.

A nontraditional approach to optimization is to replace the problem of minimizing the given objective function by the problem of numerically evolving dynamical systems governed by appropriately defined differential equations. In this dissertation, we have constructed structure-preserving discretizations of a carefully designed family of time-dependent Bregman Lagrangian and Hamiltonian systems whose trajectories converge to the minimizer of the objective function.

After conducting a detailed computational study to select symplectic discretizations for the Bregman dynamics, ease tuning, boost computational performance, and improve stability and robustness, we obtained symplectic accelerated optimization algorithms, the BrAVO algorithms, which can be used for machine learning applications. An important advantage of our methods is that they have an associated well-structured continuous-time dynamical systems, yielding in particular better interpretability. By considering these continuous-time dynamics, we may be able to leverage the theory of dynamical systems and differential equations to improve the performance of the optimization algorithms or derive theoretical guarantees.

This research direction is still in its early stages, and there is still a lot of potential for improvement, both in the design of the differential equations discretizations and in their implementation to leverage the full power of modern hardware and parallel computing architectures.

Since many optimization problems are better formulated as optimization problems on Riemannian manifolds, we have generalized all the theory to the Riemannian setting. We replaced the problem of optimizing a function on Riemannian manifolds by the problem of simulating the dynamics of a family of time-dependent Bregman Lagrangian and Hamiltonian systems evolving on Riemannian manifolds. However, the task of integrating geometric mechanics on Riemannian manifolds is significantly more challenging and still lacks fully satisfying solutions. In this dissertation, we proposed a few different numerical approaches to design geometric integrators for the Riemannian Bregman family of dynamics, but their performance is still limited and manifold dependent, in particular by the capacity to perform standard Riemannian operations on the Riemannian manifold of interest. Better characterizations of Riemannian manifolds of interest and better computational methods for the standard Riemannian operations on those manifolds would certainly foster improvements in the computational efficiency of Riemannian optimization algorithms. In addition, many practical aspects which could improve the performance of our algorithms. For instance, preliminary experiments suggest that the practical considerations discussed in Chapter 5 which led to significant improvements on normed vector spaces in terms of computational performance, stability, and robustness, also extend to the Riemannian manifold setting.

Another important direction for future work would be to derive analytical guarantees for the convergence rates of the resulting discrete-time algorithms. This will most likely be a very challenging task, especially in the Riemannian manifold where all the usual vector space operations and objects have to be replaced by their much more convoluted Riemannian generalizations. Note however that in the normed vector space setting, some theoretical results have been derived for discrete-time optimization algorithms based on discretizations of dynamical systems, and in particular based on simpler special cases of the Bregman family of dynamics. These could inform and inspire future attempts at deriving more general theoretical guarantees for discretizations of the Bregman family of dynamics, or at least for special cases that are of greater interest due to their computational advantages and superiority.

Part III

Structure-Preserving Dynamics Learning

Introduction

Dynamical systems evolve according to the laws of physics, which can usually be described using differential equations. By solving these differential equations, it is possible to predict the future states of the dynamical systems. Identifying accurate and efficient dynamic models based on observed trajectories is thus critical not only for predicting future behavior, but also for designing control laws that ensure desirable properties such as safety, stability, and generalization to different operational conditions. We will consider the problem of learning dynamics: given a dataset of observed trajectories followed by a dynamical system, we wish to infer the dynamical law responsible for these trajectories and use it to predict the evolution of the system from different initial states. We are also interested in the surrogate modeling problem: the underlying dynamical system is known, but traditional simulations are either too slow or expensive for some optimization task. This problem can be addressed by learning a less expensive surrogate for the simulations.

Models obtained from first principles are extensively used in practice, across science and engineering. Unfortunately, due to incomplete knowledge, these models based on physical laws tend to over-simplify or incorrectly describe the underlying structure of the dynamical systems, which usually leads to high prediction errors that cannot be corrected by optimizing over the few parameters in the models.

Deep learning techniques and architectures can provide very expressive models for function approximation, and have proven very effective in numerous contexts [Jin et al., 2020; Burby et al., 2020; Karniadakis et al., 2021]. Unfortunately, standard non-structure-preserving neural networks struggle to learn the symmetries and conservation laws underlying dynamical systems, and as a result do not generalize well. Indeed, they tend to prefer certain representations of the dynamics where the symmetries and conservation laws of the system are not exactly enforced. As a result, these models do not

generalize well as they are often not capable of producing physically plausible results when applied to new unseen states. Deep learning models capable of learning and generalizing dynamics effectively are typically over-parameterized, and as a consequence tend to have high variance and can be very difficult to interpret [Willard et al., 2020]. Also, training these models usually requires large datasets and a long computational time, which makes them prohibitively expensive for many applications such as robotics.

A recent research direction has been considering a hybrid approach, which combines knowledge of physics laws and deep learning architectures [Burby et al., 2020; Jin et al., 2020; Lei et al., 2020; Qin, 2020; Karniadakis et al., 2021]. The idea is to encode physics laws and the geometric properties of the underlying systems in the design of the neural networks or in the learning process. Available physics prior knowledge can be used to construct physics-constrained neural networks with improved design and efficiency and a better generalization capacity, which take advantage of the function approximation power of neural networks to deal with incomplete knowledge.

One of the most important examples of geometric structure underlying dynamical systems is the symplecticity of the flows of Hamiltonian (and Lagrangian) systems. We have already seen numerous times in this dissertation that preserving the symplecticity of a Hamiltonian system when constructing a discrete approximation of its flow map ensures the preservation of many aspects of the dynamical system such as energy conservation, and leads to physically well-behaved discrete solutions. It is therefore important to have structure-preserving neural network architectures which can learn symplectic maps and ensure that the learnt surrogate map preserves symplecticity. Furthermore, many mechanical and robotic systems can be modeled as (controlled) Hamiltonian or Lagrangian systems, so the problem of learning Hamiltonian and Lagrangian dynamics is of considerable importance. Many physics-informed machine learning approaches have recently been proposed to learn Hamiltonian and Lagrangian dynamics and symplectic maps [Lutter et al., 2019b; Greydanus et al., 2019; Bertalan et al., 2019; Zhong et al., 2019; Jin et al., 2020; Burby et al., 2020; Sæmundsson et al., 2020; Chen et al., 2020; Cranmer et al., 2020; Zhong et al., 2020, 2021; Havens and Chowdhary, 2021; David and Méhats, 2021; Rath et al., 2021; Chen et al., 2021; Offen and Ober-Blöbaum, 2022; Santos et al., 2022; Valperga et al., 2022; Mathiesen et al., 2022].

In the next two chapters, we will build upon some of these physics-informed machine learning approaches to learn special classes of Hamiltonian or Lagrangian dynamics while preserving the symplecticity and additional geometric properties possessed by these special Hamiltonian or Lagrangian systems.

More precisely, inspired by (Forced) Variational Integrator Networks ((F)VINs) [Sæmundsson et al., 2020; Havens and Chowdhary, 2021], we will introduce a new structure-preserving deep learning architecture in Chapter 6, the Lie group Forced Variational Integrator Network (LieFVIN) [Duruisseaux et al., 2023c], capable of learning surrogate maps for the flow maps of controlled Lagrangian or Hamiltonian dynamics evolving on Lie groups, either from position-velocity or position-only data. By design, LieFVINs preserve both the Lie group structure on which the dynamics evolve and the symplectic structure underlying the controlled Hamiltonian or Lagrangian systems of interest.

In Chapter 7, we construct a novel structure-preserving neural network architecture, to approximate nearly-periodic symplectic maps. Nearly-periodic symplectic maps are the discrete-time analogues of nearly-periodic Hamiltonian systems, where a dynamical system with parameter ε is said to be nearly-periodic if all its trajectories are periodic with nowhere-vanishing angular frequency as ε approaches 0. Nearly-periodic symplectic maps possess approximately conserved discrete-time quantities, called adiabatic invariants, and our novel neural network architecture, which we call symplectic gyroceptron [Duruisseaux et al., 2023a], ensures that the resulting surrogate map is nearly-periodic and symplectic, and that it gives rise to a discrete-time adiabatic invariant and a long-time stability as a consequence. As mentioned earlier in this introduction, many physics-informed and structure-preserving machine learning approaches have recently been proposed to learn symplectic maps. In particular, Hénon Neural Networks (HénonNets) [Burby et al., 2020] can approximate arbitrary well any symplectic map via compositions of simple yet expressive elementary symplectic mappings called Hénon-like mappings. In the numerical experiments conducted in Chapter 7, HénonNets will be our preferred choice of symplectic map approximator to use as building block in our framework for approximation of nearly-periodic symplectic maps, although some of the other approaches listed above for approximating symplectic mappings can be used within our framework as well.

6 Lie Group Forced Variational Integrator Networks for Learning and Control of Robot Systems

6.1 Introduction

Incorporating prior knowledge of physics laws and structural properties of dynamical systems into the design of deep learning architectures has proven to be a powerful technique for improving their computational efficiency and generalization capacity. An accurate model of the dynamics of a control system is important, not only for predicting future behavior, but also for designing control laws that ensure desirable properties such as safety, stability, and generalization to different operational conditions. Autonomous mobile robots, including wheeled, aerial, and underwater vehicles, can be modeled as controlled Lagrangian or Hamiltonian rigid-body systems evolving on matrix Lie groups.

In this chapter, we introduce and test a novel structure-preserving deep learning architecture, the Lie group Forced Variational Integrator Network (**LieFVIN**), to learn controlled Lagrangian or Hamiltonian dynamics on Lie groups, either from position-velocity or position-only data. By design, LieFVINs preserve both the Lie group structure on which the dynamics evolve and the symplectic structure underlying the systems of interest. The proposed architecture learns surrogate discrete-time flow maps allowing accurate and fast prediction without numerical integration, neural-ODE, or adjoint techniques, which are needed for vector fields. Furthermore, the learnt discrete-time dynamics can be used with computationally scalable discrete-time (optimal) control strategies.

As discussed earlier, it is important to have structure-preserving architectures which can learn symplectic flow maps and ensure that the learnt surrogate maps are symplectic, and many physics-informed approaches have recently been proposed to learn Hamiltonian dynamics and symplectic maps. Our physics-informed strategy, inspired by (Forced) Variational Integrator Networks ((F)VINs) [Sæmundsson et al., 2020; Havens and Chowdhary, 2021], differs from most of these approaches by learning a discrete-time symplectic approximation to the flow map of the dynamical system, instead of learning the vector field for the continuous-time dynamics. This allows fast prediction for simulation, planning and control without the need to integrate differential equations or use neural ODEs and adjoint techniques. Additionally, the learnt discrete-time dynamics can be combined with computationally scalable discrete-time control strategies.

The novelty of our approach with respect to (F)VINs resides in the enforcement not only of the preservation of symplecticity but also of the Lie group structure when learning a surrogate map for a controlled Lagrangian system which evolves on a Lie group. This is achieved by working in Lie group coordinates instead of Euclidean coordinates, by matching the training data to a parameterized forced Lie group variational integrator which evolves intrinsically on the Lie group. More specifically, we extend the discrete-time Euclidean formulation of FVINs with control from [Havens and Chowdhary, 2021] to Lie groups in a structure-preserving way, which is particularly relevant when considering robot systems (e.g., wheeled, aerial, and underwater vehicles) since they can often be modeled as controlled Lagrangian rigid-body systems evolving on Lie groups.

Given a learnt dynamical system, it is often desirable to control its behavior to achieve stabilization, tracking, or other control objectives. Control designs for continuous-time Hamiltonian systems rely on the Hamiltonian structure [Lutter et al., 2019a; Zhong et al., 2019; Duong and Atanasov, 2021, 2022]. Since the Hamiltonian captures the system energy, control techniques for stabilization inject additional energy into the system via the control input to ensure that the minimum of the total energy is at a desired equilibrium. For fully-actuated Hamiltonian systems, it is sufficient to shape the potential energy only using energy-shaping and damping-injection (ES-DI) [Van Der Schaft and Jeltsema, 2014]. For under-actuated systems, both the kinetic and potential energies are shaped, e.g.,

via interconnection and damping assignment passivity-based control (IDA-PBC) [Ortega et al., 2002; Van Der Schaft and Jeltsema, 2014; Acosta et al., 2014; Cieza and Reger, 2019]. The most widely used control approach for discrete-time dynamics is based on Model Predictive Control (MPC) [Borrelli et al., 2017; Grüne and Pannek, 2017]. MPC techniques determine an open-loop control sequence that solves a finite-horizon optimal control problem, apply the first few control inputs, and repeat the process. A key result in MPC is that an appropriate choice of terminal cost and terminal constraints in the sequence of finite-horizon problems can guarantee recursive feasibility and asymptotic optimality with respect to the infinite-horizon cost [Borrelli et al., 2017]. The ability to learn a structure-preserving discrete-time model of a dynamics system, also allows employing MPC techniques for optimal control of the learnt system dynamics.

6.2 Problem Statement

We consider the problem of learning controlled Lagrangian dynamics. Given a position-velocity dataset of trajectories for a Lagrangian system, we wish to infer the update map that generates these trajectories, while preserving the symplectic structure underlying the dynamical system and constraining the updates to the Lie group on which it evolves. More precisely, we consider the following problem.

Problem 6.1. *Given a dataset of N position-velocity updates,*

$$\left\{\left(q_0^{(i)}, \dot{q}_0^{(i)}, u_0^{(i)}\right) \mapsto \left(q_1^{(i)}, \dot{q}_1^{(i)}\right)\right\}_{i=1}^{N},$$

for a controlled Lagrangian dynamical system evolving on a Lie group $\mathcal{Q}$, we wish to find a symplectic mapping $\Psi : T\mathcal{Q} \times \mathcal{U} \to T\mathcal{Q}$ which minimizes

$$\sum_{i=1}^{N} \mathscr{D}_{T\mathcal{Q}}\left(\left(q_1^{(i)}, \dot{q}_1^{(i)}\right), \Psi\left(q_0^{(i)}, \dot{q}_0^{(i)}, u_0^{(i)}\right)\right), \tag{6.1}$$

where $\mathscr{D}_{T\mathcal{Q}}$ is a distance metric on the tangent bundle $T\mathcal{Q}$ of $\mathcal{Q}$.

As discussed in Section 2.1, the Legendre transform is diffeomorphic for most mechanical systems and thus the Lagrangian and Hamiltonian formulations are equivalent. The approaches presented in this chapter are based on the Lagrangian formulation, but also apply to the equivalent Hamiltonian systems whenever they are well-defined.

6.3 LieFVINs: Lie Group Forced Variational Integrators Networks

To solve Problem 6.1, we will introduce Lie group Forced Variational Integrators Networks (**LieFVINs**). Our main idea is to parametrize the updates of a forced Lie group variational integrator and match them with observed updates. We focus on specific forced SO(3) and SE(3) variational integrators, but the general strategy that will be presented extends to any Lie group forced variational integrator.

6.3.1 Rigid-body kinematics on SE(3)

We first present a brief introduction to rigid-body kinematics on SE(3), mostly extracted from Chapters 2, 6, 7 of [Lee et al., 2017].

A rigid body is an idealization of a real mechanical system, defined as a collection of material particles such that the relative distance between any two particles in the body does not change (i.e, the body does not deform). The configuration of a rigid body is a representation of its position and attitude in 3-dimensional space. The kinematics of a rigid body describe how its configuration changes under the influence of linear and angular velocities. Defining the configuration of the rigid body is of the utmost importance for rigid-body kinematics, and depends on the constraints imposed on the rigid-body motion.

Rotational Rigid-Body Motion

If a rigid body has fixed position but is allowed to rotate arbitrarily in $\mathbb{R}^3$, its configuration can be represented by a rotation matrix. Hence, the manifold of rotation matrices, SO(3), is the configuration manifold for rigid-body rotational motion. Since the dimension of SO(3) is three, rigid-body rotations have three degrees of freedom.

We use two Euclidean frames: an arbitrary reference frame and another frame fixed to the rigid body which rotates with it (with origin selected at the center of mass of the rigid body). A rotation matrix $R \in \mathrm{SO}(3)$ is a linear transformation on $\mathbb{R}^3$ between the body-fixed and reference frames:

- if $v \in \mathbb{R}^3$ represents a vector in the body frame, then $Rv \in \mathbb{R}^3$ represents the same vector in the reference frame,
- if $v \in \mathbb{R}^3$ represents a vector in the reference frame, then $R^\top v \in \mathbb{R}^3$ represents the same vector in the body frame.

We can describe the rotation of the rigid body through the rotation of the body-fixed frame: the configuration of a rotating rigid body is the linear transformation that relates the representation of a vector in the body-fixed frame to its representation in the reference frame.

Suppose that $R(t) \in \mathrm{SO}(3)$ represents the rotational motion of a rigid body. Differentiating the orthogonality condition $R^\top R = \mathbb{I}_3$, we get

$$\dot{R}^\top R = -R^\top \dot{R} \tag{6.2}$$

which implies that $R^\top \dot{R}$ remains skew-symmetric at all time. Therefore, there exists a skew-symmetric matrix $\xi(t) \in \mathfrak{so}(3)$ such that $R^\top \dot{R} = \xi$, from which we can obtain the rotational kinematics:

$$\dot{R} = R\xi. \tag{6.3}$$

Using the isomorphism between the Lie algebra $\mathfrak{so}(3)$ and $\mathbb{R}^3$ given by $\xi = S(\omega)$ for $\omega \in \mathbb{R}^3$ and $\xi \in \mathfrak{so}(3)$, we can rewrite the rotational kinematics as

$$\dot{R} = RS(\omega), \tag{6.4}$$

where $\omega \in \mathbb{R}^3$ is referred to as the angular velocity vector of the rigid body expressed in the body frame. Thus, the rotational kinematics describe the rate of change $\dot{R}$ of the configuration in terms of the angular velocity $\omega \in \mathbb{R}^3$ represented in the body frame.

General Rigid-Body Motion

General rigid-body motion can be described by a combination of rotations and translations.

As before, we use two inertial frames: a first arbitrary reference frame and another frame fixed to the rigid body which translates and rotates with the rigid body (with origin usually selected at the center of mass of the rigid body). The translational configuration of the rigid body characterizes the motion of the body-fixed frame origin and can be selected to lie in the configuration manifold $\mathbb{R}^3$.

The configuration manifold for a rigid-body that is simultaneously translating and rotating can be selected as the semidirect product of $\mathbb{R}^3$ and $\mathrm{SO}(3)$. Therefore, we can represent the configuration via $(R, x) \in \mathrm{SE}(3)$ in the sense that $R \in \mathrm{SO}(3)$ is the orientation and $x \in \mathbb{R}^3$ is the position of the body-fixed frame in the reference frame. Consequently, the Lie group $\mathrm{SE}(3)$ can be viewed as the configuration manifold for general rigid-body motion.

As before, the rotational kinematic equations describe the rate of change $\dot{R}$ of the configuration in terms of the angular velocity vector $\omega \in \mathbb{R}^3$ of the rigid body represented in the body-fixed frame:

$$\dot{R} = RS(\omega). \tag{6.5}$$

Now, the translational velocity vector $v \in \mathbb{R}^3$ of the rigid body (i.e., of the origin of the body-fixed frame) is the time derivative of the position vector from the origin of the reference frame to the origin of the body-fixed frame. In the reference frame, the translational velocity vector $\dot{x} \in \mathbb{R}^3$ of the rigid body is

$$\dot{x} = Rv. \tag{6.6}$$

These are referred to as the translational kinematics of the rigid body. Altogether, we obtain the kinematics for general rigid-body motion:

$$\dot{R} = RS(\omega), \qquad \dot{x} = Rv. \tag{6.7}$$

6.3.2 Forced Variational Integrator on SO(3) and SE(3)

On SE(3), $q = (x, R)$ and $\dot{q} = (v, \omega)$ where x is position, R is orientation, v is velocity, and ω is angular velocity. A Lagrangian on SE(3) is given by

$$L(x, R, v, \omega) = \frac{1}{2} v^\top m v + \frac{1}{2} \omega^\top J \omega - U(x, R), \tag{6.8}$$

where m is mass, $J \in \mathbb{R}^{3\times 3}$ is a symmetric positive-definite inertia matrix, and U is a potential energy function.

Consider the continuous-time kinematics equation $\dot{R} = RS(\omega)$, with constant $\omega(t) \equiv \omega_k$ for a short period of time $t \in [t_k, t_{k+1})$ where $t_{k+1} = t_k + h$. Then,

$$R(t_{k+1}) = R(t_k) \exp(hS(\omega_k)). \tag{6.9}$$

Thus, with $R_k := R(t_k)$, $R_{k+1} := R(t_{k+1})$ and $Z_k := \exp(hS(\omega_k))$, we obtain $R_{k+1} = R_k Z_k$ and for sufficiently small h, we have $Z_k \approx \mathbb{I}_3 + hS(\omega_k)$. With $(x_k, R_k) \in \mathrm{SE}(3)$, the discrete SE(3) kinematic equations are given by

$$R_{k+1} = R_k Z_k, \qquad \text{and} \qquad x_{k+1} = x_k + R_k y_k, \tag{6.10}$$

where $(y_k, Z_k) \in \mathrm{SE}(3)$, which ensures that the sequence $\{(x_k, R_k)\}_k$ remains on SE(3).

Using the approximation $S(\omega_k) \approx \frac{1}{h}(Z_k - \mathbb{I}_3)$, we choose the discrete Lagrangian

$$\begin{aligned} L_d(x_k, R_k, y_k, Z_k) \;=\; & \frac{m}{2h} y_k^\top y_k + \frac{1}{h} \mathrm{Trace}\left([\mathbb{I}_3 - Z_k] J_d\right) \\ & - (1-\alpha) h U(x_k, R_k) - \alpha h U(x_k + R_k y_k, R_k Z_k), \end{aligned} \tag{6.11}$$

where $\alpha \in [0, 1]$ and $J_d = \frac{1}{2}\mathrm{Trace}(J)\mathbb{I}_3 - J$.

We will use R and x superscripts for $f_{d_k}^{\pm}$ to denote the R and x components of $f_{d_k}^{\pm}$, and we define U_k and ξ_k via

$$U_k = U(x_k, R_k) \qquad \text{and} \qquad S(\xi_k) = \frac{\partial U_k}{\partial R_k}^\top R_k - R_k^\top \frac{\partial U_k}{\partial R_k}. \tag{6.12}$$

In Appendix A.8, we show that the forced discrete Euler–Lagrange equations corresponding to the discrete Lagrangian in (6.11) and discrete control forces $f_{d_k}^{\pm} \equiv f_d^{\pm}(x_k, R_k, u_k)$ can be written in Hamiltonian form, using

$$\pi_k = J\omega_k, \qquad \text{and} \qquad \gamma_k = mv_k, \tag{6.13}$$

as

$$hS(\pi_k) + hS(f_{d_k}^{R-}) + (1-\alpha)h^2 S(\xi_k) = Z_k J_d - J_d Z_k^\top, \tag{6.14}$$

$$R_{k+1} = R_k Z_k, \tag{6.15}$$

$$\pi_{k+1} = Z_k^\top \pi_k + (1-\alpha)hZ_k^\top \xi_k + \alpha h \xi_{k+1} + Z_k^\top f_{d_k}^{R-} + f_{d_k}^{R+}, \tag{6.16}$$

$$x_{k+1} = x_k + \frac{h}{m}\gamma_k - (1-\alpha)\frac{h^2}{m}\frac{\partial U_k}{\partial x_k} - \frac{h}{m}R_k f_{d_k}^{x-}, \tag{6.17}$$

$$\gamma_{k+1} = \gamma_k - (1-\alpha)h\frac{\partial U_k}{\partial x_k} - \alpha h \frac{\partial U_{k+1}}{\partial x_{k+1}} + R_k f_{d_k}^{x-} + R_{k+1} f_{d_k}^{x+}. \tag{6.18}$$

Given $(x_k, R_k, \gamma_k, \pi_k, u_k)$, we first solve equation (6.14) which is of the form

$$S(a) = ZJ_d - J_d Z^\top, \tag{6.19}$$

as outlined in Remark 6.1, and then get $R_{k+1} = R_k Z_k$. We then obtain π_{k+1}, x_{k+1} and γ_{k+1} from equations (6.16)-(6.18). The discrete equations of motion can be rewritten as an update from $(x_k, R_k, v_k, \omega_k, u_k)$ to $(x_{k+1}, R_{k+1}, v_{k+1}, \omega_{k+1})$ by using $\pi_k = J\omega_k$ and $\gamma_k = mv_k$.

Remark 6.1 (Solving $S(a) = ZJ_d - J_d Z^\top$)**.** *Using the* ***Cayley transform***

$$Z = \mathrm{Cay}(\mathfrak{z}) \equiv (\mathbb{I}_3 + S(\mathfrak{z}))(\mathbb{I}_3 - S(\mathfrak{z}))^{-1} = \frac{1}{1+\|\mathfrak{z}\|_2^2}\left((1-\|\mathfrak{z}\|_2^2)\mathbb{I}_3 + 2S(\mathfrak{z}) + 2\mathfrak{z}\mathfrak{z}^\top\right), \tag{6.20}$$

the equation $S(a) = ZJ_d - J_d Z^\top$ can be converted into an equivalent vector equation

$$\phi(\mathfrak{z}) \;\equiv\; a + a \times \mathfrak{z} + \mathfrak{z}(a^\top \mathfrak{z}) - 2J\mathfrak{z} \;=\; 0, \qquad \mathfrak{z} \in \mathbb{R}^3, \tag{6.21}$$

as shown in Appendix A.9. The solution $Z = \mathrm{Cay}(\mathfrak{z})$ can be obtained after solving this vector equation for $\mathfrak{z}$ by using (typically 2 or 3 steps of) Newton's method:

$$\mathfrak{z}^{(n+1)} = \mathfrak{z}^{(n)} - \left[\nabla\phi(\mathfrak{z}^{(n)})\right]^{-1}\phi(\mathfrak{z}^{(n)}), \qquad \nabla\phi(\mathfrak{z}) = S(a) + (a^\top \mathfrak{z})\mathbb{I}_3 + \mathfrak{z}a^\top - 2J. \tag{6.22}$$

6.3.3 SE(3) Lie Group Forced Variational Integrator Networks

We now describe the construction of **Lie group Forced Variational Integrator Networks** (**LieFVINs**), for the forced variational integrator on SE(3) presented earlier in Section 6.3.2. The idea is to parametrize the updates of the integrator and match them with observed updates. Here, we consider the case where position-velocity data is available, in which case the LieFVIN is based on equations (6.14)-(6.18). The case where only position data is available will be presented more briefly in Section 6.5.

We parametrize m, $f_d^{\pm}$ and U as neural networks. The inertia J is a symmetric positive-definite matrix-valued function of (x, R) constructed via a Cholesky decomposition $J = LL^\top$ for a lower-triangular matrix L implemented as a neural network. Given J, we also obtain $J_d = \frac{1}{2}\text{Trace}(J)\mathbb{I}_3 - J$. To deal with the implicit nature of equation (6.14), we propose two algorithms, based either on an explicit iterative solver or by penalizing deviations away from equation (6.14):

Algorithm 10. Given position-velocity updates $\{(x_0, R_0, v_0, \omega_0, u_0) \mapsto (x_1, R_1, v_1, \omega_1)\}$, minimize discrepancies between the observed $(x_1, R_1, v_1, \omega_1)$ quadruples and the predicted $(\tilde{x}_1, \tilde{R}_1, \tilde{v}_1, \tilde{\omega}_1)$ quadruples, obtained as follows: for each $(x_0, R_0, v_0, \omega_0, u_0)$ data tuple,

1. Get $f_{d_0}^{R\pm}$ and $f_{d_0}^{x\pm}$ from (x_0, R_0, u_0)
2. Get ξ_0 from $S(\xi_0) = \frac{\partial U_0}{\partial R_0}^\top R_0 - R_0^\top \frac{\partial U_0}{\partial R_0}$
3. Get $Z_0 = \text{Cay}(\mathcal{Z})$ where $\mathcal{Z}$ is obtained using a few steps of Newton's method to solve the vector equation (6.21) equivalent to the equation
$$hS(J\omega_0) + hS(f_{d_0}^{R-}) + (1-\alpha)h^2 S(\xi_0) \;=\; ZJ_d - J_d Z^\top$$
4. Compute $\tilde{R}_1 = R_0 Z_0$
5. Get ξ_1 from $S(\xi_1) = \frac{\partial U_1}{\partial \tilde{R}_1}^\top \tilde{R}_1 - \tilde{R}_1^\top \frac{\partial U_1}{\partial \tilde{R}_1}$
6. Get $\tilde{\omega}_1$ from $J\tilde{\omega}_1 = Z_0^\top J\omega_0 + (1-\alpha)hZ_0^\top \xi_0 + \alpha h\xi_1 + Z_0^\top f_{d_0}^{R-} + f_{d_0}^{R+}$
7. Compute $\tilde{x}_1 = x_0 + hv_0 - (1-\alpha)h^2 \frac{\partial U_0}{\partial x_0} - \frac{h}{m} R_0 f_{d_0}^{x-}$
8. Compute $\tilde{v}_1 = v_0 + \frac{1}{m}\left[-(1-\alpha)h^2 \frac{\partial U_0}{\partial x_0} - \alpha h^2 \frac{\partial U_1}{\partial x_1} + R_0 f_{d_0}^{x-} + R_1 f_{d_0}^{x+}\right]$

Algorithm 11. Given position-velocity updates $\{(x_0, R_0, v_0, \omega_0, u_0) \mapsto (x_1, R_1, v_1, \omega_1)\}$, minimize

- Discrepancies between the observed (x_1, v_1, ω_1) and the predicted $(\tilde{x}_1, \tilde{v}_1, \tilde{\omega}_1)$ triples
- Deviations away from the equation

$$hS(J\omega_0) + hS(f_{d_0}^{R-}) + (1-\alpha)h^2 S(\xi_0) \;=\; J_d Z_0 - Z_0^\top J_d$$

where, for each $(x_0, R_0, v_0, \omega_0, u_0, R_1)$ data tuple,

1. $f_{d_0}^{R\pm}$ and $f_{d_0}^{x\pm}$ are obtained from (x_0, R_0, u_0)
2. ξ_0 and ξ_1 are obtained from $S(\xi_k) = \frac{\partial U_k}{\partial R_k}^\top R_k - R_k^\top \frac{\partial U_k}{\partial R_k}$
3. $Z_0 = R_0^\top R_1$
4. $\tilde{\omega}_1 = J^{-1}\left[Z_0^\top J\omega_0 + (1-\alpha)hZ_0^\top \xi_0 + \alpha h\xi_1 + Z_0^\top f_{d_0}^{R-} + f_{d_0}^{R+}\right]$
5. $\tilde{x}_1 = x_0 + hv_0 - (1-\alpha)h^2 \frac{\partial U_0}{\partial x_0} - \frac{h}{m} R_0 f_{d_0}^{x-}$
6. $\tilde{v}_1 = v_0 + \frac{1}{m}\left[-(1-\alpha)h^2 \frac{\partial U_0}{\partial x_0} - \alpha h^2 \frac{\partial U_1}{\partial x_1} + R_0 f_{d_0}^{x-} + R_1 f_{d_0}^{x+}\right]$

To train the dynamics model with Algorithm 10, we minimize the loss function

$$\mathcal{L}_{10}(\theta) \;=\; \sum_{i=1}^{N} \left\| x_1^{(i)} - \tilde{x}_1^{(i)} \right\|^2 + \left\| \log\left(\tilde{R}_1^{(i)} R_1^{(i)\top}\right)^\vee \right\|^2 + \left\| v_1^{(i)} - \tilde{v}_1^{(i)} \right\|^2 + \left\| \omega_1^{(i)} - \tilde{\omega}_1^{(i)} \right\|^2, \tag{6.23}$$

while we use the following loss function for Algorithm 11

$$\begin{aligned} \mathcal{L}_{11}(\theta) \;=\; & \sum_{i=1}^{N} \left\| x_1^{(i)} - \tilde{x}_1^{(i)} \right\|^2 + \left\| v_1^{(i)} - \tilde{v}_1^{(i)} \right\|^2 + \left\| \omega_1^{(i)} - \tilde{\omega}_1^{(i)} \right\|^2 \\ & + \left\| hS\left(J\omega_0^{(i)}\right) + hS\left((f_{d_0}^{R-})^{(i)}\right) + (1-\alpha)h^2 S\left(\xi_0^{(i)}\right) - J_d Z_0^{(i)} + Z_0^{(i)\top} J_d \right\|^2. \end{aligned} \tag{6.24}$$

The network parameters θ are updated using Adam [Kingma and Ba, 2014], where the gradients $\partial\mathcal{L}/\partial\theta$ are calculated by back-propagation.

This general strategy extends to any other Lie group integrator. In particular, LieFVINs on SO(3) can be obtained from the algorithms above as the special case where x is constant, in which case we can disregard all the variables and operations in blue.

Lie group variational integrator networks without forces (**LieVINs**) can be obtained by setting $f_{d_0}^{R\pm} = f_{d_0}^{x\pm} = 0$. Note that the strategy behind Algorithm 10 enforces the structure of the system in a stronger way than in Algorithm 11. However, for certain Lie groups and variational integrators, it might not be practical to use Newton's method to solve for the implicit updates, in which case Algorithm 11 is preferred.

6.3.4 Control Strategy

Given the discrete-time flow map Ψ learnt by a LieFVIN, we can formulate a Model Predictive Control (MPC) problem to design a discrete-time control policy for the dynamical system:

At each step $t_\ell = \ell h$,

1. Obtain an estimate $(\tilde{q}_\ell, \dot{\tilde{q}}_\ell)$ of the current state.

2. Solve a N-step finite horizon optimal control problem starting at $(\tilde{q}_\ell, \dot{\tilde{q}}_\ell)$, formulated as a constrained optimization problem: *Minimize the discrete cost function*

$$\mathcal{G}_d(U_\ell) \;=\; \sum_{k=0}^{N-1} C_d(q_{\ell+k}, q_{\ell+k+1}, \dot{q}_{\ell+k}, u_{\ell+k}) + \Phi_d(q_{\ell+N-1}, q_{\ell+N}, \dot{q}_{\ell+N}, u_{\ell+N-1}), \tag{6.25}$$

over admissible discrete controls $U_\ell = \{u_\ell, u_{\ell+1}, ..., u_{\ell+N-1}\}$, *subject to path constraints*

$$\mathcal{P}_d(q_{\ell+k}, q_{\ell+k+1}, \dot{q}_{\ell+k}, u_{\ell+k}) \;\geq\; 0 \qquad \textit{for} \;\; k = 1, ..., N-1, \tag{6.26}$$

and to the termination condition

$$\mathcal{T}_d(q_{\ell+N-1}, q_{\ell+N}, \dot{q}_{\ell+N}, u_{\ell+N-1}) \;=\; 0, \tag{6.27}$$

and where the evolution of the controlled system is prescribed by the surrogate symplectic map Ψ *learnt by the LieFVIN.*

3. Apply the resulting optimal control u_ℓ^* to the system in state $(\tilde{q}_\ell, \dot{\tilde{q}}_\ell)$ until the next step $t_{\ell+1} = (\ell+1)h$.

The Lie group constraints do not need to be added as path constraints since they are automatically satisfied to (almost) machine precision, by the design of the LieFVINs.

6.4 Evaluation

We now demonstrate our approach to learn and control a planar pendulum and a crazyflie quadrotor. For the control tasks considered in this chapter, we will use the PyTorch MPC framework[1] [Tassa et al., 2014; Amos et al., 2018], and the code to reproduce these numerical experiments is open-sourced at

`https://thaipduong.github.io/LieFVIN/`

6.4.1 Pendulum

We consider a planar pendulum with dynamics

$$\ddot{\varphi} = -15 \sin \varphi + 3u, \tag{6.28}$$

where φ is the angle of the pendulum with respect to its vertically downward position and u is a scalar control input. The ground-truth mass of the pendulum, the potential energy, and the input coefficient are given by

$$m = 1/3, \qquad U(\varphi) = 5(1 - \cos \varphi), \qquad g(\varphi) = 1. \tag{6.29}$$

The forces were specified as

$$f_d^{R+} = 0 \qquad \text{and} \qquad f_d^{R-} = g(q)u. \tag{6.30}$$

We used neural networks to represent the inertial matrix $J(q) = L(q)L(q)^\top + \epsilon$, the potential energy $U(q)$ and the input gains $g(q)$ as follows:

- $L(q)$: 9 - 10 Tanh - 10 Tanh - 10 Linear - 6
- $U(q)$: 9 - 10 Tanh - 10 Tanh - 10 Linear - 1
- $g(q)$: 9 - 10 Tanh - 10 Tanh - 10 Linear - 3

where the first number is the input dimension while the last number is the output dimension, and the hidden layers are listed in-between with their dimensions and activation functions.

[1] Code: `https://locuslab.github.io/mpc.pytorch/`

The training data of the form $\{(\cos\varphi, \sin\varphi, \dot{\varphi})\}$ was collected from an OpenAI Gym environment, provided by [Zhong et al., 2019]. The control inputs were sampled in $[-3, 3]$ and applied to the planar pendulum for 10 time intervals of 0.02s to generate a total of 512 state-control trajectories. The SO(3) LieFVIN, as described in Algorithm 10 with $\alpha = 0.5$, was trained with a fixed learning rate of 10^{-3} for 10000 iterations.

Figures 6.1(a),(b),(c) show that the LieFVIN successfully learned the correct inertia matrix J, control gain $g(q)$, and potential energy function U (up to a constant offset). Without control input, i.e., $f_d^{R\pm} = 0$, we use the model learnt from short-term trajectories of 10 steps of 0.02s to generate long-term predictions (2000 steps, i.e. 40s). Figure 6.1(d) shows that the total energy of the learnt system fluctuates but stays close to the ground-truth value. The fluctuation comes from the discretization errors in equations (6.14)-(6.18) and model errors for the learnt quantities J, U, and $g(q)$. Note from Figure 6.1(e) that the SO(3) constraint errors remain very small, around 10^{-14}. The phase portraits of the system and the learnt dynamics are close to the ground-truth ones, illustrating the ability to generate long-term predictions using the model learnt from short-term data.

For comparison, we also learned the dynamics using a black-box model which is a Multilayer Perceptron MLP$(q, \dot{q}, u)$ with architecture [22 - 1000 Tanh - 1000 Tanh - 1000 Linear - 18]. As can be seen from Figures 6.1(d)(e)(g), that black-box model struggles to infer the SO(3) constraints from data and is not able to conserve the total energy.

The dynamics model learnt using the LieFVIN is combined with Model Predictive Control (MPC) as described in Section 6.3.4 to drive the planar pendulum from downward position $\varphi = 0$ to a stabilized upright position $\varphi^* = \pi$, $\dot{\varphi}^* = 0$, with input constraint $|u| \leq 20$. The running cost C_d and terminal cost Φ_d in the MPC problem are chosen to be

$$C_d(R_{\ell+k}, \omega_{\ell+k}, u_{\ell+k}) = \mathrm{Trace}(\mathbb{I}_3 - R^{*\top} R_{\ell+k}) + 0.1\|\omega_{\ell+k}\|^2 + 10^{-4}\|u_{\ell+k}\|^2, \tag{6.31}$$

$$\Phi_d(R_{\ell+k}, \omega_{\ell+k}, u_{\ell+k}) = \mathrm{Trace}(\mathbb{I}_3 - R^{*\top} R_{\ell+k}) + 0.1\|\omega_{\ell+k}\|^2 + 10^{-4}\|u_{\ell+k}\|^2. \tag{6.32}$$

Figure 6.1(h) plots the angle φ, angular velocity $\dot{\varphi}$, and control input u, showing that the planar pendulum is successfully stabilized in the upright position using the learnt discrete dynamics model.

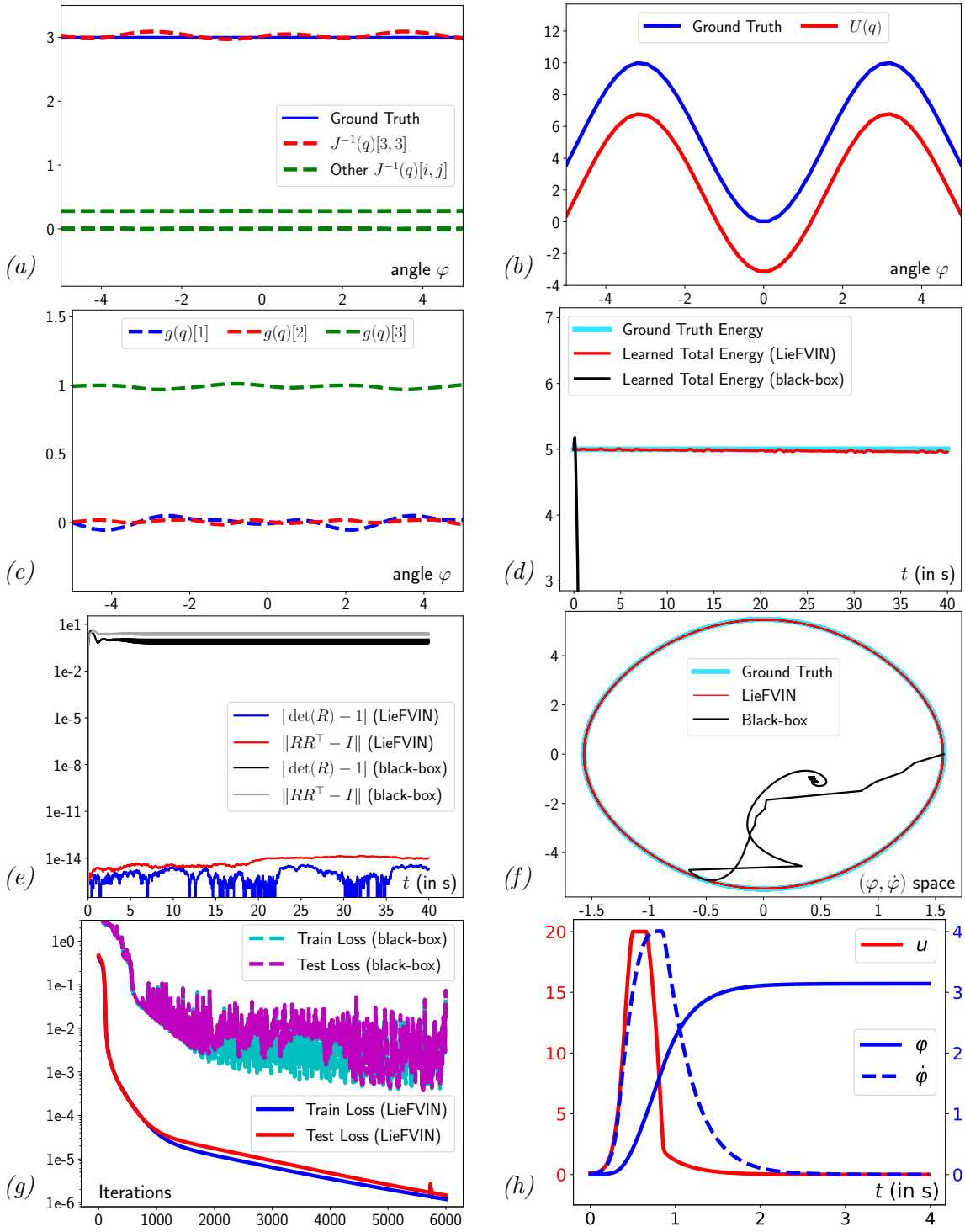


Figure 6.1: LieFVIN learns the correct mass (a), potential energy (b), input (c), and respects energy conservation (d), SO(3) constraints (e), and phase portraits (f). The loss is shown in (g). We use MPC to drive the pendulum to the upright position in (h).

6.4.2 Crazyflie Quadrotor

We now demonstrate that our SE(3) dynamics learning and control approach can achieve trajectory tracking for an under-actuated system.

We consider a Crazyflie quadrotor simulated in the physics-based simulator PyBullet [Panerati et al., 2020]. The control input $u = [f, \tau]$ includes the thrust $f \in \mathbb{R}_{\geq 0}$ and torque vector $\tau \in \mathbb{R}^3$ generated by the 4 rotors. The generalized coordinates q include position x and orientation R, and the velocity $\dot{q}$ includes linear velocity v and angular velocity ω. The forces are specified as

$$f_d^{x\pm} = 0.5 g_x(q) u \qquad \text{and} \qquad f_d^{R\pm} = 0.5 g_R(q) u. \tag{6.33}$$

We used neural networks to represent the mass $m = r^2$ of the quadrotor, the inertial matrix $J(q) = LL^\top + \epsilon$, the potential energy $U(q)$ and the input gains $g(q) = [g_x(q) \;\; g_R(q)]$ as follows:

- r: 1D PyTorch parameter
- L: 3×3 upper-triangular parameter matrix
- $U(q)$: 9 - 10 Tanh - 10 Tanh- 10 Tanh - 10 Linear - 1
- $g(q)$: 9 - 10 Tanh - 10 Tanh- 10 Tanh - 10 Linear - 24

where the first number is the input dimension, the last number is the output dimension, and the hidden layers are listed in-between with their dimensions and activation functions.

To obtain the training data, the crazyflie quadrotor was controlled from a random starting point to 36 different desired poses using a PID controller, yielding 36 4-second trajectories. The trajectories were used to generate a dataset of $N = 2700$ position-velocity updates $\{(q_0, \dot{q}_0, u_0) \mapsto (q_1, \dot{q}_1)\}$ with time-step 0.02s. The SE(3) LieFVIN, as described in Algorithm 11 with $\alpha = 0.5$, was trained with a decaying learning rate initialized at 5×10^{-3} for 20000 iterations.

Figures 6.2(a)(b)(c)(d)(e) show that the SE(3) LieFVIN model learned the correct mass m, inertia matrix J, control gains $g_x(q)$ and $g_R(x)$, and potential energy $U(q)$ (up to a constant offset).

Without control input, i.e., $f_d^{R\pm} = 0$, we use the dynamics LieFVIN model learnt from short-term trajectories of 5 steps of 0.02s to generate long-term predictions (2000 steps, for a total of 40s). Figure 6.2(f) shows that the total energy of the learnt system has bounded fluctuations while SO(3) constraint errors are around 10^{-14}, verifying the near-energy conservation and manifold constraints guaranteed by our approach.

The learnt dynamics model is then combined with MPC as described in Section 6.3.4 to track a predefined diamond-shaped trajectory. The running cost C_d and terminal cost Φ_d in the MPC problem are chosen to be

$$\begin{aligned} C_d(x_{\ell+k}, R_{\ell+k}, v_{\ell+k}, \omega_{\ell+k}, u_{\ell+k}) \;=\; & 1.2\|x_{\ell+k}\|^2 + 10^{-5}\ \mathrm{Trace}(\mathbb{I}_3 - R_{\ell+k}) \\ & + 1.2\|v_{\ell+k}\|^2 + 10^{-4}\|\omega_{\ell+k}\|^2 + 10^{-6}\|u_{\ell+k}\|^2, \end{aligned} \tag{6.34}$$

and

$$\begin{aligned} \Phi_d(x_{\ell+k}, R_{\ell+k}, v_{\ell+k}, \omega_{\ell+k}, u_{\ell+k}) \;=\; & 1.2\|x_{\ell+k}\|^2 + 10^{-5}\ \mathrm{Trace}(\mathbb{I}_3 - R_{\ell+k}) \\ & + 1.2\|v_{\ell+k}\|^2 + 10^{-4}\|\omega_{\ell+k}\|^2 + 10^{-6}\|u_{\ell+k}\|^2. \end{aligned} \tag{6.35}$$

The control input constraints on the thrust f and torque vector τ are given by

$$0 \;\le\; f \;\le\; 0.595, \qquad\qquad |\tau| \;\le\; 10^{-3}\,[5.9 \;\; 5.9 \;\; 7.4]^\top. \tag{6.36}$$

Figure 6.3 displays the robot trajectory and plots the tracking errors over time, showing that the quadrotor successfully completes the task.

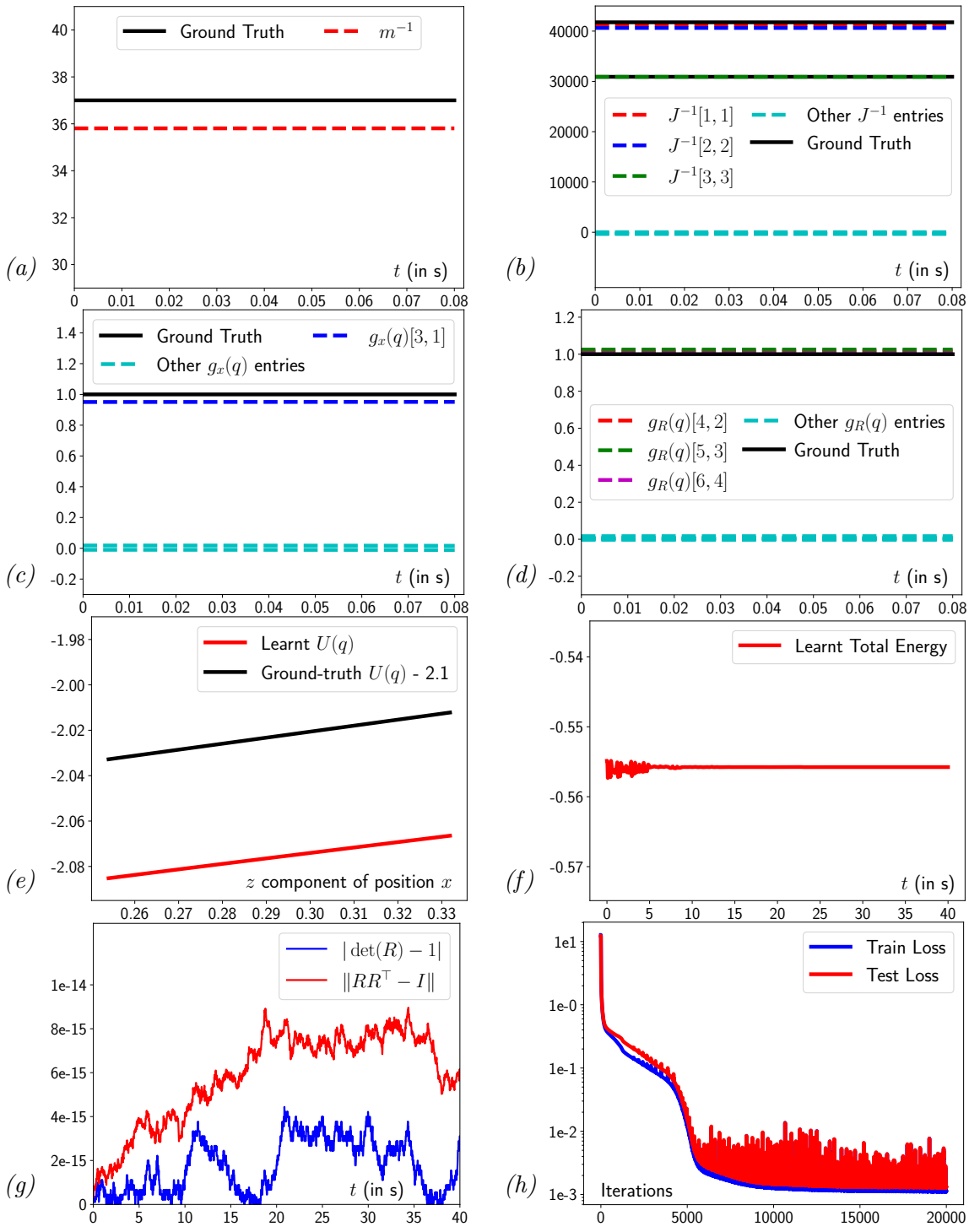


Figure 6.2: LieFVIN learns the correct mass m (a), inertia matrix J (b), input coefficients $g_x(q)$ (c) and $g_R(q)$ (d), potential energy $U(q)$ (e). The learnt model respects energy conservation (f), SO(3) constraints (g). The evolution of the loss function is shown in (h).

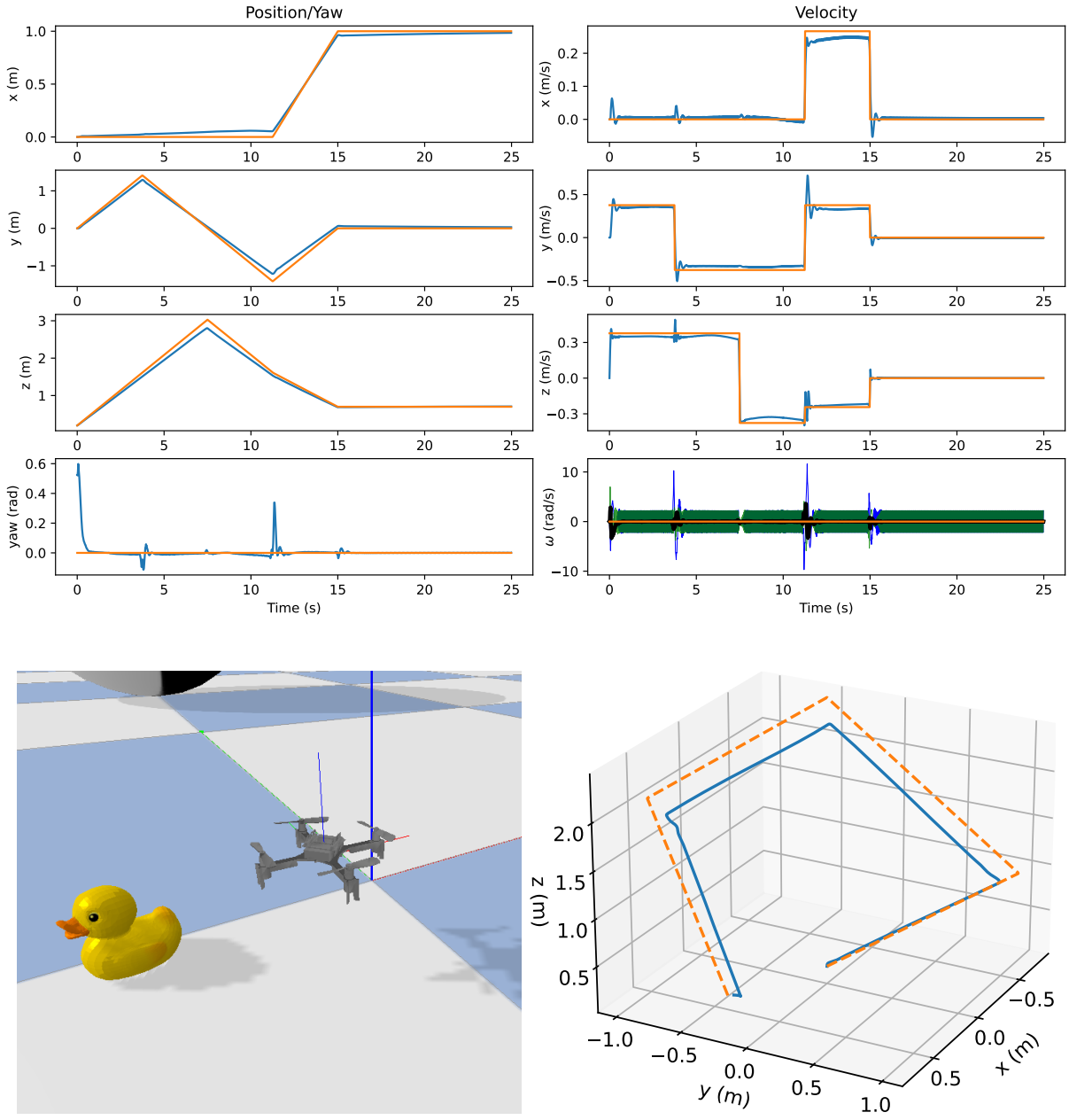


Figure 6.3: Trajectory tracking with the learned quadrotor model. The tracking errors (top plots) between reference trajectory (orange) and the actual trajectory, and the robot trajectory (lower right plot) show that the task is completed successfully.

6.5 Learning and controlling Lagrangian systems from position data

6.5.1 Problem Statement

We now consider the problem of learning controlled Lagrangian dynamics only from position data: given a position-only dataset of trajectories for a controlled Lagrangian system, we wish to infer the update map that generates these trajectories, while preserving the symplectic structure underlying the dynamical system and constraining the updates to remain on the Lie group on which it evolves. More precisely, we wish to solve the following problem:

Problem 6.2. *Given a dataset of N position-only updates*

$$\left\{\left(q_0^{(i)}, q_1^{(i)}, u_0^{(i)}, u_1^{(i)}\right) \mapsto q_2^{(i)}\right\}_{i=1}^{N}$$

for a controlled Lagrangian system evolving on the Lie group $\mathcal{Q}$, we wish to find a symplectic mapping $\Psi : \mathcal{Q} \times \mathcal{Q} \times \mathcal{U} \times \mathcal{U} \to \mathcal{Q}$ which minimizes

$$\sum_{i=1}^{N} \mathcal{D}_{\mathcal{Q}}\left(q_2^{(i)}, \Psi\left(q_0^{(i)}, q_1^{(i)}, u_0^{(i)}, u_1^{(i)}\right)\right), \tag{6.37}$$

where $\mathcal{D}_{\mathcal{Q}}$ is a distance metric on $\mathcal{Q}$.

6.5.2 Forced Variational Integrator in Lagrangian Form

As before, we choose the discrete Lagrangian

$$\begin{aligned} L_d(x_k, R_k, y_k, Z_k) \;=\; & \frac{m}{2h} y_k^{\mathsf{T}} y_k + \frac{1}{h}\mathrm{Trace}\left([\mathbb{I}_3 - Z_k] J_d\right) \\ & - (1-\alpha) h U(x_k, R_k) - \alpha h U(x_k + R_k y_k, R_k Z_k), \end{aligned} \tag{6.38}$$

where $\alpha \in [0,1]$ and $J_d = \frac{1}{2}\mathrm{Trace}(J)\mathbb{I}_3 - J$. We also define U_k and ξ_k via

$$U_k = U(x_k, R_k) \qquad \text{and} \qquad S(\xi_k) = \frac{\partial U_k}{\partial R_k}^{\mathsf{T}} R_k - R_k^{\mathsf{T}} \frac{\partial U_k}{\partial R_k}. \tag{6.39}$$

It is demonstrated in Appendix A.8 that the forced discrete Euler–Lagrange equations associated to the discrete Lagrangian (6.38) and the discrete control forces $f_{d_k}^{\pm} \equiv f_d^{\pm}(x_k, R_k, u_k)$ are given by

$$h^2 S(\xi_k) + hS(f_{d_k}^{R-}) + hS(f_{d_{k-1}}^{R+}) + (J_d Z_{k-1} - Z_{k-1}^\top J_d) \;=\; Z_k J_d - J_d Z_k^\top, \tag{6.40}$$

$$x_{k+1} = 2x_k - x_{k-1} - \frac{h^2}{m}\frac{\partial U_k}{\partial x_k} + \frac{h}{m} R_k (f_{d_k}^{x-} - f_{d_{k-1}}^{x+}), \tag{6.41}$$

$$R_{k+1} = R_k Z_k. \tag{6.42}$$

Since $(J_d Z_{k-1} - Z_{k-1}^\top J_d) \in \mathfrak{so}(3)$, equation (6.40) can be rewritten as

$$S(a) = Z_k J_d - J_d Z_k^\top \tag{6.43}$$

with

$$a = h^2 \xi_k + h f_{d_k}^{R-} + h f_{d_{k-1}}^{R+} + S^{-1}(J_d Z_{k-1} - Z_{k-1}^\top J_d). \tag{6.44}$$

Given $(x_{k-1}, x_k, R_{k-1}, R_k, u_{k-1}, u_k)$, we first solve $S(a) = ZJ_d - J_d Z^\top$ for $Z = Z_k$ as outlined in Remark 6.1, and then get $R_{k+1} = R_k Z_k$. We then update x_{k+1} using equation (6.41).

6.5.3 Lie Group Forced Variational Integrator Networks

We now describe the construction of Lie group Forced Variational Integrator Networks for the forced variational integrator on SE(3) presented in Section 6.5.2, in the case where only position data is available. The LieFVIN is based on the discrete forced Euler–Lagrange equations (6.40)-(6.42). As before, the main idea is to parametrize the updates of the forced variational integrator and match them with the observed updates.

We parametrize m, $f_d^{\pm}$ and U as neural networks, and the matrix J is a symmetric positive-definite matrix-valued function of (x, R) constructed via a Cholesky decomposition $J = LL^\top$ for a lower-triangular matrix L implemented as a neural network. We can also get $J_d = \frac{1}{2}\text{Trace}(J)\mathbb{I}_3 - J$. To deal with the implicit nature of equation (6.40), we propose two algorithms, based either on an explicit iterative solver or by penalizing deviations away from equation (6.40):

Algorithm 12. Given $(x_0, x_1, R_0, R_1, u_0, u_1) \mapsto (x_2, R_2)$ data, minimize discrepancies between the observed (x_2, R_2) pairs and the predicted $(\tilde{x}_2, \tilde{R}_2)$ pairs obtained as follows: For each $(x_0, x_1, R_0, R_1, u_0, u_1)$ data tuple,

1. Get $f_{d_0}^{R\pm}$, $f_{d_1}^{R\pm}$, $f_{d_0}^{x\pm}$,$f_{d_1}^{x\pm}$ from $(x_0, x_1, R_0, R_1, u_0, u_1)$
2. Get $S(\xi_1) = \frac{\partial U_1}{\partial R_1}^\top R_1 - R_1^\top \frac{\partial U_1}{\partial R_1}$
3. Get $\tilde{R}_2 = R_1 \mathrm{Cay}(\zeta)$ where ζ is obtained using a few steps of Newton's method to solve the vector equation (6.21) equivalent to the equation

$$h^2 S(\xi_1) + hS(f_{d_1}^{R-} + f_{d_0}^{R+}) + (J_d Z_0 - Z_0^\top J_d) \;=\; Z J_d - J_d Z$$

4. Compute $\tilde{x}_2 = 2x_1 - x_0 - \frac{h^2}{m}\frac{\partial U_1}{\partial x_1} + \frac{h}{m} R_1 (f_{d_1}^{x-} + f_{d_0}^{x})$

Algorithm 13. Given $(x_0, x_1, R_0, R_1, u_0, u_1) \mapsto (x_2, R_2)$ data, minimize

- Discrepancies between observed x_2 and the predicted $\tilde{x}_2$ values obtained via

$$\tilde{x}_2 = 2x_1 - x_0 - \frac{h^2}{m}\frac{\partial U_1}{\partial x_1} + \frac{h}{m} R_1 (f_{d_1}^{x-} + f_{d_0}^{x})$$

- Deviations away from the equation

$$J_d(R_0^\top R_1 + R_2^\top R_1) - (R_1^\top R_0 + R_1^\top R_2) J_d + h^2 \left(\frac{\partial U_1}{\partial R_1}^\top R_1 - R_1^\top \frac{\partial U_1}{\partial R_1} \right) + hS(f_{d_1}^{R-} + f_{d_0}^{R+}) = 0$$

This general strategy extends to any other Lie group integrator. In particular, LieFVINs on SO(3) can be obtained from the algorithms above as the special case where x is constant, in which case we can disregard all the variables and operations in blue. Lie group variational integrator networks without forces (**LieVINs**) can be obtained by setting $f_{d_0}^{R\pm} = f_{d_0}^{x\pm} = 0$. Note that the strategy behind Algorithm 12 enforces the structure of the system in a stronger way than in Algorithm 13. However, for certain Lie groups and variational integrators, it might not be practical to use Newton's method to solve for the implicit updates, in which case Algorithm 13 is preferred.

When combined with Model Predictive Control as described in Section 6.3.4, the initial conditions $(q_{\ell-1}, q_\ell)$ for the optimal control problems can be obtained either from the position estimates $(\tilde{q}_{\ell-1}, \tilde{q}_\ell)$ or from (position,velocity) estimates $(\tilde{q}_\ell, \dot{\tilde{q}}_\ell)$ with finite difference approximations. As before, the Lie group constraints for the system do not need to be added as path constraints since they are automatically satisfied to (almost) machine precision, by the design of the Lie group forced variational integrator networks.

6.6 Conclusion

In this chapter, we introduced a new structure-preserving deep learning strategy to learn discrete-time flow maps for controlled Lagrangian or Hamiltonian dynamics on a Lie group, from position-velocity or position-only data. The resulting maps evolve intrinsically on the Lie group and preserve the symplecticity underlying the systems of interest, which allows to generate physically well-behaved long-term predictions based on short-term trajectories data. Learning discrete-time flow maps instead of vector fields yields better prediction without requiring the use of a numerical integrator, neural ODE, or adjoint techniques. The proposed approach can also be combined with discrete-time optimal control strategies, for instance to achieve stabilization and tracking for robot systems on SE(3). Possible future directions include extensions to multi-link robots and multi-agent systems (for instance, on $(\mathrm{SE}(3))^n$).

➢ Chapter 6 contains original material from

① "Lie Group Forced Variational Integrator Networks for Learning and Control of Robot Systems" by V. Duruisseaux, T. Duong, M. Leok, and N. Atanasov. *Proceedings of the 5th Learning for Dynamics and Control Conference (L4DC),* PMLR 211:731-744, 2023

The dissertation author and Thai Duong were the primary investigators and authors of this paper.

7 Approximation of Nearly-Periodic Symplectic Maps via Structure-Preserving Neural Networks

In this chapter, we will consider the problem of learning dynamics for highly-oscillatory Hamiltonian systems. Examples include the Klein–Gordon equation in the weakly-relativistic regime, charged particles moving through a strong magnetic field, and the rotating inviscid Euler equations in quasi-geostrophic scaling [Cotter and Reich, 2004]. More generally, any Hamiltonian system may be embedded as a normally-stable elliptic slow manifold in a nearly-periodic Hamiltonian system [Burby and Hirvijoki, 2021]. Highly-oscillatory Hamiltonian systems exhibit two basic structural properties whose interactions play a crucial role in their long-term dynamics. First is preservation of the symplectic form, as for all Hamiltonian systems. Second is timescale separation, corresponding to the relatively short timescale of oscillations compared with slower secular drifts. Coexistence of these two structural properties implies the existence of an adiabatic invariant [Kruskal, 1962; Burby and Squire, 2020; Burby and Hirvijoki, 2021; Burby et al., 2023]. Adiabatic invariants differ from true constants of motion, in particular energy invariants, which do not change at all over arbitrary time intervals. Instead adiabatic invariants are conserved with limited precision over very large time intervals. There are no learning frameworks available today that exactly preserve the two structural properties whose interplay gives rise to adiabatic invariants. This work addresses this challenge by exploiting a recently-developed theory of *nearly-periodic symplectic maps* [Burby et al., 2023], which can be thought of as discrete-time analogues of highly-oscillatory Hamiltonian systems [Kruskal, 1962].

7.1 Introduction

A continuous-time dynamical system with parameter ε is nearly-periodic if all its trajectories are periodic with nowhere-vanishing angular frequency as ε approaches 0. Nearly-periodic maps, first introduced in [Burby et al., 2023], are natural discrete-time analogues of nearly-periodic systems. More precisely, they are defined as parameter-dependent diffeomorphisms that limit to rotations along a circle action. They have important applications to numerical integration of nearly-periodic systems, and may also be used as tools for structure-preserving simulation of non-canonical Hamiltonian systems on exact symplectic manifolds [Burby et al., 2023], which have numerous applications across the physical sciences. Noncanonical Hamiltonian systems play an especially important role in modeling weakly-dissipative plasma systems [Morrison, 1980; Morrison and Greene, 1980; Morrison and Kotschenreuther, 1989; Morrison, 1998; Burby et al., 2015; Morrison and Vanneste, 2016; Burby, 2022].

It is shown in details in [Kruskal, 1962] that every nearly-periodic system admits an approximate $\mathrm{U}(1)$ symmetry, determined to leading order by the unperturbed periodic dynamics. It is well-known that a Hamiltonian system which admits a continuous family of symmetries also admits a corresponding conserved quantity. It is thus not surprising that a nearly-periodic Hamiltonian system, which admits an approximate symmetry, must also have an approximate conservation law [Burby et al., 2023], and the approximately conserved quantity is referred to as an *adiabatic invariant*. Similarly to the continuous-time case, nearly-periodic maps with a Hamiltonian structure (that is symplecticity) admit formal $\mathrm{U}(1)$ symmetries to all orders when the limiting rotation is non-resonant, and as a result also possess a discrete-time adiabatic invariant [Burby et al., 2023]. The adiabatic invariants that our networks target only arise in purely Hamiltonian systems. Just like dissipation breaks the link between symmetries and conservation laws in Hamiltonian systems, dissipation also breaks the link between approximate symmetries and approximate conservation laws in Hamiltonian systems. We are not considering systems with symmetries that are broken by dissipation or some other mechanism, but rather considering systems which possess approximate symmetries. This should be contrasted with other frameworks [Hernandez et al., 2021; Huang et al., 2022; Hernández et al., 2023] which develop machine learning techniques for systems that explicitly include dissipation.

It is important to preserve the geometric structure of a dynamical system when trying to learn a surrogate for its dynamics. In particular, it is necessary to have structure-preserving architectures which can learn symplectic flow maps and ensure that the learnt surrogate maps are symplectic, and many physics-informed approaches have been proposed to learn Hamiltonian dynamics and symplectic maps. In particular, *Hénon Networks* (HénonNets) [Burby et al., 2020] can approximate arbitrary well any symplectic map via compositions of simple yet expressive elementary symplectic maps, and will be presented in Section 7.2 since they form a basis for the architecture introduced in this chapter.

Here, we construct a novel structure-preserving neural network, the *symplectic gyroceptron*, to approximate nearly-periodic symplectic maps, while ensuring that the resulting surrogate map is nearly-periodic and symplectic, and that it gives rise to a discrete-time adiabatic invariant and a long-time stability. This new framework provides a promising architecture for surrogate modeling of non-dissipative dynamical systems that automatically steps over short timescales without introducing spurious instabilities.

Note that deep learning architectures designed for multi-scale dynamics and long-time dependencies exist [Rusch and Mishra, 2021], and that algorithms specifically designed to efficiently step over high-frequency oscillations [Chen et al., 2011; Chen and Chacón, 2015; Miller et al., 2019] have been introduced as well. However, the problem of developing surrogates for dynamical systems that avoid resolving short oscillations remains open. Such surrogates would accelerate optimization algorithms that require querying the dynamics of an oscillatory system during the optimization. The architecture presented here represents a first important step toward a general solution of this problem. Some of its advantages are that it aims to learn a fast surrogate that can resolve long-time dynamics using very short time data, and that it is guaranteed to enjoy symplectic universal approximation within the class of nearly periodic maps. As developed here, our method applies to dynamical systems that exhibit a single fast mode of oscillation. In particular, when initial conditions for the surrogate are selected on the zero level set of the learned adiabatic invariant, the network automatically integrates along the slow manifold [Lorenz, 1986; Lorenz and Krishnamurthy, 1987; Lorenz, 1992; MacKay, 2004; Burby and Klotz, 2020]. While our architecture generalizes in a straightforward manner to handle multiple non-resonant modes, it cannot be applied to dynamical systems that exhibit resonant surfaces.

Note that many of the approaches listed earlier for physics-based or structure-preserving learning of Hamiltonian dynamics focus on learning the vector field associated to the continuous-time Hamiltonian system, while others learn a discrete-time symplectic approximation to the flow map of the Hamiltonian system. In many contexts, we do not need to infer the continuous-time dynamics, and only need a surrogate model which can rapidly generate accurate predictions which remain physically consistent for a long time. Learning a discrete-time approximation to the evolution or flow map, instead of learning the continuous-time vector field, allows for fast prediction and simulation without the need to integrate differential equations or use neural ODEs and adjoint techniques (which can be very expensive and can introduce additional errors due to discretization). Here, we will learn nearly-periodic symplectic approximations to the flow maps of nearly-periodic Hamiltonian systems, with the intention of obtaining algorithms which can generate accurate and physically-consistent simulations much faster than traditional integrators.

Outline. We first discuss how symplectic maps can be approximated using HénonNets in Section 7.2, before defining and reviewing the important properties of nearly-periodic systems and nearly-periodic maps in Section 7.3, with a specific focus on the Hamiltonian and symplectic case. In Section 7.4, we introduce novel neural network architectures, the *gyroceptrons* and *symplectic gyroceptrons*, to approximate nearly-periodic maps in the symplectic and non-symplectic cases. We show that symplectic gyroceptrons admit adiabatic invariants regardless of the values of their weights in Section 7.5. Finally, in Section 7.6, we demonstrate how the proposed architecture can be used to learn surrogate maps for the nearly-periodic symplectic flow maps associated to two different systems: a nearly-periodic Hamiltonian system composed of two nonlinearly coupled oscillators (in Section 7.6.1), and the nearly-periodic Hamiltonian system describing the evolution of a charged particle interacting with its self-generated electromagnetic field (in Section 7.6.2).

In this chapter, M denotes a smooth manifold equipped with a smooth auxiliary Riemannian metric g, and $\mathcal{E}$ will always denote a vector space for the parameter ε. We say that a map $f_\varepsilon : M_1 \to M_2$, $\varepsilon \in \mathcal{E}$, between smooth manifolds M_1 and M_2, is a smooth ε-dependent mapping when the mapping $M_1 \times \mathbb{R} \to M_2 : (m, \varepsilon) \mapsto f_\varepsilon(m)$ is smooth.

7.2 Approximation of Symplectic Maps via Hénon Neural Networks

We have already seen numerous times throughout this dissertation that preserving the symplecticity of a Hamiltonian system when constructing a discrete approximation of its flow map is highly desirable since it ensures the preservation of many aspects of the dynamical system such as energy conservation, and leads to physically well-behaved discrete solutions. It is thus important to have structure-preserving network architectures which can learn symplectic maps.

The space of all symplectic maps is infinite dimensional [Weinstein, 1971], so the problem of approximating an arbitrary symplectic map using compositions of simpler symplectic mappings is inherently interesting. In [Turaev, 2002], Turaev showed that every symplectic mapping may be approximated arbitrarily well by compositions of *Hénon-like maps*, which are special elementary symplectic mappings.

Definition 7.1. *Let $V : \mathbb{R}^n \to \mathbb{R}$ be a smooth function and let $\eta \in \mathbb{R}^n$ be a constant. We define the* ***Hénon-like map*** *$H[V,\eta] : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$ with potential V and shift η via*

$$\boxed{H[V,\eta]\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} y+\eta \\ -x+\nabla V(y) \end{pmatrix}.} \tag{7.1}$$

Theorem 7.1 ([Turaev, 2002])**.** *Let $\Phi : U \to \mathbb{R}^n \times \mathbb{R}^n$ be a C^{r+1} symplectic mapping. For each compact set $C \subset U$ and $\delta > 0$ there is a smooth function $V : \mathbb{R}^n \to \mathbb{R}$, a constant η, and a positive integer N such that $H[V,\eta]^{4N}$ approximates Φ within δ in the C^r topology.*

Remark 7.1. *The significance of the number 4 in Turaev's Theorem follows from the fact that the fourth iterate of the Hénon-like map with trivial potential $V = 0$ is the identity map: $H[0,\eta]^4 = \mathrm{Id}_{\mathbb{R}^n \times \mathbb{R}^n}$.*

Turaev's result suggests the specific neural network architecture introduced in [Burby et al., 2020] to approximate symplectic mappings using Hénon-like maps. We review the construction of HénonNets [Burby et al., 2020], starting with *Hénon layers*.

Definition 7.2. *Let $\eta \in \mathbb{R}^n$ be a constant vector, and let V be a scalar feed-forward neural network on $\mathbb{R}^n$, that is., a smooth mapping $V : \mathcal{W} \times \mathbb{R}^n \to \mathbb{R}$, where $\mathcal{W}$ is a space of neural network weights. The* ***Hénon layer*** *with potential V, shift η, and weight W is the iterated Hénon-like map*

$$\boxed{L[V[W], \eta] = H[V[W], \eta]^4,} \tag{7.2}$$

where we use the notation $V[W]$ to denote the mapping

$$V[W](y) = V(W, y), \qquad \text{for any } y \in \mathbb{R}^n, \quad W \in \mathcal{W}. \tag{7.3}$$

There are various network architectures for the potential $V[W]$ that are capable of approximating any smooth function $V : \mathbb{R}^n \to \mathbb{R}$ with any desired level of accuracy. For example, a fully-connected neural network with a single hidden layer of sufficient width can approximate any smooth function. Therefore a corollary of Theorem 7.1 is that any symplectic map may be approximated arbitrarily well by the composition of sufficiently many Hénon layers with various potentials and shifts. This leads to the notion of a *Hénon Neural Network*.

Definition 7.3. *Let N be a positive integer and*

- $\boldsymbol{V} = \{V_k\}_{k \in \{1,\dots,N\}}$ *be a family of scalar feed-forward neural networks on $\mathbb{R}^n$*
- $\boldsymbol{W} = \{W_k\}_{k \in \{1,\dots,N\}}$ *be a family of network weights for $\boldsymbol{V}$*
- $\boldsymbol{\eta} = \{\eta_k\}_{k \in \{1,\dots,N\}}$ *be a family of constants in $\mathbb{R}^n$*

The ***Hénon neural network (HénonNet)*** *with layer potentials $\boldsymbol{V}$, layer weights $\boldsymbol{W}$, and layer shifts $\boldsymbol{\eta}$ is the mapping*

$$\boxed{\begin{aligned} \mathcal{H}[\boldsymbol{V}[\boldsymbol{W}], \boldsymbol{\eta}] &= L[V_N[W_N], \eta_N] \circ \dots \circ L[V_2[W_2], \eta_2] \circ L[V_1[W_1], \eta_1] \\ &= H[V_N[W_N], \eta_N]^4 \circ \dots \circ H[V_2[W_2], \eta_2]^4 \circ H[V_1[W_1], \eta_1]^4. \end{aligned}} \tag{7.4}$$

A composition of symplectic mappings is also symplectic, so every HénonNet is a symplectic mapping, regardless of the architectures for the networks V_k and of the weights W_k. Furthermore, Turaev's Theorem 7.1 implies that the family of HénonNets is sufficiently expressive to approximate any symplectic mapping:

Lemma 7.1. *Let $\Phi : U \to \mathbb{R}^n \times \mathbb{R}^n$ be a C^{r+1} symplectic mapping. For each compact set $C \subset U$ and $\delta > 0$ there is a HénonNet $\mathcal{H}$ that approximates Φ within δ in the C^r topology.*

Remark 7.2. *Note that Hénon-like maps are easily invertible,*

$$H[V,\eta]\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} y+\eta \\ -x+\nabla V(y) \end{pmatrix} \quad \Rightarrow \quad H^{-1}[V,\eta]\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \nabla V(x-\eta)-y \\ x-\eta \end{pmatrix}, \tag{7.5}$$

so we can also easily invert Hénon networks by composing inverses of Hénon-like maps.

We also introduce here modified versions of Hénon-like maps and HénonNets to approximate symplectic maps possessing a near-identity property:

Definition 7.4. *Let $V : \mathbb{R}^n \to \mathbb{R}$ be a smooth function and let $\eta \in \mathbb{R}^n$ be a constant. We define the* ***near-identity Hénon-like map*** *$H_\varepsilon[V,\eta] : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$ with potential V and shift η via*

$$\boxed{H_\varepsilon[V,\eta]\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} y+\eta \\ -x+\varepsilon\nabla V(y) \end{pmatrix}.} \tag{7.6}$$

Near-identity Hénon-like maps satisfy the near-identity property $H_0[V,\eta]^4 = \mathrm{Id}_{\mathbb{R}^n\times\mathbb{R}^n}$.

Definition 7.5. *Let N be a positive integer and*

- $\boldsymbol{V} = \{V_k\}_{k\in\{1,\dots,N\}}$ *be a family of scalar feed-forward neural networks on $\mathbb{R}^n$*
- $\boldsymbol{W} = \{W_k\}_{k\in\{1,\dots,N\}}$ *be a family of network weights for $\boldsymbol{V}$*
- $\boldsymbol{\eta} = \{\eta_k\}_{k\in\{1,\dots,N\}}$ *be a family of constants in $\mathbb{R}^n$*

The ***near-identity Hénon network*** *with layer potentials $\boldsymbol{V}$, layer weights $\boldsymbol{W}$, and layer shifts $\boldsymbol{\eta}$ is the mapping defined via*

$$\boxed{\mathcal{H}_\varepsilon[\boldsymbol{V}[\boldsymbol{W}],\boldsymbol{\eta}] = H_\varepsilon[V_N[W_N],\eta_N]^4 \circ \dots \circ H_\varepsilon[V_2[W_2],\eta_2]^4 \circ H_\varepsilon[V_1[W_1],\eta_1]^4,} \tag{7.7}$$

and it satisfies the near-identity property $\mathcal{H}_0[\boldsymbol{V}[\boldsymbol{W}],\boldsymbol{\eta}] = \mathrm{Id}_{\mathbb{R}^n\times\mathbb{R}^n}$.

7.3 Nearly-Periodic Systems and Maps

7.3.1 Nearly-Periodic Systems

Intuitively, a continuous-time dynamical system with vector parameter ε is *nearly-periodic* if all of its trajectories are periodic with nowhere-vanishing angular frequency in the limit $\varepsilon \to 0$. Such a system characteristically displays limiting short-timescale dynamics that ergodically cover circles in phase space. More precisely, a nearly-periodic systems can be defined as follows:

Definition 7.6 ([Burby et al., 2023])**.** *A* ***nearly-periodic system*** *on a manifold M is a smooth ε-dependent vector field X_ε on M such that $X_0 = \omega_0 R_0$, where*

- *R_0 is the infinitesimal generator for a circle action $\Phi_\theta : M \to M$, $\theta \in \mathrm{U}(1)$.*
- *$\omega_0 : M \to \mathbb{R}$ is strictly positive and its Lie derivative satisfies $\mathcal{L}_{R_0}\omega_0 = 0$.*

The vector field R_0 is called the ***limiting roto-rate****, and the frequency ω_0 is called the* ***limiting angular frequency****.*

Examples from physics include charged particle dynamics in a strong magnetic field, the weakly-relativistic Dirac equation, and any mechanical system subject to a high-frequency, time-periodic force. In the broader context of multi-scale dynamical systems, nearly-periodic systems play a special role because they display perhaps the simplest possible non-dissipative short-timescale dynamics. They therefore provide a useful proving ground for analytical and numerical methods aimed at more complex multi-scale models.

Remark 7.3. *In a seminal paper [Kruskal, 1962] on basic properties of continuous-time nearly-periodic systems, Kruskal assumed that R_0 is nowhere vanishing, in addition to requiring that ω_0 is sign-definite. This assumption is usually not essential and it is enough to require that ω_0 vanishes nowhere. This is an important restriction to lift since many interesting circle actions have fixed points.*

It was shown in [Kruskal, 1962] that every nearly-periodic system admits an approximate $\mathrm{U}(1)$-symmetry, known as the *roto-rate*, that is determined to leading order by the unperturbed periodic dynamics:

Definition 7.7. *A **roto-rate** for a nearly-periodic system X_ε on a manifold M is a formal power series $R_\varepsilon = R_0 + \varepsilon R_1 + \varepsilon^2 R_2 + \dots$ with vector field coefficients such that R_0 is equal to the limiting roto-rate and the following equalities hold in the sense of formal series:*

$$\exp(2\pi \mathcal{L}_{R_\varepsilon}) = 1 \qquad \text{and} \qquad [X_\varepsilon, R_\varepsilon] = 0. \tag{7.8}$$

Proposition 7.1 ([Kruskal, 1962])**.** *A nearly-periodic system admits a unique roto-rate R_ε.*

A subtle bootstrapping argument allows to upgrade leading-order $\mathrm{U}(1)$-invariance to all-orders $\mathrm{U}(1)$-invariance for integral invariants:

Proposition 7.2 ([Burby et al., 2023])**.** *Let α_ε be a smooth ε-dependent differential form on a manifold M. Suppose that α_ε is an absolute integral invariant for a smooth nearly-periodic system X_ε on M. If $\mathcal{L}_{R_0}\alpha_0 = 0$ then*

$$\mathcal{L}_{R_\varepsilon}\alpha_\varepsilon = 0, \tag{7.9}$$

where R_ε is the roto-rate for X_ε.

7.3.2 Nearly-Periodic Maps

Nearly-periodic maps are the natural discrete-time analogues of nearly-periodic systems, which were first introduced and studied in [Burby et al., 2023]. The following provides a precise definition.

Definition 7.8. *A **nearly-periodic map** on a manifold M with parameter vector space $\mathcal{E}$ is a smooth mapping $F : M \times \mathcal{E} \to M$ such that $F_\varepsilon : M \to M : m \mapsto F(m, \varepsilon)$ has the following properties:*

- *F_ε is a diffeomorphism for each $\varepsilon \in \mathcal{E}$.*
- *There exists a $\mathrm{U}(1)$-action $\Phi_\theta : M \to M$ and a constant $\theta_0 \in \mathrm{U}(1)$ such that $F_0 = \Phi_{\theta_0}$.*

*We say F is **resonant** if θ_0 is a rational multiple of 2π, otherwise F is **non-resonant**. The infinitesimal generator of Φ_θ, R_0, is the **limiting roto-rate**.*

Example 7.1. *Let X_ε be a nearly-periodic system on a manifold M with limiting roto-rate R_0 and limiting angular frequency ω_0. Assume that ω_0 is constant. For each $\varepsilon \in \mathbb{R}$ let $\mathcal{F}_t^\varepsilon$ denote the time-t flow for X_ε. Then, the mapping*

$$F(m, \varepsilon) = \mathcal{F}_{t_0}^\varepsilon(m) \tag{7.10}$$

is nearly-periodic for each t_0.

To see why, first note that the flow of the limiting vector field $X_0 = \omega_0 R_0$ is given by $\mathcal{F}_t^0(m) = \Phi_{\omega_0 t}(m)$, where Φ_θ denotes the $\mathrm{U}(1)$-action generated by the limiting roto-rate R_0. It follows that

$$F(m, 0) \;=\; \Phi_{\omega_0 t_0}(m) \;=\; \Phi_{\theta_0}(m), \tag{7.11}$$

where $\theta_0 = \omega_0 t_0 \bmod 2\pi$. This example is more general than it first appears since any nearly-periodic system can be rescaled to have a constant limiting angular frequency. Indeed if the nearly-periodic system X_ε has non-constant limiting angular frequency ω_0 then $X'_\varepsilon = X_\varepsilon/\omega_0$ is a nearly-periodic system with limiting angular frequency 1. The integral curves of X'_ε are merely time reparameterizations of integrals curves of X_ε.

Let X be a vector field on a manifold M with time-t flow map $\mathcal{F}_t$. Recall that $\mathrm{U}(1)$-action Φ_θ is a $\mathrm{U}(1)$-*symmetry* for X if

$$\mathcal{F}_t \;\circ\; \Phi_\theta \;=\; \Phi_\theta \;\circ\; \mathcal{F}_t, \tag{7.12}$$

for each $t \in \mathbb{R}$ and $\theta \in \mathrm{U}(1)$. Differentiating this condition with respect to θ at the identity implies, and is implied by, $\mathcal{F}_t^* R = R$, where R denotes the infinitesimal generator for the $\mathrm{U}(1)$-action. Since we would like to think of nearly-periodic maps as playing the part of a nearly-periodic system's flow map, the latter characterization of symmetry allows us to naturally extend Kruskal's notion of roto-rate to our discrete-time setting.

Definition 7.9. *A **roto-rate** for a nearly-periodic map $F : M \times \mathcal{E} \to M$ is a formal power series*

$$R_\varepsilon = R_0 + R_1\varepsilon + R_2\varepsilon^2 + \dots \tag{7.13}$$

whose coefficients are vector fields on M such that R_0 is the limiting roto-rate and the following equalities hold in the sense of formal power series:

$$F_\varepsilon^* R_\varepsilon = R_\varepsilon \qquad \textit{and} \qquad \exp(2\pi \pounds_{R_\varepsilon}) = 1. \tag{7.14}$$

A first fundamental result fornearly-periodic maps establishes the existence and uniqueness of the roto-rate in the non-resonant case. Similarly to the corresponding result [Kruskal, 1962] in continuous time, this discrete time result holds to all orders in perturbation theory.

Theorem 7.2 ([Burby et al., 2023])**.** *Each non-resonant nearly-periodic map admits a unique roto-rate.*

Thus, non-resonant nearly-periodic maps formally reduce to mappings on the space of U(1)-orbits, corresponding to the elimination of a single dimension in phase space.

7.3.3 Nearly-Periodic Systems and Maps with a Hamiltonian Structure

Definition 7.10. *A* ***ε-dependent presymplectic manifold*** *is a manifold M equipped with a smooth ε-dependent 2-form Ω_ε such that $\mathbf{d}\Omega_\varepsilon = 0$ for each $\varepsilon \in \mathcal{E}$. We say (M, Ω_ε) is* ***exact*** *when there is a smooth ε-dependent 1-form ϑ_ε such that $\Omega_\varepsilon = -\mathbf{d}\vartheta_\varepsilon$.*

Definition 7.11. *A* ***nearly-periodic Hamiltonian system*** *on an exact presymplectic manifold (M, Ω_ε) is a nearly-periodic system X_ε on M such that*

$$\iota_{X_\varepsilon}\Omega_\varepsilon = \mathbf{d}H_\varepsilon, \tag{7.15}$$

for some smooth ε-dependent function $H_\varepsilon : M \to \mathbb{R}$.

We already know from Proposition 7.1 that every nearly-periodic system admits a unique roto-rate R_ε. In the Hamiltonian setting, it can be shown that both the dynamics and the Hamiltonian structure are U(1)-invariant to all orders in ε.

Proposition 7.3 ([Kruskal, 1962; Burby et al., 2023])**.** *The roto-rate R_ε for a nearly-periodic Hamiltonian system X_ε on an exact presymplectic manifold (M, Ω_ε) with Hamiltonian H_ε satisfies*

$$\mathcal{L}_{R_\varepsilon}H_\varepsilon = 0, \qquad \textit{and} \qquad \mathcal{L}_{R_\varepsilon}\Omega_\varepsilon = 0, \tag{7.16}$$

in the sense of formal power series.

According to Noether's celebrated theorem, a Hamiltonian system that admits a continuous family of symmetries also admits a corresponding conserved quantity [Abraham and Marsden, 1978; Arnol′d, 1989; Marsden and Ratiu, 1999]. Therefore one might expect that a Hamiltonian system with an approximate symmetry must also have an approximate conservation law. This is indeed the case for nearly-periodic Hamiltonian systems:

Proposition 7.4 ([Burby et al., 2023])**.** *Let X_ε be a nearly-periodic Hamiltonian system on the exact presymplectic manifold (M, Ω_ε). Let R_ε be the associated roto-rate. There is a formal power series $\theta_\varepsilon = \theta_0 + \varepsilon\,\theta_1 + \ldots$ with coefficients in $\Omega^1(M)$ such that*

$$\Omega_\varepsilon = -\mathbf{d}\theta_\varepsilon, \qquad \text{and} \qquad \mathcal{L}_{R_\varepsilon}\theta_\varepsilon = 0. \tag{7.17}$$

Moreover, the formal power series

$$\mu_\varepsilon = \iota_{R_\varepsilon}\theta_\varepsilon \tag{7.18}$$

is a constant of motion for X_ε to all orders in perturbation theory. In other words,

$$\mathcal{L}_{X_\varepsilon}\mu_\varepsilon = 0, \tag{7.19}$$

in the sense of formal power series. The formal constant of motion μ_ε is the ***adiabatic invariant*** *associated with the nearly-periodic Hamiltonian system.*

General expressions for the adiabatic invariant μ_ε may be found in [Burby and Squire, 2020]. Additionally, it was shown in [Burby and Hirvijoki, 2021], that the (formal) set of fixed points for the roto-rate is an elliptic almost invariant slow manifold whose normal stability is mediated by the adiabatic invariant associated with the nearly-periodic Hamiltonian system.

A similar theory can be established in the discrete case for nearly-periodic maps with a Hamiltonian structure.

Definition 7.12. *A* ***presymplectic nearly-periodic map*** *on a ε-dependent presymplectic manifold (M, Ω_ε) is a nearly-periodic map F such that $F_\varepsilon^*\Omega_\varepsilon = \Omega_\varepsilon$ for each $\varepsilon \in \mathcal{E}$.*

Theorem 7.3 ([Burby et al., 2023])**.** *If F is a non-resonant presymplectic nearly-periodic map on a ε-dependent presymplectic manifold (M, Ω_ε) with roto-rate R_ε then $\mathcal{L}_{R_\varepsilon}\Omega_\varepsilon = 0$.*

Definition 7.13. *A **Hamiltonian nearly-periodic map** on a ε-dependent presymplectic manifold (M, Ω_ε) is a nearly-periodic map F such that there is a smooth (t, ε)-dependent vector field $Y_{t,\varepsilon}$ with $t \in \mathbb{R}$ such that the following properties hold true:*

- $\iota_{Y_{t,\varepsilon}} \Omega_\varepsilon = \mathbf{d} H_{t,\varepsilon}$, *for some smooth* (t, ε)*-dependent function* $H_{t,\varepsilon}$.
- *For each* $\varepsilon \in \mathcal{E}$, F_ε *is the* $t = 1$ *flow of* $Y_{t,\varepsilon}$.

Lemma 7.2. *Hamiltonian nearly-periodic maps are presymplectic nearly-periodic maps.*

Using presymplecticity of the roto-rate, Noether's theorem can be used to establish existence of adiabatic invariants for many interesting presymplectic nearly-periodic maps.

Theorem 7.4 ([Burby et al., 2023])**.** *Let F be a non-resonant presymplectic nearly-periodic map on the exact ε-dependent presymplectic manifold (M, Ω_ε) with roto-rate R_ε. Assume that F is Hamiltonian or that the manifold M is connected and the limiting roto rate R_0 has at least one zero. Then there exists a smooth ε-dependent 1-form θ_ε such that*

$$\mathcal{L}_{R_\varepsilon} \theta_\varepsilon = 0, \qquad \text{and} \qquad -\mathbf{d}\theta_\varepsilon = \Omega_\varepsilon, \tag{7.20}$$

in the sense of formal power series. Moreover the quantity

$$\mu_\varepsilon = \iota_{R_\varepsilon} \theta_\varepsilon \tag{7.21}$$

satisfies

$$F_\varepsilon^* \mu_\varepsilon = \mu_\varepsilon \tag{7.22}$$

in the sense of formal power series, that is, μ_ε is an adiabatic invariant for F.

When an adiabatic invariant exists, the phase-space dimension is formally reduced by two. On the slow manifold $\mu_\varepsilon = 0$, the reduction in dimensionality may be even more dramatic. For example, the slow manifold for the symplectic Lorentz system introduced in [Burby and Hirvijoki, 2021] has half the dimension of the full system.

7.4 Novel Structure-Preserving Neural Network Architectures

7.4.1 Approximating Nearly-Periodic Maps via Gyroceptrons

We first consider the problem of approximating an arbitrary nearly-periodic map $P : M \times \mathcal{E} \to M$ on a manifold M. From Definition 7.8, there must be a corresponding circle action $\Phi_\theta : M \to M$ and $\theta_0 \in \mathrm{U}(1)$ such that $P_0 = \Phi_{\theta_0}$. Consider the map $I_\varepsilon : M \to M$ given by

$$I_\varepsilon \;=\; P_\varepsilon \;\circ\; \Phi_{\theta_0}^{-1} \qquad \forall \varepsilon \in \mathcal{E}. \tag{7.23}$$

This defines a near-identity map on M satisfying $I_0 = \mathrm{Id}_M$.

By composing both sides of equation (7.23) on the right by Φ_{θ_0}, we obtain a representation for any nearly-periodic map P as the composition of a near-identity map and a circle action,

$$P_\varepsilon \;=\; I_\varepsilon \;\circ\; \Phi_{\theta_0} \qquad \forall \varepsilon \in \mathcal{E}. \tag{7.24}$$

As a consequence, if we can approximate any near-identity map and any circle action, then by the above representation we can approximate any nearly-periodic map.

Different circle actions can act on manifolds in topologically different ways, so it would be very challenging, if not impossible, to construct a single strategy which allows to approximate any circle action to arbitrary accuracy. Here, we will consider the simpler case where we assume that we know a priori the topological type of action for the nearly-periodic system, and work within conjugation classes. Conjugation of a circle action $\Phi_\theta : M \to M$ with a diffeomorphism ψ results in the map

$$\psi \;\circ\; \Phi_\theta \;\circ\; \psi^{-1}, \tag{7.25}$$

and two circle actions belong to the same conjugation class if one can be written as the conjugation with a diffeomorphism of the other one. Note that although compositions of nearly-periodic maps are not necessarily nearly-periodic, the map obtained by conjugation of a nearly-periodic map with a diffeomorphism is nearly-periodic:

Lemma 7.3. *Let $P : M \times \mathcal{E} \to M$ be a nearly-periodic map on a manifold M, and let $\psi : M \to M$ be a diffeomorphism on M. Then the map $\tilde{P} : M \times \mathcal{E} \to M$ defined for any $\varepsilon \in \mathcal{E}$ via*

$$\tilde{P}_\varepsilon \;\equiv\; \psi \;\circ\; P_\varepsilon \;\circ\; \psi^{-1} \tag{7.26}$$

is a nearly-periodic map.

Proof. ψ and P_ε are diffeomorphisms for any $\varepsilon \in \mathcal{E}$ so $\tilde{P}_\varepsilon$ is also a diffeomorphism for any $\varepsilon \in \mathcal{E}$. Now, from Definition 7.8, there is a circle action $\Phi_\theta : M \to M$ and $\theta_0 \in \mathrm{U}(1)$ such that $P_0 = \Phi_{\theta_0}$. Define $\tilde{\Phi}_\theta : M \to M$ via $\tilde{\Phi}_\theta \;\equiv\; \psi \;\circ\; \Phi_\theta \;\circ\; \psi^{-1}$ for any $\theta \in \mathrm{U}(1)$. Then, for any $\theta, \theta_1, \theta_2 \in \mathrm{U}(1)$,

- $\tilde{\Phi}_{\theta+2\pi} \;=\; \psi \;\circ\; \Phi_{\theta+2\pi} \;\circ\; \psi^{-1} \;=\; \psi \;\circ\; \Phi_\theta \;\circ\; \psi^{-1} \;=\; \tilde{\Phi}_\theta$
- $\tilde{\Phi}_0 \;=\; \psi \;\circ\; \Phi_0 \;\circ\; \psi^{-1} \;=\; \psi \;\circ\; \mathrm{Id}_M \;\circ\; \psi^{-1} \;=\; \mathrm{Id}_M$
- $\tilde{\Phi}_{\theta_1} \circ \tilde{\Phi}_{\theta_2} = \psi \circ \Phi_{\theta_1} \circ \psi^{-1} \circ \psi \circ \Phi_{\theta_2} \circ \psi^{-1} = \psi \circ \Phi_{\theta_1} \circ \Phi_{\theta_2} \circ \psi^{-1} = \psi \circ \Phi_{\theta_1+\theta_2} \circ \psi^{-1} = \tilde{\Phi}_{\theta_1+\theta_2}$

Therefore, $\tilde{\Phi}_\theta$ is a circle action, and $\theta_0 \in \mathrm{U}(1)$ is such that

$$\tilde{\Phi}_{\theta_0} \;=\; \psi \;\circ\; \Phi_{\theta_0} \;\circ\; \psi^{-1} \;=\; \psi \;\circ\; P_0 \;\circ\; \psi^{-1} \;=\; \tilde{P}_0.$$

As a consequence, $\tilde{P}$ is a nearly-periodic map. □

We also have the following useful factorization result for nearly-periodic maps with limiting rotation within a given conjugacy class:

Lemma 7.4. *Let $\Phi_\theta : M \to M$ be a circle action on a manifold M. Every nearly-periodic map $P_\varepsilon : M \to M$ whose limiting rotation $\Phi'_{\theta_0} = P_0$ is conjugate to Φ_{θ_0} admits the decomposition*

$$\boxed{P_\varepsilon \;=\; I_\varepsilon \;\circ\; \psi \;\circ\; \Phi_{\theta_0} \;\circ\; \psi^{-1},} \tag{7.27}$$

where $\psi : M \to M$ is a diffeomorphism and $I_\varepsilon : M \to M$ is a near-identity diffeomorphism.

We will thus assume that we know in advance the topological type of the circle action Φ_θ for the dynamics of interest, and then propose to learn the nearly-periodic map P_ε by learning each component map in the composition

$$P_\varepsilon \;=\; I_\varepsilon \;\circ\; \psi \;\circ\; \Phi_\theta \;\circ\; \psi^{-1}. \tag{7.28}$$

This formula may be interpreted intuitively as follows. The map ψ learns the mode structure of an oscillatory system's short timescale dynamics. The circle action Φ_θ provides an aliased phase advance for the learnt mode. Finally, I_ε captures the averaged dynamics that occurs on timescales much larger than the limiting oscillation period.

I_ε and ψ can be learnt using any standard neural network architecture, as long as the near-identity property is enforced in the representation for I_ε. It is however important to invert ψ exactly, and this strongly motivates using explicitly invertible neural network architectures for ψ. It has been shown that those coupling-based invertible neural networks are universal diffeomorphism approximators [Teshima et al., 2020]. The parameter θ in the circle action Φ_θ can also be considered as a trainable parameter. We will refer to the resulting neural network architecture as a *gyroceptron*, named after a combination of gyrations of phase with perceptron.

Definition 7.14. *A **gyroceptron** is a feed-forward neural network*

$$\boxed{P_\varepsilon[W] \;=\; I_\varepsilon[W_I] \;\circ\; \psi[W_\psi] \circ\; \Phi_\theta \;\circ\; \psi[W_\psi]^{-1}} \tag{7.29}$$

with weights $W = (W_I, W_\psi)$ and rotation parameter $\theta \in \mathrm{U}(1)$, where

- *$I_\varepsilon[W_I] : M \to M$ is a diffeomorphism for each (ε, W_I), which satisfies $I_0[W_I] = \mathrm{Id}_M$ for all W_I*
- *$\psi[W_\psi] : M \to M$ is a diffeomorphism for each W_ψ*
- *$\Phi_\theta : M \to M$ is a circle action on M*

Gyroceptrons enjoy the following universal approximation property.

Theorem 7.5. *Fix a circle action $\Phi_\theta : M \to M$ and a compact set $C \subset M$. Let $P_\varepsilon : M \to M$ be a nearly-periodic map whose limiting rotation is conjugate to Φ_θ. Let $\psi[W_\psi] : M \to M$ be a feed-forward network architecture that provides a universal approximation within the class of diffeomorphisms, and let $I_\varepsilon[W_I]$ be a feed-forward network architecture that provides a universal approximation within the class of ε-dependent diffeomorphisms with $I_0[W] = \mathrm{Id}_M$. For each $\delta > 0$, there exist weights W_ψ^* and W_I^* such that the gyroceptron*

$$P_\varepsilon[W^*] \;=\; I_\varepsilon[W_I^*] \;\circ\; \psi[W_\psi^*] \;\circ\; \Phi_\theta \;\circ\; \psi[W_\psi^*]^{-1} \tag{7.30}$$

approximates P_ε within δ on C.

7.4.2 Approximating Nearly-Periodic Symplectic Maps via Symplectic Gyroceptrons

We now focus on approximating an arbitrary nearly-periodic symplectic map $P : M \times \mathcal{E} \to M$ on a manifold M. We will restrict our attention to symplectic manifolds with ε-independent symplectic forms (the ε-dependent case is more subtle and will not be pursued here).

From Definition 7.8, there is a corresponding symplectic circle action $\Phi_\theta : M \to M$ and $\theta_0 \in \mathrm{U}(1)$ such that $P_0 = \Phi_{\theta_0}$. As before, we can consider the map $I_\varepsilon : M \to M$ given by

$$I_\varepsilon \;=\; P_\varepsilon \;\circ\; \Phi_{\theta_0}^{-1}, \qquad \forall \varepsilon \in \mathcal{E}. \tag{7.31}$$

Now, the inverse of a symplectic map is symplectic and any composition of symplectic maps is also symplectic. Thus, the mao $\Phi_{\theta_0}^{-1} = P_0^{-1}$ is symplectic, and as a result, I_ε is a symplectic map on M for any $\varepsilon \in \mathcal{E}$ and it satisfies the near-identity property $I_0 = \mathrm{Id}_M$.

By composing both sides of equation (7.31) on the right by Φ_{θ_0}, we obtain a representation for any nearly-periodic symplectic map P as the composition of a near-identity symplectic map and a symplectic circle action:

$$P_\varepsilon \;=\; I_\varepsilon \;\circ\; \Phi_{\theta_0}, \qquad \forall \varepsilon \in \mathcal{E}. \tag{7.32}$$

We will work within symplectic conjugacy classes of circle actions, where two circle actions $\Phi_\theta, \Phi'_\theta$ are said to be symplecticly conjugate if there exists a symplectomorphism ψ (i.e. a symplectic diffeomorphism) such that

$$\Phi'_\theta \;=\; \psi \;\circ\; \Phi_\theta \;\circ\; \psi^{-1} \tag{7.33}$$

We obtain the following useful factorization result for nearly-periodic symplectic maps with limiting rotation within a given symplectic conjugacy class:

Lemma 7.5. *Let $\Phi_\theta : M \to M$ be a symplectic circle action on a symplectic manifold (M, ω). Every nearly-periodic symplectic map $P_\varepsilon : M \to M$ whose limiting rotation $\Phi'_{\theta_0} = P_0$ is symplecticly conjugate to Φ_{θ_0} admits the decomposition*

$$\boxed{P_\varepsilon \;=\; I_\varepsilon \;\circ\; \psi \;\circ\; \Phi_{\theta_0} \;\circ\; \psi^{-1},} \tag{7.34}$$

where $\psi : M \to M$ is a symplectic diffeomorphism and $I_\varepsilon : M \to M$ is a near-identity symplectic diffeomorphism.

As a consequence, if we can approximate any near-identity symplectic map and any symplectic circle action, then by the above representation we can approximate any nearly-periodic symplectic map.

As before, we will assume that we know a priori the topological type of the circle action Φ_θ for the nearly-periodic symplectic system of interest, and work within symplectic conjugation classes.

Since compositions of symplectic maps are symplectic, it follows from Lemma 7.3 that the map

$$\psi \;\circ\; P \;\circ\; \psi^{-1}, \tag{7.35}$$

obtained by conjugation of a nearly-periodic symplectic map P with a a symplectic diffeomorphism ψ is also a nearly-periodic symplectic map. We will then learn the nearly-periodic symplectic map by learning each component map in the composition

$$P_\varepsilon \;=\; I_\varepsilon \;\circ\; \psi \;\circ\; \Phi_\theta \;\circ\; \psi^{-1}, \tag{7.36}$$

where I_ε is a near-identity symplectic map and ψ is symplectic.

The symplectic map ψ can be learnt using any neural network architecture which strongly enforces symplecticity. It is preferable however to choose an architecture which can easily be inverted, so that the computations involving ψ^{-1} can be conducted efficiently. The near-identity symplectic map I_ε can be learnt using any neural network architecture strongly enforcing symplecticity with the additional property that it limits to the identity as ε goes to 0. The parameter θ in the circle action Φ_θ can also be considered as a trainable parameter. We will refer to any such resulting composition of neural network architectures as a *symplectic gyroceptron*.

Definition 7.15. *A **symplectic gyroceptron** is a feed-forward neural network*

$$\boxed{P_\varepsilon[W] \;=\; I_\varepsilon[W_I] \;\circ\; \psi[W_\psi] \;\circ\; \Phi_\theta \;\circ\; \psi[W_\psi]^{-1}} \tag{7.37}$$

with weights $W = (W_I, W_\psi)$ and rotation parameter $\theta \in \mathrm{U}(1)$, where

- *$I_\varepsilon[W_I] : M \to M$ is a symplectic diffeomorphism for each (ε, W_I), which satisfies the near-identity property $I_0[W_I] = \mathrm{Id}_M$ for all W_I*
- *$\psi[W_\psi] : M \to M$ is a symplectic diffeomorphism for each W_ψ*
- *$\Phi_\theta : M \to M$ is a symplectic circle action on M*

Symplectic gyroceptrons enjoy a universal approximation property comparable to the non-symplectic case.

Theorem 7.6. *Fix a symplectic circle action $\Phi_\theta : M \to M$ on the symplectic manifold (M, ω) and a compact set $C \subset M$. Let*

- *$P_\varepsilon : M \to M$ be a nearly-periodic symplectic map whose limiting circle action is symplecticly conjugate to Φ_θ,*
- *$\psi[W_\psi] : M \to M$ be a feed-forward network architecture that provides a universal approximation within the class of symplectic diffeomorphisms,*
- *$I_\varepsilon[W_I]$ be a feed-forward network architecture that provides a universal approximation within the class of ε-dependent symplectic diffeomorphisms, with the near-identity property $I_0[W] = \mathrm{Id}_M$.*

Then, for every $\delta > 0$, there exist weights W_ψ^* and W_I^* for the feed-forward neural networks ψ and I_ε such that the symplectic gyroceptron

$$P_\varepsilon[W^*] \;=\; I_\varepsilon[W_I^*] \;\circ\; \psi[W_\psi^*] \;\circ\; \Phi_\theta \;\circ\; \psi[W_\psi^*]^{-1} \tag{7.38}$$

approximates the nearly-periodic symplectic P_ε within δ on C.

In this chapter, we will use HénonNets [Burby et al., 2020] as the main building blocks of our symplectic gyroceptrons. The symplectic map ψ will be learnt using a standard HénonNet (see Definition 7.3), its inverse ψ^{-1} can be obtained easily by composing inverses of Hénon-like maps (see Remark 7.2), and the near-identity symplectic map I_ε will be learnt using a near-identity HénonNet (see Definition 7.5). The neural network architectures considered in this chapter are summarized in Figure 7.1.

We would like to emphasize that symplectic building blocks other than HénonNets could have been used as the basis for our symplectic gyroceptrons. For instance, a possible option would have been to use SympNets [Jin et al., 2020] since they also strongly ensure symplecticity and enjoy a universal approximation property for symplectic maps. However, numerical experiments conducted in the original HénonNet paper [Burby et al., 2020] suggested that HénonNets have a higher per layer expressive power than SympNets, and as a result SympNets are typically much deeper than HénonNets, and slower for prediction. This is consistent with the observations we will make later in Section 7.6.1 where we will see that a SympNet took 127 seconds to generate trajectories that were generated by a HénonNet of similar size in 3 seconds. Together with the fact that SympNets are not as easily invertible as HénonNets, the computational advantage of HénonNets makes them more desirable as building blocks than SympNets.

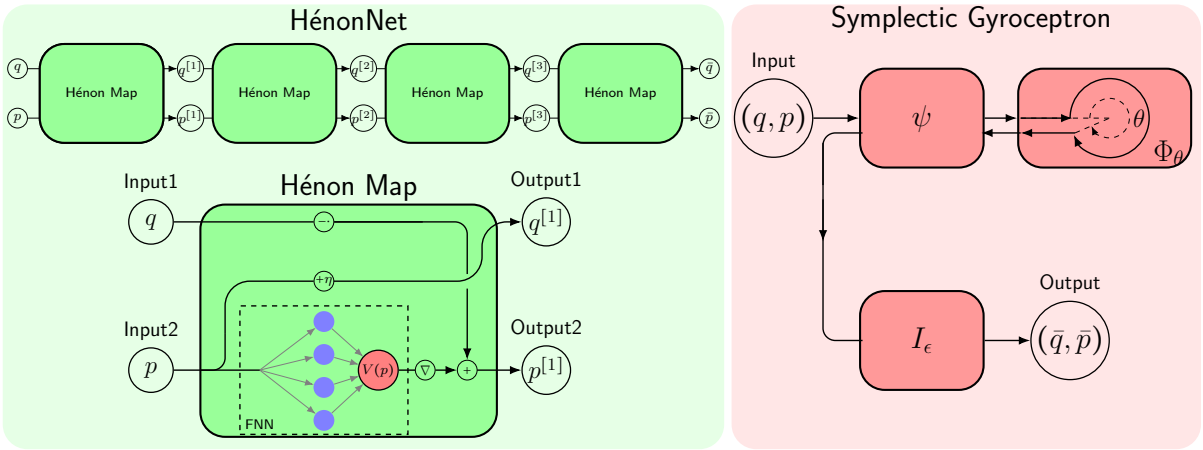


Figure 7.1: Network diagrams. Left: HénonNet. Right: Symplectic Gyroceptron.

7.5 Numerical Confirmation of the Existence of Adiabatic Invariants

In this section, we will confirm numerically that for any random set of weights and bias, the dynamical system generated by the symplectic gyroceptron

$$I_\varepsilon \circ \psi \circ \Phi_\theta \circ \psi^{-1}, \tag{7.39}$$

introduced in Section 7.4.2, admits an adiabatic invariant.

In our numerical experiments, we will take the circle action associated to the clockwise rotation matrix

$$\mathcal{R}_\theta = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}. \tag{7.40}$$

The quantity

$$\mathfrak{I}_0(q,p) = \frac{1}{2}q^2 + \frac{1}{2}p^2 \tag{7.41}$$

is an invariant of the dynamics associated to the circle action (7.40), and as a result

$$\mu = \mathfrak{I}_0 \circ \psi^{-1} \tag{7.42}$$

is an invariant of the dynamics associated to the composition $\psi \circ \Phi_\theta \circ \psi^{-1}$, and an adiabatic invariant of the dynamics associated to the symplectic gyroceptron (7.39).

Figure 7.2 displays the evolution of the adiabatic invariant (7.42) over a sequence of 10000 iterations of the nearly-periodic symplectic map generated by the symplectic gyroceptron (7.39), for different values of ε. Here, ψ is a HénonNet and I_ε a near-identity HénonNet, both with 3 Hénon layers, each of which has 8 neurons in its single-hidden-layer fully-connected neural networks layer potential. We can clearly see from Figure 7.2 that the conservation of the adiabatic invariant gets significantly better as ε gets closer to 0, going from chaotic oscillations of large amplitude when $\varepsilon = 0.1$ to very regular oscillations of minute amplitude when $\varepsilon = 10^{-8}$.

We investigated further by obtaining the number of iterations needed for the adiabatic invariant μ to deviate significantly from its original value μ_0 as the value of the parameter ε is varied. More precisely, given a value of ε, we search for the smallest integer $N(\varepsilon)$ such that

$$|\mu_{N(\varepsilon)} - \mu_0| > \rho \max_{k=0,\ldots,K(\varepsilon)} |\mu_k - \mu_0|, \qquad \text{where } K(\varepsilon) = \lfloor 10 + \varepsilon^{-1/4} \rfloor. \tag{7.43}$$

In other words, we record the first iteration where the value of the adiabatic invariant μ deviates from its original value μ_0 by more than some constant factor $\rho > 1$ of the maximum deviations experienced in the first few $K(\varepsilon)$ iterations. The results are plotted in Figure 7.3 for $\rho = 1.1$.

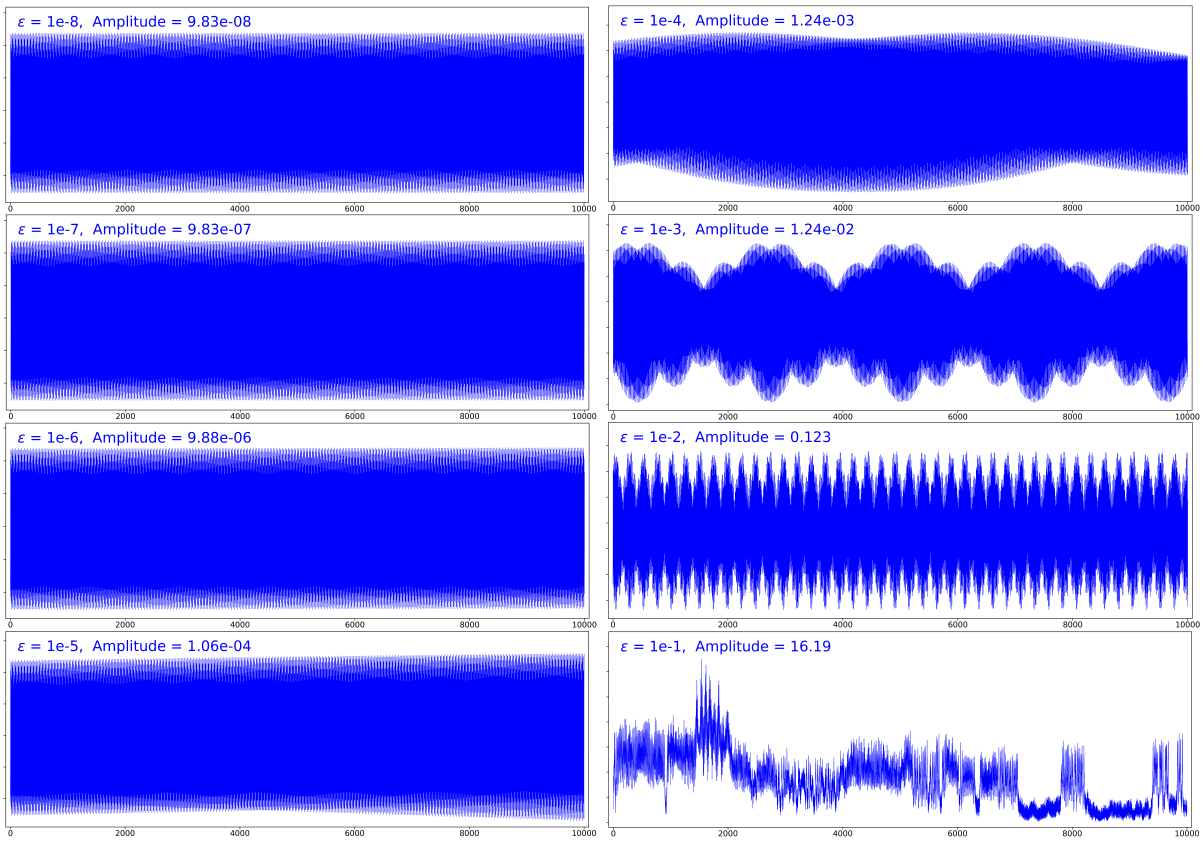


Figure 7.2: Conservation of the adiabatic invariant (7.42) over 10000 iterations of the map generated by the symplectic gyroceptron (7.39) as ε is increased.

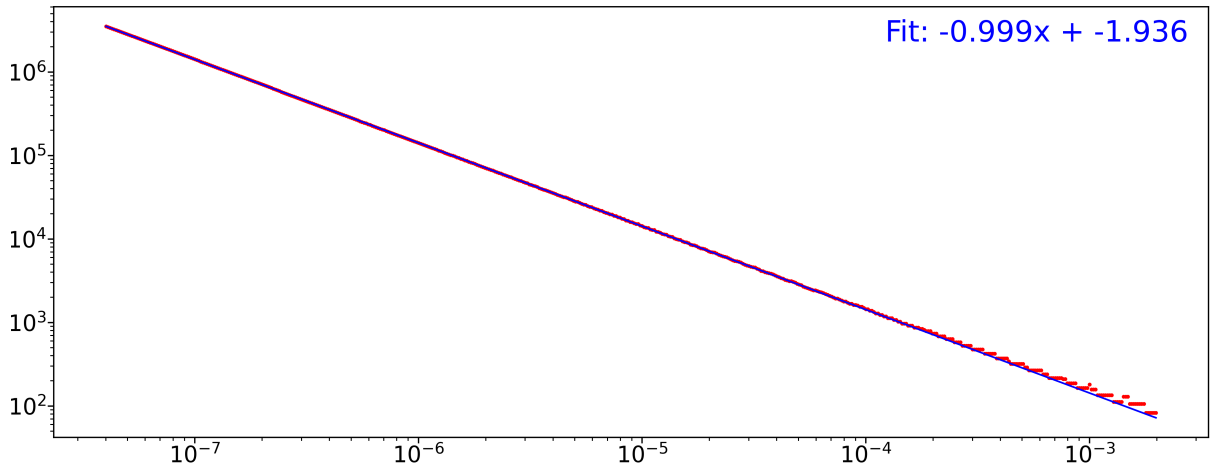


Figure 7.3: $N(\varepsilon)$ as a function of ε for $\rho = 1.1$ and a random set of weights.

We can clearly see from Figure 7.3 that $N(\varepsilon)$, the number of iterations needed for the adiabatic invariant μ to deviate from its original value μ_0 by more than $\rho = 1.1$ times the maximum deviations experienced in the first few iterations, increases sharply as ε gets closer to 0. This is consistent with theoretical expectations. Note that using higher values of ρ and smaller values of ε would probably generate more interesting and meaningful results. Unfortunately, this is not computationally realizable since $N(\varepsilon)$ becomes very large when ρ is increased beyond 1.2. Even for larger values of ε, computing a single point would take several days.

7.6 Numerical Examples of Learning Surrogate Maps

7.6.1 Nonlinearly Coupled Oscillators

In this section, we use the symplectic gyroceptron architecture introduced in Section 7.4.2 to learn a surrogate map for the nearly-periodic symplectic flow map associated to a nearly-periodic Hamiltonian system composed of two nonlinearly coupled oscillators, where one of them oscillates significantly faster than the other:

$$\begin{cases} \dot{q}_1 = p_1 & \dot{p}_1 = -q_1 - \varepsilon \partial_{q_1} U(q_1, q_2) \\ \dot{q}_2 = \varepsilon p_2 & \dot{p}_2 = -\varepsilon q_2 - \varepsilon \partial_{q_2} U(q_1, q_2) \end{cases} \tag{7.44}$$

These equations of motion are the Hamilton's equations associated to the Hamiltonian

$$H_\varepsilon(q_1, q_2, p_1, p_2) = \frac{1}{2}(q_1^2 + p_1^2) + \frac{1}{2}\varepsilon(q_2^2 + p_2^2) + \varepsilon U(q_1, q_2). \tag{7.45}$$

The limiting $\varepsilon = 0$ dynamics are decoupled, where the first oscillator, initialized at $(q_1(0), p_1(0)) = (q, p)$, follows the periodic trajectory

$$q_1(t) = q\cos t + p\sin t, \qquad p_1(t) = p\cos t - q\sin t, \tag{7.46}$$

characterized by periodic clockwise circular rotation in phase space, while the second oscillator remains immobile. This is therefore a nearly-periodic Hamiltonian system on $\mathbb{R}^4$ with associated $\varepsilon = 0$ circle action given by the clockwise rotation $\mathcal{R}_\theta$:

$$\mathcal{R}_\theta = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{7.47}$$

We will use the nonlinear coupling potential

$$U(q_1, q_2) = q_1 q_2 \sin(2q_1 + 2q_2) \tag{7.48}$$

in our numerical experiments since the resulting nearly-periodic Hamiltonian system displays complicated dynamics as ε is increased from 0. We have plotted in Figure 7.4 a few trajectories of this dynamical system corresponding to different values of ε.

To learn a surrogate map for the nearly-periodic symplectic flow map associated to this nearly-periodic Hamiltonian system, we use the symplectic gyroceptron

$$I_\varepsilon \circ \psi \circ \Phi_\theta \circ \psi^{-1}, \tag{7.49}$$

introduced in Section 7.4.2. In our numerical experiments, $\varepsilon = 0.01$, θ is a trainable parameter, and

- ψ is a HénonNet with 10 Hénon layers each of which has 8 neurons in its single-hidden-layer fully-connected neural network layer potential
- I_ε is a near-identity HénonNet with 8 Hénon layers each of which has 6 neurons in its single-hidden-layer fully-connected neural network layer potential

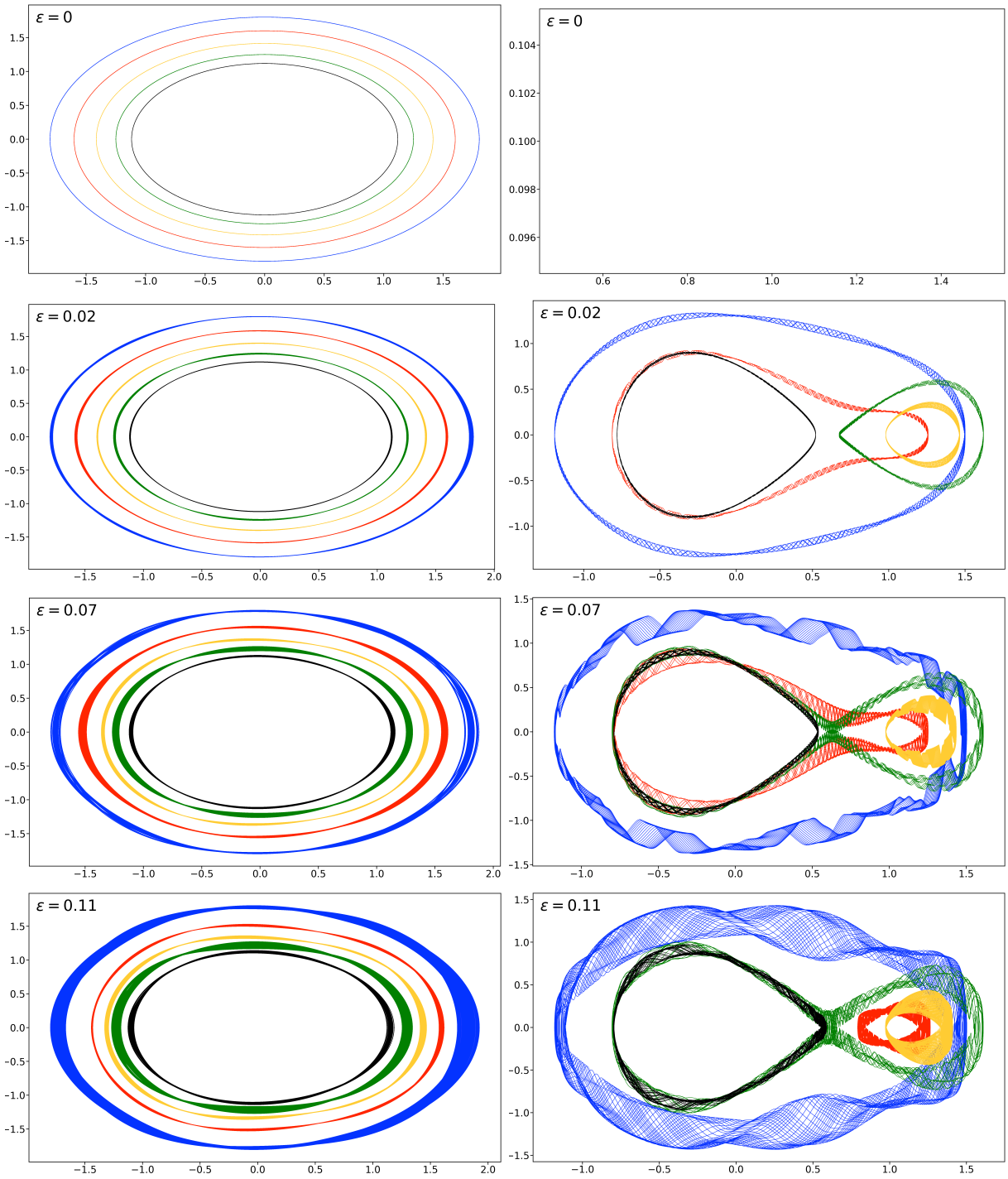


Figure 7.4: Sample trajectories of the first oscillator in (q_1, p_1) phase space (on the left) and of the second oscillator in (q_2, p_2) phase space (on the right) for the nearly-periodic Hamiltonian system (7.44) as ε is increased.

The resulting symplectic gyroceptron of 549 trainable parameters was trained for a few thousands epochs on a dataset of 60,000 updates $(q_1, q_2, p_1, p_2) \mapsto (\tilde{q}_1, \tilde{q}_2, \tilde{p}_1, \tilde{p}_2)$ of the time-0.05 flow map associated to the nearly-periodic Hamiltonian system (7.44). The training data was generated using the classical Runge–Kutta 4 integrator with very small time-steps, and the Mean Squared Error was used as the loss function in the training.

Figure 7.6 shows the dynamics predicted by the symplectic gyroceptron for seven different initial conditions with the same initial values of (q_1, p_1) against the reference trajectories generated by the classical Runge–Kutta 4 integrator with very small time-steps. We only display the trajectories of the second oscillator since the motion of the first oscillator follows a simple nearly-circular curve.

We can see from Figure 7.6 that the dynamics learnt by the symplectic gyroceptron match almost perfectly the reference trajectories and from Figure 7.5 that these trajectories follow the level sets of the averaged Hamiltonian

$$\bar{H} = \frac{1}{2\pi} \int_0^{2\pi} \Phi_\theta^* H \, d\theta. \tag{7.50}$$

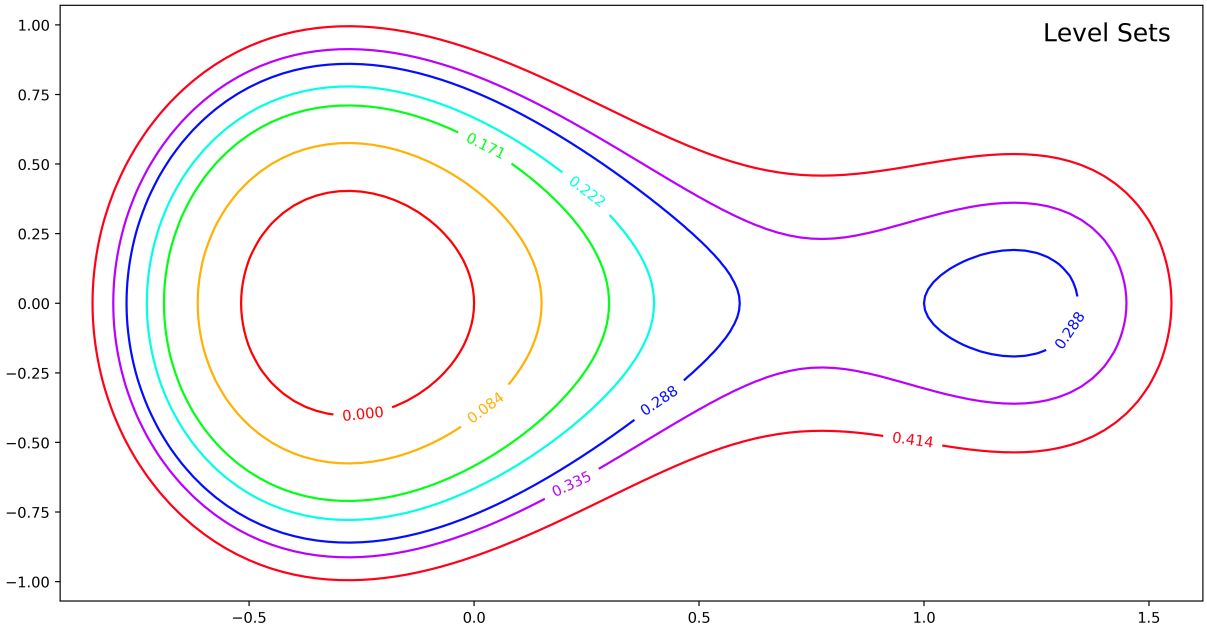


Figure 7.5: Level sets of the averaged Hamiltonian (7.53).

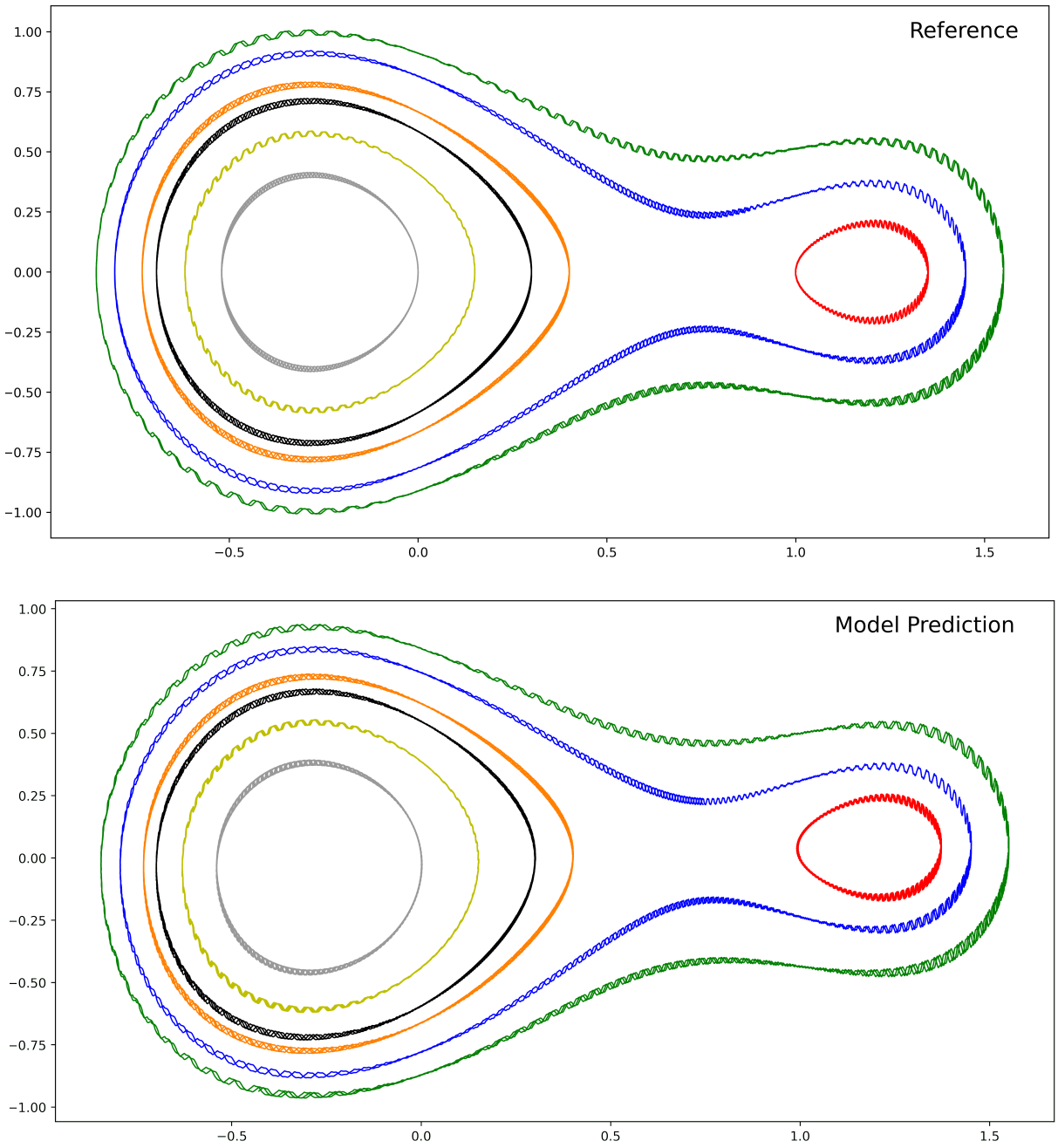


Figure 7.6: Symplectic gyroceptron predictions against the reference trajectories in (q_2, p_2) phase space for the second oscillator in the nearly-periodic Hamiltonian system (7.44) with $\varepsilon = 0.01$ and a time-step of 0.05.

Up to an unimportant constant, the averaged Hamiltonian is given by

$$\bar{H}(q_2, p_2) = \frac{1}{2}(q_2^2 + p_2^2) + \frac{1}{2\pi}\int_0^{2\pi} U(q_1(t), q_2)dt \tag{7.51}$$

$$= \frac{1}{2}(q_2^2 + p_2^2) + \frac{q_2}{2\pi}\int_0^{2\pi} (q\cos t + p\sin t)\sin\left(2\left[q\cos t + p\sin t\right] + 2q_2\right)dt \tag{7.52}$$

$$= \frac{1}{2}(q_2^2 + p_2^2) + q_2\cos(2q_2)\sqrt{q^2+p^2}\ \mathcal{J}_1\left(2\sqrt{q^2+p^2}\right) \tag{7.53}$$

where $\mathcal{J}_1(x)$ is the first order Bessel function of the first kind. Using Kruskal's theory of nearly-periodic systems, it is straightforward to show that this averaged Hamiltonian is the leading-order approximation of the Hamiltonian for the formal $U(1)$-reduction of the two-oscillator system.

We also learned a surrogate map for the nearly-periodic symplectic time-5 flow map associated to the nearly-periodic Hamiltonian system (7.44), using a symplectic gyroceptron where $\varepsilon = 0.01$, θ is a trainable parameter, and ψ and I_ε both have 10 Hénon layers each of which has 8 neurons in its single-hidden-layer fully-connected neural network layer potential. This symplectic gyroceptron of 681 trainable parameters was trained for a few thousands epochs on a dataset of 60,000 updates $(q_1, q_2, p_1, p_2) \mapsto (\tilde{q}_1, \tilde{q}_2, \tilde{p}_1, \tilde{p}_2)$. For comparison, we also trained a HénonNet and a SympNet of similar sizes and ran simulations from the same seven different initial conditions. The HénonNet used has 16 layers each of which has 10 neurons in its single-hidden-layer fully-connected neural network layer potential, for a total of 672 trainable parameters. The SympNet used has 652 trainable parameters in a network structure of the form $\mathcal{L}_n^{(k+1)} \circ (\mathcal{N}_{\text{up/low}} \circ \mathcal{L}_n^{(k)}) \circ \ldots \circ (\mathcal{N}_{\text{up/low}} \circ \mathcal{L}_n^{(1)})$, where each $\mathcal{L}_n^{(k)}$ is the composition of n trainable linear symplectic layers, and $\mathcal{N}_{\text{up/low}}$ is a non-trainable symplectic activation map.

Figure 7.7 shows the dynamics predicted by the symplectic gyroceptron, the HénonNet, and the SympNet, for seven different initial conditions with the same initial values of (q_1, p_1) against the reference trajectories generated by the Runge–Kutta 4 integrator (RK4) with small time-steps. As before, we only display the trajectories of the second oscillator. We can see that the dynamics predicted by the symplectic gyroceptron match the reference trajectories very well, although the predicted oscillations around the level sets of the averaged Hamiltonian are unsurprisingly larger than when learning the time-0.05 flow map.

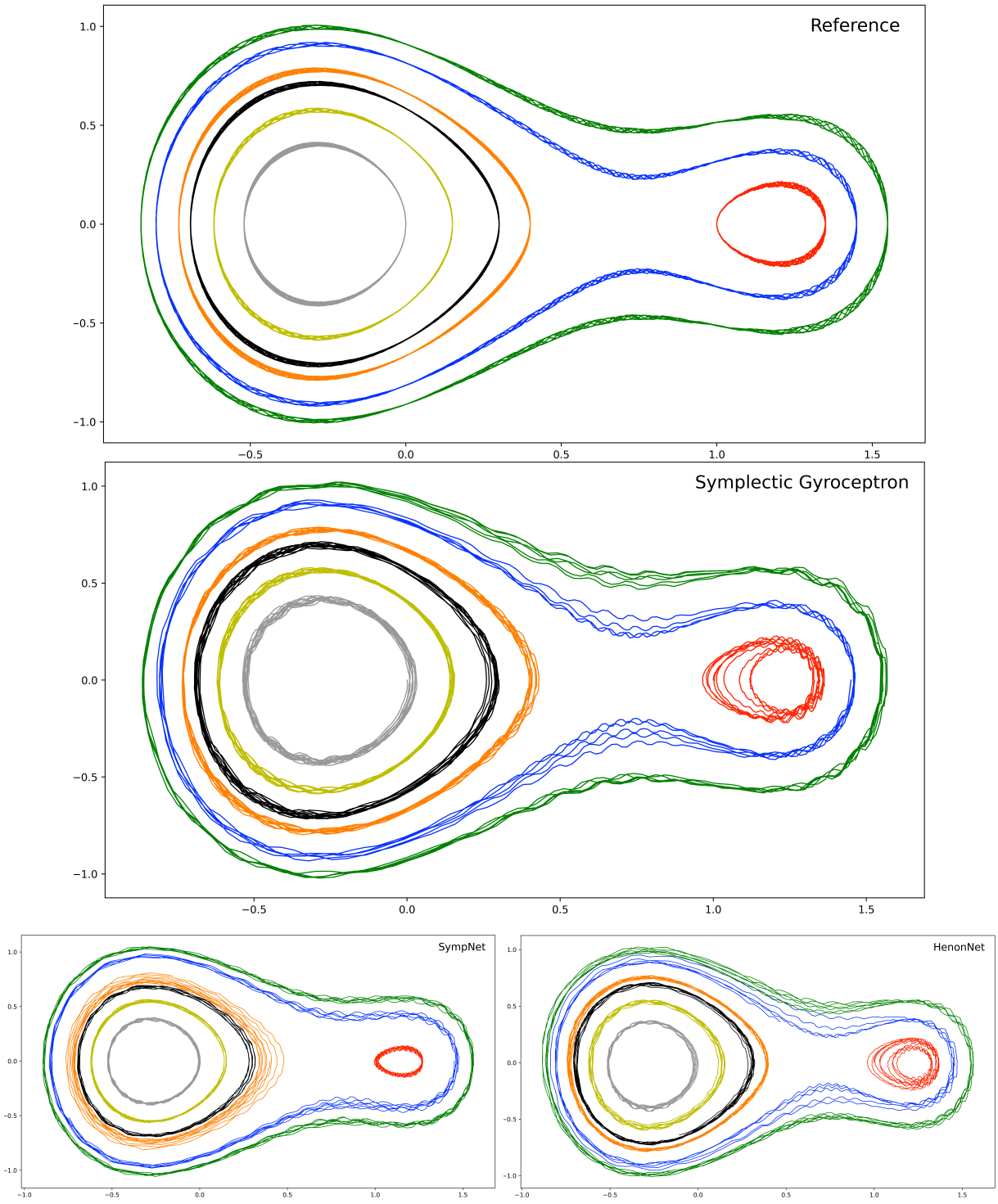


Figure 7.7: Predictions from a Symplectic Gyroceptron, a SympNet, and a HenonNet, against the reference trajectories for the second oscillator in the nearly-periodic Hamiltonian system (7.44) with $\varepsilon = 0.01$ and the larger time-step of 5.

The crucial advantage that the symplectic gyroceptron architecture offer over the other architectures considered, which only enforce the symplectic constraint, is provable existence of an adiabatic invariant. After training, the other architectures may empirically display preservation of an adiabatic invariant, but this cannot be proved rigorously from first principles. In contrast, the symplectic gyroceptron enjoys provable existence of an adiabatic invariant before, during, and after training.

Note that the symplectic gyroceptron generated the seven trajectories in 5 seconds, which is several orders of magnitude faster than RK4 with small time-steps which took 6,055 seconds. The HénonNet allowed to simulate the dynamics slightly faster, in 3 seconds, while the SympNet was much slower with a running time of 127 seconds, consistently with the observations made in the original HénonNet paper [Burby et al., 2020] which motivated choosing HénonNets over SympNets in the symplectic gyroceptrons.

7.6.2 Charged Particle Interacting with its Self-Generated Electromagnetic Field

Problem Formulation

Next we test the ability of symplectic gyroceptrons to function as surrogates for higher-dimension nearly-periodic systems, and for systems where the limiting circle action is not precisely known.

To formulate the ground-truth dynamical system, first fix a positive integer K and a sequence of single-variable functions $V_k : \mathbb{R} \to \mathbb{R}$, $k = 1, \dots, K$. We consider the canonical Hamiltonian system on $\mathbb{R}^2 \times (\mathbb{R}^2)^K$ with coordinates $(q, p, Q_1, P_1, \dots, Q_K, P_K)$, defined by the Hamiltonian

$$H_\epsilon \;=\; \frac{1}{2}\epsilon\left(p - \sum_{k=1}^{K} \sin(kq)\, Q_k\right)^2 \;+\; \frac{1}{2}\sum_{k=1}^{K} k\left([P_k - V_k(Q_k)]^2 + Q_k^2\right). \tag{7.54}$$

The equations of motion are

$$\dot{q} = \partial_p H_\epsilon = \epsilon \left(p - \sum_{\ell=1}^{K} \sin(\ell q)\, Q_\ell \right), \tag{7.55}$$

$$\dot{Q}_k = \partial_{P_k} H_\epsilon = k\,(P_k - V_k(Q_k)), \tag{7.56}$$

$$\dot{p} - \partial_q H_\epsilon = \epsilon \left(p - \sum_{\ell=1}^{K} \sin(\ell q)\, Q_\ell \right) \sum_{m=1}^{K} m\, \cos(mq)\, Q_m, \tag{7.57}$$

$$\dot{P}_k = -\partial_{Q_k} H_\epsilon = -k\, Q_k + k\,(P_k - V_k(Q_k))\, V_k'(Q_k) + \epsilon \left(p - \sum_{\ell=1}^{K} \sin(\ell q)\, Q_\ell \right) \sin(kq), \tag{7.58}$$

These equations of motion may be regarded as a simplified model of a charged particle (q, p) interacting with its self-generated electromagnetic field $(Q_1, P_1, \ldots, Q_K, P_K)$. We will describe the application of symplectic gyroceptrons to the development of a dynamical surrogate for this system when $\epsilon \ll 1$.

First, we verify that this Hamiltonian system is nearly-periodic, since this is the type of dynamical systems that symplectic gyroceptrons are designed to handle. So consider the limiting dynamics when $\epsilon = 0$. The equations of motion reduce to

$$\dot{q} = 0, \qquad \dot{Q}_k = \partial_{P_k} H_\epsilon = k\,(P_k - V_k(Q_k)), \tag{7.59}$$

$$\dot{p} = 0, \qquad \dot{P}_k = -\partial_{Q_k} H_\epsilon = -k\, Q_k + k\,(P_k - V_k(Q_k))\, V_k'(Q_k). \tag{7.60}$$

While these equations of motion may appear impenetrable at first glance, the symplectic transformation of variables given by

$$\Lambda_0^{-1} : (q, p, Q_1, P_1, \ldots, Q_K, P_K) \mapsto (q, p, Q_1, \Pi_1, \ldots, Q_K, \Pi_K) \tag{7.61}$$

where $\Pi_k = P_k - V_k(Q_k)$ simplifies them dramatically into

$$\dot{q} = 0, \qquad \dot{p} = 0, \qquad \dot{Q}_k = k\,\Pi_k, \qquad \dot{\Pi}_k = -k\, Q_k, \tag{7.62}$$

which correspond to a family of harmonic oscillators parametrized by their angular frequencies k. The solution map in these nice variables is therefore

$$\Phi_t^0(q, p, Q_1, \Pi_1, \ldots, Q_K, \Pi_K) = (q, p, Q_1(t), \Pi_1(t), \ldots, Q_K(t), \Pi_K(t)), \tag{7.63}$$

where

$$Q_k(t) = \cos(k\,t)\, Q_k + \sin(k\,t)\, \Pi_k, \qquad \Pi_k(t) = -\sin(k\,t)\, Q_k + \cos(k\,t)\, \Pi_k. \tag{7.64}$$

Note that Φ_t^0 is periodic with minimal period 2π. The solution map in terms of the original variables (Q_k, P_k) is therefore $\Phi_t = \Lambda_0 \circ \Phi_\theta^0 \circ \Lambda_0^{-1}$. Since Φ_t is periodic in t with minimal period 2π the ground-truth equations are Hamiltonian nearly-periodic. The leading-order adiabatic invariant is

$$\mu_0 \;=\; \frac{1}{2}\sum_{k=1}^{K} k\,(\Pi_k^2 + Q_k^2) \;=\; \frac{1}{2}\sum_{k=1}^{K} k\,([P_k - V_k(Q_k)]^2 + Q_k^2). \tag{7.65}$$

Symplectic gyroceptrons are therefore well-suited to surrogate modeling for this system.

Numerical Experiments

Here, we learn the nearly-periodic Hamiltonian system (7.54) in the 6-dimensional case (i.e., $K = 2$) with

$$V_1(Q_1) = \frac{1}{2}\sin(2Q_1) \qquad \text{and} \qquad V_2(Q_2) = \frac{1}{2}\exp(-5Q_2^2). \tag{7.66}$$

In our symplectic gyroceptron $I_\varepsilon \circ \psi \circ \Phi_\theta \circ \psi^{-1}$, the circle action Φ_θ is taken to be the rotation in equation (7.64) with θ treated as a trainable parameter, and the HénonNets ψ and I_ε both have 12 Hénon layers each of which has 8 neurons in its single-hidden-layer fully-connected neural network layer potential. The resulting architecture of 1,033 trainable parameters was trained for a few thousands epochs on a dataset of 50,000 updates $(q, p, Q_1, P_1, Q_2, P_2) \mapsto (\tilde{q}, \tilde{p}, \tilde{Q}_1, \tilde{P}_2, \tilde{Q}_1, \tilde{P}_2)$.

To verify visually that we have learnt the dynamics successfully, we select initial conditions on the zero level set of the adiabatic invariant μ_0. There, dynamics should remain on that slow manifold which is lower-dimensional and thus more easily portrayed. For the Hamiltonian system (7.54), the slow manifold is the zero level set of $\mu_0 = 0$, which we can see from equation (7.65), is the set of points $(q, p, Q_1, P_1, Q_2, P_2)$ such that $Q_1 = Q_2 = 0$ and $P_1 = V_1(Q_1) = V_1(0),\ P_2 = V_2(Q_2) = V_2(0)$. On that slow manifold, the dynamics reduce to

$$\dot{q} = \epsilon p, \qquad \dot{p} = 0 \qquad \dot{Q}_1 = 0, \qquad \dot{Q}_2 = 0, \qquad \dot{P}_1 = \epsilon p \sin(q), \qquad \dot{P}_2 = \epsilon p \sin(2q), \tag{7.67}$$

where in particular the (q, p) dynamics are now independent of (Q_1, Q_2, P_1, P_2) and can easily be solved for explicitly, given some initial conditions $(q(0), p(0)) = (q, p)$:

$$q(t) = q + \epsilon p t, \qquad p(t) = p. \tag{7.68}$$

Figures 7.8 a)b) show that the trained symplectic gyroceptron generates predictions for the evolution of q and p which remain very close to the true trajectories on the slow manifold when the initial conditions are selected on the zero level set of μ_0.

We also generate dynamics outside the zero level set of μ_0 and verify that the quantity $\mathfrak{I}_0 \circ \psi^{-1}$ matches the learnt adiabatic invariant μ_0^{learnt} along the trajectories generated by the symplectic gyroceptron $I_\varepsilon \circ \psi \circ \Phi_\theta \circ \psi^{-1}$, where

$$\mu_0^{learnt}(q,p,Q_1,P_1,Q_2,P_2) \;=\; \frac{1}{2}\sum_{k=1}^{K=2} k\left([P_k - V_k(Q_k)]^2 + Q_k^2\right), \tag{7.69}$$

and

$$\mathfrak{I}_0(q,p,Q_1,\Pi_1,Q_2,\Pi_2) \;=\; \frac{1}{2}\sum_{k=1}^{K=2} k\left(\Pi_k^2 + Q_k^2\right). \tag{7.70}$$

More precisely, we check whether

$$\mathfrak{I}_0 \;\circ\; \psi^{-1} \;=\; \mu_0^{learnt} \tag{7.71}$$

with both quantities being approximately constant along trajectories generated by the symplectic gyroceptron, where

$$\mathfrak{I}_0 \;\circ\; \psi^{-1}(q,p,Q_1,P_1,Q_2,P_2) \;=\; \frac{1}{2}\sum_{k=1}^{K=2} k\left(\tilde{\Pi}_k^2 + \tilde{Q}_k^2\right), \tag{7.72}$$

with

$$(\tilde{q},\tilde{p},\tilde{Q}_1,\tilde{\Pi}_k,\tilde{Q}_2,\tilde{\Pi}_2) \;=\; \psi^{-1}(q,p,Q_1,P_1,Q_2,P_2). \tag{7.73}$$

From Figure 7.8 c), we see that along trajectories which are not started on the zero level set of μ_0, the value of $\mathfrak{I}_0 \circ \psi^{-1}$ remains very close to the approximately constant quantity μ_0^{learnt}, although $\mathfrak{I}_0 \circ \psi^{-1}$ displays small oscillations. Since the quantity $\mathfrak{I}_0 \circ \psi^{-1}$ is an adiabatic invariant for the neural network these oscillations will remain bounded in amplitude for very large time intervals. The amplitude can in principle be reduced by finding a more optimal set of weights for the neural network, but it can never be reduced to zero since the true adiabatic invariant is not exactly conserved (oscillations in μ_0 are not visible at the scales displayed in the plot).

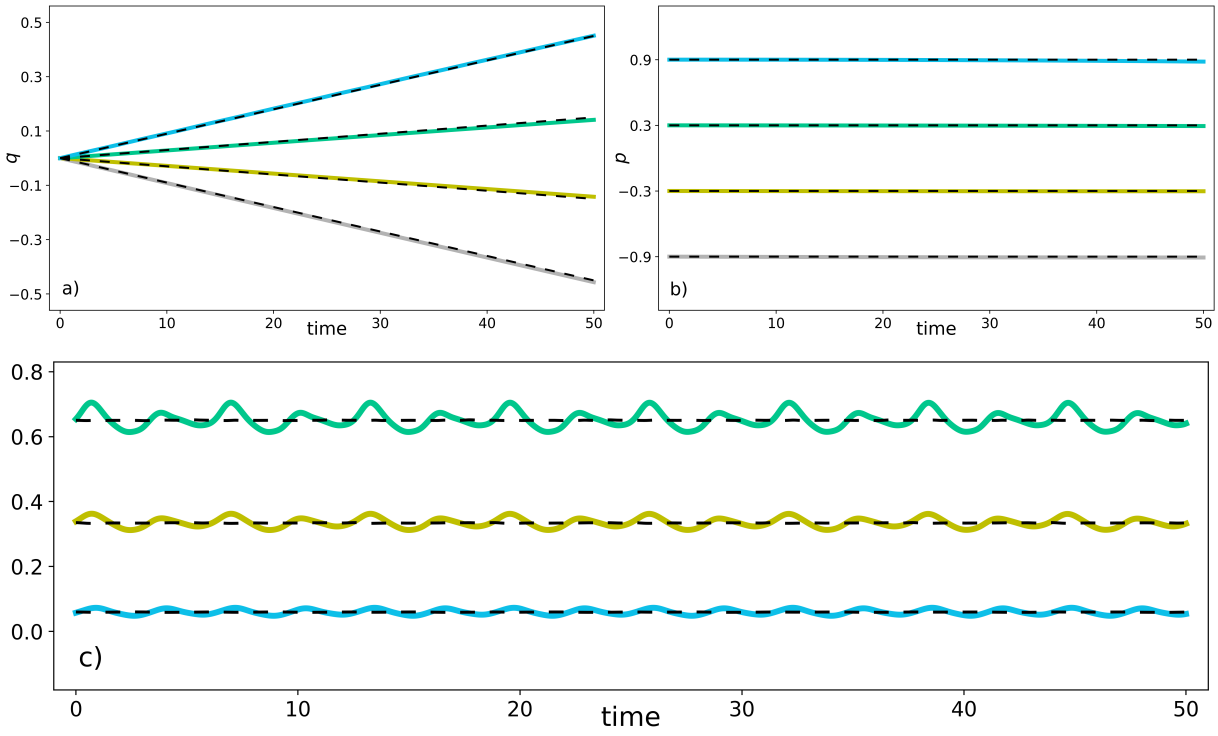


Figure 7.8: **a) b)** Symplectic gyroceptron predictions (in colors) against the true trajectories (dashed lines) with four different choices of initial conditions on the zero level set of the adiabatic invariant μ_0 for the nearly-periodic Hamiltonian system (7.54) with $\varepsilon = 0.01$. **c)** Evolution of $\mathfrak{I}_0 \circ \psi^{-1}$ (in colors) and μ_0^{learnt} (dashed lines) along trajectories generated by the symplectic gyroceptron with three different choices of initial conditions for the nearly-periodic Hamiltonian system (7.54) with $\varepsilon = 0.01$.

Conclusion

In this chapter, we have successfully constructed novel structure-preserving neural network architectures, gyroceptrons and symplectic gyroceptrons, to learn nearly-periodic maps and nearly-periodic symplectic maps, respectively. By construction, these proposed architectures define nearly-periodic maps, and symplectic gyroceptrons also preserve symplecticity. Furthermore, it was confirmed experimentally that in the symplectic case, the maps generated by the proposed symplectic gyroceptrons admit discrete-time adiabatic invariants, regardless of the value of their parameters and weights.

We also demonstrated that the proposed architectures can be effectively used in practice, by learning very precisely surrogate maps for the nearly-periodic symplectic flow maps associated to two different nearly-periodic Hamiltonian systems. Note that the hyperparameters in our architectures have not been optimized to maximize the quality of our training outcomes, and future applications of this architecture may benefit from further hyperparameter tuning.

Symplectic gyroceptrons provide a promising strategy for surrogate modeling of non-dissipative dynamical systems that automatically steps over short timescales without introducing spurious instabilities, and could have potential future applications for the Klein–Gordon equation in the weakly-relativistic regime, for charged particles moving through a strong magnetic field, and also for the rotating inviscid Euler equations in quasi-geostrophic scaling [Cotter and Reich, 2004]. Symplectic gyroceptrons could also be used for structure-preserving simulation of non-canonical Hamiltonian systems on exact symplectic manifolds [Burby et al., 2023], which have numerous applications across the physical sciences, for instance in modeling weakly-dissipative plasma systems [Morrison, 1980; Morrison and Greene, 1980; Morrison and Kotschenreuther, 1989; Morrison, 1998; Burby et al., 2015; Morrison and Vanneste, 2016; Burby, 2022].

Symplectic gyroceptrons aims to solve surrogate modeling problems, where the dynamical system of interest is known but slow or expensive to simulate. In principle, symplectic gyroceptrons could also be used to discover dynamical models from observational data without detailed knowledge of the underlying dynamical system. However, in order to apply symplectic gyroceptrons effectively in this context data-mining methods must be developed for learning the topological conjugacy class of the limiting circle action. Given a topological classification of circle actions on the relevant state space (e.g. see [Raymond, 1968] for the case of a 3-dimensional state space), a straightforward approach would be to test an ensemble of topologically-distinct circle actions for best results. A more nuanced approach would use the observed dynamics to estimate values for the classifying topological invariants of a circle action. This topological learning problem warrants further investigation.

➢ Chapter 7 contains original material from

① "Approximation of Nearly-Periodic Symplectic Maps via Structure-Preserving Neural Networks" by V. Duruisseaux, J. W. Burby, and Q. Tang. *Scientific Reports,* Vol.13, No.8351, 2023

The dissertation author was the primary investigator and author of this paper.

Part III Conclusion

Identifying accurate and efficient dynamic surrogate models based on observed trajectories is critical for fast prediction and design of control laws that ensure desirable properties such as safety, stability, and generalization to different operational conditions. To circumvent some of the limitations experienced by first principles modeling and naive deep learning architectures, structure-preserving machine learning encodes physics laws and geometric properties of the dynamical systems in deep learning techniques to design algorithms with improved efficiency and generalization capacity. Preserving the underlying geometric properties of a dynamical system (such as symplecticity for Hamiltonian systems) when simulating its dynamics has numerous benefits, which have been extensively studied, so it is reasonable to seek structure-preserving architectures for dynamics learning.

Here, we considered two specific classes of dynamical systems. We introduced LieFVINs to learn surrogates for the flow maps of controlled Lagrangian or Hamiltonian dynamics evolving on Lie groups, while strongly preserving both the Lie group structure on which the dynamics evolve and the symplectic structure underlying the controlled systems of interest. We also introduced symplectic gyroceptrons to approximate nearly-periodic symplectic maps while preserving symplecticity and near-periodicity.

A possible future direction is to extend and scale up these approaches to tackle more complicated dynamical systems which arise in practice. Symplectic gyroceptrons could be used in numerous contexts, including relativistic systems, electromagnetic systems, fluid flows, and weakly-dissipative plasma systems. LieFVINs could be designed for more complicated robot systems such as multi-link robots and multi-agent systems.

Another possible research direction is to carefully design novel structure-preserving architectures for many more different classes of dynamical systems. These could be reversible systems, dynamics possessing invariants of motion (such as energy, volume, momenta, and flux), periodic and highly oscillatory systems, dynamics constrained to evolve on manifolds and Lie groups, and combinations of these structures. This could also involve identifying new geometric structures whose preservation leads to significant benefits for certain specific applications of interest.

Appendix

A Proofs

A.1 HTVI Error Analysis Theorem 2.6

The proof of Theorem 2.6 is similar to the one presented in the Appendix of [Schmitt et al., 2018] for Lagrangian Taylor variational integrators. We first start with the right Hamiltonian Taylor variational integrator case.

Let $q(t)$ and $p(t)$ denote the solutions of Hamilton's boundary value problem

$$\dot{q}(t) = g(q(t), p(t), t), \qquad \dot{p}(t) = f(q(t), p(t), t), \qquad q(0) = q_0, \quad p(h) = p_1,$$

and let $q_1 = q(h)$ and $p_0 = p(0)$.

Lemma A.1. *Given a r-order Taylor method $\Psi_h^{(r)}$ approximating the exact time-h flow map corresponding to Hamilton's equations, let $\tilde{p}_0$ solve the problem $p_1 = \pi_{T^*Q} \circ \Psi_h^{(r)}(q_0, \tilde{p}_0)$. Then,*

$$\tilde{p}_0 = p_0 + \mathcal{O}(h^{r+1}).$$

Proof. Solving the equation $p_1 = \pi_{T^*Q} \circ \Psi_h^{(r)}(q_0, \tilde{p}_0)$ for $\tilde{p}_0$ yields

$$\tilde{p}_0 = p_1 - \sum_{k=1}^{r} \frac{h^k}{k!} f^{(k-1)}(q_0, \tilde{p}_0, 0).$$

The exact solution $p(t)$ belongs to $C^{r+1}([0, h])$ so Taylor's Theorem gives

$$p_0 = p_1 - \sum_{k=1}^{r} \frac{h^k}{k!} f^{(k-1)}(q_0, p_0, 0) + R_r(h).$$

Now, since $p(t)$ belongs to $C^{r+1}([0,h])$, $f^{(k-1)}$ is Lipschitz continuous in its arguments for $k = 1, ..., r-1$. Let M be the largest of the corresponding $(r-1)$ Lipschitz constants with respect to the second argument over the compact interval $[0, C]$. Then, using the triangle inequality,

$$\begin{aligned}\|\tilde{p}_0 - p_0\| &= \left\| R_r(h) - \sum_{k=1}^{r} \frac{h^k}{k!} \left[f^{(k-1)}(q_0, \tilde{p}_0, 0) - f^{(k-1)}(q_0, p_0, 0) \right] \right\| \\ &\leq M \sum_{k=1}^{r} \frac{h^k}{k!} \|\tilde{p}_0 - p_0\| + \|R_r(h)\|.\end{aligned}$$

Thus,

$$\left(1 - M \sum_{k=1}^{r} \frac{h^k}{k!}\right) \|\tilde{p}_0 - p_0\| \leq \|R_r(h)\| = \mathcal{O}(h^{r+1}),$$

and by continuity, $\exists \tilde{C} \in (0, C)$ such that $\forall h \in (0, \tilde{C})$, the term $\left(1 - M \sum_{k=1}^{r} \frac{h^k}{k!}\right)$ is bounded away from zero, which concludes the proof. □

We now show that starting the r-order Taylor method with initial conditions $(q_0, \tilde{p}_0)$ rather than (q_0, p_0) will not affect the order of accuracy of the method.

Lemma A.2. *The r-order Taylor method $\Psi_h^{(r)}$ with initial conditions $(q_0, \tilde{p}_0)$ and where $\tilde{p}_0$ solves $p_1 = \pi_{T^*Q} \circ \Psi_h^{(r)}(q_0, \tilde{p}_0)$ is accurate to at least $\mathcal{O}(h^{r+1})$ for the Hamiltonian boundary-value problem.*

Proof. Let $(\tilde{q}(t), \tilde{p}(t))$ denote the exact solution to Hamiltonian's equations with initial values $(q_0, \tilde{p}_0)$, and let $(q_d(t), p_d(t))$ denote the values generated by the r-order Taylor method with initial conditions $(q_0, \tilde{p}_0)$. The Hamiltonian initial-value problem is well-posed, so denoting the Lipschitz constant with respect to the second argument by M, we get

$$\begin{aligned}\|(q(t), p(t)) - (q_d(t), p_d(t))\| &\leq \|(q(t), p(t)) - (\tilde{q}(t), \tilde{p}(t))\| + \|(\tilde{q}(t), \tilde{p}(t)) - (q_d(t), p_d(t))\| \\ &\leq M\|p_0 - \tilde{p}_0\| + \mathcal{O}(h^{r+1}) \leq \mathcal{O}(h^{r+1}),\end{aligned}$$

where we have used the triangle inequality, and the fact that the local truncation error of a r-order Taylor method is $\mathcal{O}(h^{r+1})$ to bound $\|(\tilde{q}(t), \tilde{p}(t)) - (q_d(t), p_d(t))\|$. □

We are now ready to prove Theorem 2.6 for right HTVIs.

Theorem A.1. *Consider a Hamiltonian H such that H and $\frac{\partial H}{\partial p}$ are Lipschitz continuous in both variables. Given a r-order accurate Taylor method $\Psi_h^{(r)}$ and a s-order accurate quadrature formula with weights and nodes (b_i, c_i), define the associated Taylor discrete right Hamiltonian*

$$H_d^+(q_0, p_1; h) = p_1^\intercal \tilde{q}_1 - h \sum_{i=1}^{m} b_i \left[p_{c_i}^\intercal \dot{q}_{c_i} - H(q_{c_i}, p_{c_i}) \right],$$

where $\tilde{p}_0$ solves $p_1 = \pi_{T^*Q} \circ \Psi_h^{(r)}(q_0, \tilde{p}_0)$ where

$$\tilde{q}_1 = \pi_Q \circ \Psi_h^{(r+1)}(q_0, \tilde{p}_0) \qquad \text{and} \qquad (q_{c_i}, p_{c_i}) = \Psi_{c_i h}^{(r)}(q_0, \tilde{p}_0),$$

and where we use the continuous Legendre Transform to obtain $\dot{q}_{c_i}$.

Then, the discrete right Hamiltonian H_d^+ approximates $H_d^{+,E}$ with order of accuracy at least $\min(r+1, s)$. By Theorem 2.2 in [Schmitt and Leok, 2017], the associated discrete right Hamiltonian map has the same order of accuracy.

Proof. From Lemma A.2 we have that $q(c_i h) = q_{c_i} + \mathcal{O}(h^{r+1})$ and $p(c_i h) = p_{c_i} + \mathcal{O}(h^{r+1})$, and since $\frac{\partial H}{\partial p}$ is Lipschitz in both variables

$$\dot{q}(c_i h) - \dot{q}_{c_i} = \frac{\partial H}{\partial p}(q(c_i h), p(c_i h)) - \frac{\partial H}{\partial p}(q_{c_i}, p_{c_i}) = \mathcal{O}(h^{r+1}).$$

Since the quadrature formula is of order s accurate, equation (2.50) for $H_d^{+,E}(q_0, p_1; h)$ gives

$$H_d^{+,E}(q_0, p_1; h) = p_1^\intercal q_1 - h \sum_{i=1}^{m} b_i \left[p(c_i h)^\intercal \dot{q}(c_i h) - H\left(q_{c_i} + \mathcal{O}(h^{r+1}), p_{c_i} + \mathcal{O}(h^{r+1}) \right) \right] + \mathcal{O}(h^{s+1}).$$

Now, since $\tilde{q}_1 = \pi_Q \circ \Psi_h^{(r+1)}(q_0, \tilde{p}_0)$, it follows from Lemma A.2 that $\tilde{q}_1 = q_1 + \mathcal{O}(h^{r+2})$. Therefore, combining this with the fact that H is Lipschitz continuous in both variables yields

$$\begin{aligned} H_d^{+,E}(q_0, p_1; h) &= p_1^\intercal \tilde{q}_1 - h \sum_{i=1}^{m} b_i \left[p_{c_i}^\intercal \dot{q}_{c_i} - H\left(q_{c_i}, p_{c_i}\right) \right] + \mathcal{O}(h^{r+2}) + \mathcal{O}(h^{s+1}) \\ &= H_d^+(q_0, p_1; h) + \mathcal{O}(h^{\min(r+1,s)+1}). \end{aligned}$$

Therefore, H_d^+ approximates $H_d^{+,E}$ with order of accuracy at least $\min(r+1, s)$. □

Theorem 2.6 can be proven in a similar way for left HTVIs. Let $q(t)$ and $p(t)$ denote the solutions of the Hamilton's boundary-value problem

$$\dot{q}(t) = g(q(t), p(t), t), \qquad \dot{p}(t) = f(q(t), p(t), t), \qquad q(h) = q_1, \quad p(0) = p_0,$$

and let $q_0 = q(0)$ and $p_1 = p(h)$. Lemma A.1 is replaced by

Lemma A.3. *Given a $(r+1)$-order Taylor method $\Psi_h^{(r+1)}$ approximating the exact time-h flow map corresponding to Hamilton's equations, let $\tilde{q}_0$ solve the problem*

$$q_1 \;=\; \pi_Q \;\circ\; \Psi_h^{(r+1)}(\tilde{q}_0, p_0).$$

Then,

$$\tilde{q}_0 = q_0 + \mathcal{O}(h^{r+2}).$$

Proof. We proceed in the same way as in the proof of Lemma A.1. We first solve the equation $q_1 = \pi_Q \circ \Psi_h^{(r+1)}(\tilde{q}_0, p_0)$ for $\tilde{q}_0$, and then Taylor expand the exact solution $q(t)$ which belongs to $C^{r+2}([0,h])$. Now, $q(t)$ is Lipschitz continuous in its arguments for $k = 1, ..., r$, so we can let M be the largest of the corresponding r Lipschitz constants with respect to the first argument over the compact interval $[0, C]$. Then, as before, the triangle inequality can be used to get that $\left(1 - M\sum_{k=1}^{r+1} \frac{h^k}{k!}\right) \|\tilde{q}_0 - q_0\| = \mathcal{O}(h^{r+2})$, and by continuity, the term inside the parenthesis is bounded away from zero. □

In analogy to Lemma A.2, we now show that starting the r-order Taylor method with initial conditions $(\tilde{q}_0, p_0)$ rather than (q_0, p_0) will not affect the order of accuracy of the method.

Lemma A.4. *The r-order Taylor method $\Psi_h^{(r)}$ with initial conditions $(\tilde{q}_0, p_0)$ and where $\tilde{q}_0$ solves $q_1 = \pi_Q \circ \Psi_h^{(r)}(\tilde{q}_0, p_0)$ is accurate to at least $\mathcal{O}(h^{r+1})$ for the Hamiltonian boundary-value problem.*

Proof. Let $(\tilde{q}(t), \tilde{p}(t))$ denote the exact solution to Hamiltonian's equations with initial values $(\tilde{q}_0, p_0)$, and let $(q_d(t), p_d(t))$ denote the values generated by the r-order Taylor method with initial conditions $(\tilde{q}_0, p_0)$. The Hamiltonian initial-value problem is well-posed, so denoting the Lipschitz constant with respect to the first argument by M, we can

estimate

$$\|(q(t),p(t)) - (q_d(t),p_d(t))\| \leq \|(q(t),p(t)) - (\tilde{q}(t),\tilde{p}(t))\| + \|(\tilde{q}(t),\tilde{p}(t)) - (q_d(t),p_d(t))\|$$
$$\leq M\|q_0 - \tilde{q}_0\| + \mathcal{O}(h^{r+1}) \leq \mathcal{O}(h^{r+1}),$$

where we have used the triangle inequality, and the fact that the local truncation error of a r-order Taylor method is $\mathcal{O}(h^{r+1})$ to bound $\|(\tilde{q}(t),\tilde{p}(t)) - (q_d(t),p_d(t))\|$. □

We are now ready to prove Theorem 2.6 for left HTVIs.

Theorem A.2. *Consider a Hamiltonian H such that H and $\frac{\partial H}{\partial p}$ are Lipschitz continuous in both variables. Given a r-order accurate Taylor method $\Psi_h^{(r)}$ and a s-order accurate quadrature formula with weights and nodes (b_i, c_i), define the associated Taylor discrete left Hamiltonian*

$$H_d^-(q_1,p_0;h) = -p_0^\top \tilde{q}_0 - h\sum_{i=1}^{m} b_i\left[p_{c_i}^\top \dot{q}_{c_i} - H(q_{c_i},p_{c_i})\right],$$

where $\tilde{q}_0$ solve the problem $q_1 = \pi_Q \circ \Psi_h^{(r+1)}(\tilde{q}_0,p_0)$ where $(q_{c_i},p_{c_i}) = \Psi_{c_i h}^{(r)}(q_0,\tilde{p}_0)$, and where we use the continuous Legendre Transform to obtain $\dot{q}_{c_i}$.

Then, H_d^- approximates $H_d^{-,E}$ with order of accuracy at least $\min(r+1,s)$. By a result analogous to Theorem 2.2 in [Schmitt and Leok, 2017], the associated discrete left Hamiltonian map has the same order of accuracy.

Proof. From Lemma A.4 we have that $q(c_i h) = q_{c_i} + \mathcal{O}(h^{r+1})$, and $p(c_i h) = p_{c_i} + \mathcal{O}(h^{r+1})$, and since $\frac{\partial H}{\partial p}$ is Lipschitz in both variables

$$\dot{q}(c_i h) - \dot{q}_{c_i} \;=\; \frac{\partial H}{\partial p}(q(c_i h),p(c_i h)) - \frac{\partial H}{\partial p}(q_{c_i},p_{c_i}) \;=\; \mathcal{O}(h^{r+1}).$$

Since the quadrature formula is of order s accurate, equation (2.53) for $H_d^{-,E}(q_1,p_0;h)$ gives

$$H_d^{-,E}(q_1,p_0;h) = -p_0^\top q_0 - h\sum_{i=1}^{m} b_i\left[p(c_i h)^\top \dot{q}(c_i h) - H\left(q_{c_i} + \mathcal{O}(h^{r+1}), p_{c_i} + \mathcal{O}(h^{r+1})\right)\right] + \mathcal{O}(h^{s+1}).$$

Now, since $q_1 = \pi_Q \circ \Psi_h^{(r+1)}(\tilde{q}_0,p_0)$, it follows from Lemma A.3 that $\tilde{q}_0 = q_0 + \mathcal{O}(h^{r+2})$. Then, combining this with the fact that H is Lipschitz continuous in both variables yields

$$H_d^{-,E}(q_1,p_0;h) = H_d^-(q_1,p_0;h) + \mathcal{O}(h^{\min(r+1,s)+1}).$$

□

A.2 Constrained Variational Mechanics

A.2.1 Theorem 2.7: Constrained Euler–Lagrange equations

Theorem A.3. *Consider the constrained action functional* $\mathfrak{S} : C^2([0,T], \mathcal{Q} \times \Lambda) \to \mathbb{R}$ *given by*

$$\mathfrak{S}(q(\cdot), \lambda(\cdot)) = \int_0^T \left[L(q(t), \dot{q}(t)) - \langle \lambda(t), \mathcal{C}(q(t)) \rangle \right] dt. \tag{A.1}$$

The condition that $\mathfrak{S}(q(\cdot), \lambda(\cdot))$ *is stationary with respect to the boundary conditions* $\delta q(0) = 0$ *and* $\delta q(T) = 0$ *is equivalent to* $(q(\cdot), \lambda(\cdot))$ *satisfying the constrained Euler–Lagrange equations*

$$\frac{\partial L}{\partial q} - \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \langle \lambda, \nabla \mathcal{C}(q) \rangle, \qquad \mathcal{C}(q) = 0. \tag{A.2}$$

Proof. Computing the variation of $\mathfrak{S}$ yields

$$\begin{aligned} \delta\mathfrak{S} = \int_0^T \left[\frac{\partial L}{\partial q}(q(t), \dot{q}(t)) \delta q(t) + \frac{\partial L}{\partial \dot{q}}(q(t), \dot{q}(t)) \delta \dot{q}(t) \right] dt \\ - \int_0^T \left[\langle \lambda(t), \nabla \mathcal{C}(q(t)) \delta q(t) \rangle + \langle \delta\lambda(t), \mathcal{C}(q(t)) \rangle \right] dt. \end{aligned}$$

Using integration by parts and the boundary conditions $\delta q(0) = 0$ and $\delta q(T) = 0$, we get

$$\begin{aligned} \delta\mathfrak{S} = \int_0^T \left[\frac{\partial L}{\partial q}(q(t), \dot{q}(t)) - \frac{d}{dt}\frac{\partial L}{\partial \dot{q}}(q(t), \dot{q}(t)) - \langle \lambda(t), \nabla \mathcal{C}(q(t)) \rangle \right] \delta q(t) dt \\ - \int_0^T \langle \delta\lambda(t), \mathcal{C}(q(t)) \rangle dt. \end{aligned}$$

Now, if $\delta\mathfrak{S} = 0$, then the fundamental theorem of the calculus of variations [Arnol′d, 1989] yields the constrained Euler–Lagrange equations (A.2).

Conversely, if (q, λ) satisfies the constrained Euler–Lagrange equations (A.2), then the integrand vanishes and $\delta\mathfrak{S} = 0$. □

A.2.2 Theorem 2.10: Type II Constrained H Equations

Theorem A.4. *Consider the constrained action functional* $\mathfrak{S} : C^2([0,T], T^*\mathcal{Q} \times \Lambda) \to \mathbb{R}$ *given by*

$$\mathfrak{S}(q(\cdot), p(\cdot), \lambda(\cdot)) = p(T)q(T) - \int_0^T \left[p(t)\dot{q}(t) - H(q(t), p(t)) - \langle \lambda(t), \mathcal{C}(q(t))\rangle\right] dt. \quad \text{(A.3)}$$

The condition that $\mathfrak{S}(q(\cdot), p(\cdot), \lambda(\cdot))$ *is stationary with respect to the boundary conditions* $\delta q(0) = 0$ *and* $\delta p(T) = 0$ *is equivalent to* $(q(\cdot), p(\cdot), \lambda(\cdot))$ *satisfying Hamilton's canonical constrained equations*

$$\dot{q} = \frac{\partial H}{\partial p}(q,p), \qquad \dot{p} = -\frac{\partial H}{\partial q}(q,p) - \langle \lambda, \nabla \mathcal{C}(q)\rangle, \qquad \mathcal{C}(q) = 0. \quad \text{(A.4)}$$

Proof. Computing the variation of $\mathfrak{S}$ yields

$$\begin{aligned}\delta\mathfrak{S} = {} & q(T)\delta p(T) + p(T)\delta q(T) + \int_0^T \left[\langle \lambda(t), \nabla\mathcal{C}(q(t))\delta q(t)\rangle + \langle \delta\lambda(t), \mathcal{C}(q(t))\rangle\right] dt \\ & - \int_0^T \left[\dot{q}(t)\delta p(t) + p(t)\delta\dot{q}(t) - \frac{\partial H}{\partial q}(q(t),p(t))\delta q(t) - \frac{\partial H}{\partial p}(q(t),p(t))\delta p(t)\right] dt.\end{aligned}$$

Using integration by parts and the boundary conditions $\delta q(0) = 0$ and $\delta p(T) = 0$, we get

$$\begin{aligned}\delta\mathfrak{S} = {} & q(T)\delta p(T) + p(T)\delta q(T) - p(T)\delta q(T) + p(0)\delta q(0) + \int_0^T \langle \delta\lambda(t), \mathcal{C}(q(t))\rangle dt \\ & + \int_0^T \left[\dot{p}(t) + \frac{\partial H}{\partial q}(q(t),p(t)) + \langle \lambda(t), \nabla\mathcal{C}(q(t))\rangle\right]\delta q(t) dt \\ & + \int_0^T \left[\frac{\partial H}{\partial p}(q(t),p(t)) - \dot{q}(t)\right]\delta p(t) dt \\ = {} & \int_0^T \left[\dot{p}(t) + \frac{\partial H}{\partial q}(q(t),p(t)) + \langle \lambda(t), \nabla\mathcal{C}(q(t))\rangle\right]\delta q(t) dt \\ & + \int_0^T \left[\frac{\partial H}{\partial p}(q(t),p(t)) - \dot{q}(t)\right]\delta p(t) dt \\ & + \int_0^T \langle \delta\lambda(t), \mathcal{C}(q(t))\rangle dt.\end{aligned}$$

Now, if $\delta\mathfrak{S} = 0$, then the fundamental theorem of the calculus of variations [Arnol′d, 1989] yields Hamilton's constrained equations (A.4).

Conversely, if (q, p, λ) satisfies Hamilton's constrained equations (A.4), then the integrand vanishes and $\delta\mathfrak{S} = 0$. □

A.2.3 Theorem 2.11: Type III Constrained H Equations

Theorem A.5. *Consider the constrained action functional* $\mathfrak{S} : C^2([0,T], T^*\mathcal{Q} \times \Lambda) \to \mathbb{R}$ *given by*

$$\mathfrak{S}(q(\cdot), p(\cdot), \lambda(\cdot)) = -p(0)q(0) - \int_0^T \left[p(t)\dot{q}(t) - H(q(t), p(t)) - \langle \lambda(t), \mathcal{C}(q(t))\rangle\right] dt. \quad \text{(A.5)}$$

The condition that $\mathfrak{S}(q(\cdot), p(\cdot), \lambda(\cdot))$ *is stationary with respect to the boundary conditions* $\delta q(T) = 0$ *and* $\delta p(0) = 0$ *is equivalent to* $(q(\cdot), p(\cdot), \lambda(\cdot))$ *satisfying Hamilton's canonical constrained equations*

$$\dot{q} = \frac{\partial H}{\partial p}(q,p), \qquad \dot{p} = -\frac{\partial H}{\partial q}(q,p) - \langle \lambda, \nabla \mathcal{C}(q)\rangle, \qquad \mathcal{C}(q) = 0. \quad \text{(A.6)}$$

Proof. The proof is almost identical to that of Theorem 2.10. We compute the variation of $\mathfrak{S}$ as before and get

$$\begin{aligned}\delta\mathfrak{S} = -q(0)\delta p(0) - p(0)\delta q(0) + \int_0^T \left[\langle \lambda(t), \nabla \mathcal{C}(q(t))\delta q(t)\rangle + \langle \delta\lambda(t), \mathcal{C}(q(t))\rangle\right] dt \\ - \int_0^T \left[\dot{q}(t)\delta p(t) + p(t)\delta\dot{q}(t) - \frac{\partial H}{\partial q}(q(t),p(t))\delta q(t) - \frac{\partial H}{\partial p}(q(t),p(t))\delta p(t)\right] dt.\end{aligned}$$

Integration by parts and the boundary conditions $\delta q(T) = 0$ and $\delta p(0) = 0$ yield

$$\begin{aligned}\delta\mathfrak{S} = \int_0^T \left[\dot{p}(t) + \frac{\partial H}{\partial q}(q(t),p(t)) + \langle \lambda(t), \nabla \mathcal{C}(q(t))\rangle\right] \delta q(t) dt \\ + \int_0^T \left[\frac{\partial H}{\partial p}(q(t),p(t)) - \dot{q}(t)\right] \delta p(t) dt \\ + \int_0^T \langle \delta\lambda(t), \mathcal{C}(q(t))\rangle dt.\end{aligned}$$

Then, if $\delta\mathfrak{S} = 0$, then the fundamental theorem of the calculus of variations [Arnol′d, 1989] yields Hamilton's constrained equations (A.6).

Conversely, if (q, p, λ) satisfies Hamilton's constrained equations (A.6), then the integrand vanishes and $\delta\mathfrak{S} = 0$. □

A.2.4 Theorem 2.8: Type I Generating Function

Theorem A.6. *The exact time-T flow map of Hamilton's equations $(q_0, p_0) \mapsto (q_T, p_T)$ is implicitly given by the following relations:*

$$D_1\mathcal{S}(q_0, q_T) = -\frac{\partial L}{\partial \dot{q}}(q_0, \dot{q}(0)), \qquad D_2\mathcal{S}(q_0, q_T) = \frac{\partial L}{\partial \dot{q}}(q_T, \dot{q}(T)). \tag{A.7}$$

Thus, $\mathcal{S}(q_0, q_T)$ is a Type I generating function that generates the exact flow of the constrained Euler–Lagrange equations (2.70).

Proof. Using integration by parts and simplifying gives

$$\begin{aligned}
\frac{\partial \mathcal{S}}{\partial q_0}(q_0, q_T) &= \int_0^T \left[\frac{\partial q(t)}{\partial q_0} \frac{\partial L}{\partial q}(q(t), \dot{q}(t)) + \frac{\partial \dot{q}(t)}{\partial q_0} \frac{\partial L}{\partial \dot{q}}(q(t), \dot{q}(t)) \right] dt \\
&\qquad - \int_0^T \left[\langle \lambda(t), \frac{\partial q(t)}{\partial q_0} \nabla C(q(t)) \rangle + \langle \frac{\partial \lambda(t)}{\partial q_0}, C(q(t)) \rangle \right] dt \\
&= \int_0^T \frac{\partial q(t)}{\partial q_0} \left(\frac{\partial L}{\partial q}(q(t), \dot{q}(t)) - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}}(q(t), \dot{q}(t)) - \langle \lambda(t), \nabla C(q(t)) \rangle \right) dt \\
&\qquad - \int_0^T \langle \frac{\partial \lambda(t)}{\partial q_0}, C(q(t)) \rangle dt - \frac{\partial L}{\partial \dot{q}}(q(0), \dot{q}(0)),
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{S}}{\partial q_T}(q_0, q_T) &= \int_0^T \left[\frac{\partial q(t)}{\partial q_T} \frac{\partial L}{\partial q}(q(t), \dot{q}(t)) + \frac{\partial \dot{q}(t)}{\partial q_T} \frac{\partial L}{\partial \dot{q}}(q(t), \dot{q}(t)) \right] dt \\
&\qquad - \int_0^T \left[\langle \lambda(t), \frac{\partial q(t)}{\partial q_T} \nabla C(q(t)) \rangle + \langle \frac{\partial \lambda(t)}{\partial q_T}, C(q(t)) \rangle \right] dt \\
&= \int_0^T \frac{\partial q(t)}{\partial q_T} \left(\frac{\partial L}{\partial q}(q(t), \dot{q}(t)) - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}}(q(t), \dot{q}(t)) - \langle \lambda(t), \nabla C(q(t)) \rangle \right) dt \\
&\qquad - \int_0^T \langle \frac{\partial \lambda(t)}{\partial q_T}, C(q(t)) \rangle dt + \frac{\partial L}{\partial \dot{q}}(q(T), \dot{q}(T)).
\end{aligned}$$

By Theorem 2.7, the extremum of the action functional $\mathfrak{S}$ is achieved when (q, λ) satisfies the constrained Euler–Lagrange equations (2.70), so we get the desired equations $D_1\mathcal{S}(q_0, q_T) = -\frac{\partial L}{\partial \dot{q}}(q_0, \dot{q}(0))$ and $D_2\mathcal{S}(q_0, q_T) = \frac{\partial L}{\partial \dot{q}}(q_T, \dot{q}(T))$. □

A.2.5 Theorem 2.12: Type II Generating Function

Theorem A.7. *The exact time-T flow map of Hamilton's equations $(q_0, p_0) \mapsto (q_T, p_T)$ is implicitly given by the following relations:*

$$q_T = D_2\mathcal{S}(q_0, p_T), \qquad p_0 = D_1\mathcal{S}(q_0, p_T). \tag{A.8}$$

In particular, $\mathcal{S}(q_0, p_T)$ is a Type II generating function that generates the exact flow of Hamilton's constrained equations (2.81)*.*

Proof. Using integration by parts and simplifying gives

$$\begin{aligned}
\frac{\partial \mathcal{S}}{\partial q_0}(q_0, p_T) &= \frac{\partial q_T}{\partial q_0} p_T + \int_0^T \left[\langle \lambda(t), \frac{\partial q(t)}{\partial q_0} \nabla \mathcal{C}(q(t)) \rangle + \langle \frac{\partial \lambda(t)}{\partial q_0}, \mathcal{C}(q(t)) \rangle \right] dt \\
&\quad - \int_0^T \left[\frac{\partial p(t)}{\partial q_0} \dot{q}(t) + \frac{\partial \dot{q}(t)}{\partial q_0} p(t) - \frac{\partial q(t)}{\partial q_0} \frac{\partial H}{\partial q}(q(t), p(t)) - \frac{\partial p(t)}{\partial q_0} \frac{\partial H}{\partial p}(q(t), p(t)) \right] dt \\
&= p_0 + \int_0^T \frac{\partial q(t)}{\partial q_0} \left(\dot{p}(t) + \frac{\partial H}{\partial q}(q(t), p(t)) + \langle \lambda(t), \nabla \mathcal{C}(q(t)) \rangle \right) dt \\
&\qquad - \int_0^T \frac{\partial p(t)}{\partial q_0} \left(\dot{q}(t) - \frac{\partial H}{\partial p}(q(t), p(t)) \right) dt + \int_0^T \langle \frac{\partial \lambda(t)}{\partial q_0}, \mathcal{C}(q(t)) \rangle dt,
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{S}}{\partial p_T}(q_0, p_T) &= q_T + \frac{\partial q_T}{\partial p_T} p_T + \int_0^T \left[\langle \lambda(t), \frac{\partial q(t)}{\partial p_T} \nabla \mathcal{C}(q(t)) \rangle + \langle \frac{\partial \lambda(t)}{\partial p_T}, \mathcal{C}(q(t)) \rangle \right] dt \\
&\quad - \int_0^T \left[\frac{\partial p(t)}{\partial p_T} \dot{q}(t) + \frac{\partial \dot{q}(t)}{\partial p_T} p(t) - \frac{\partial q(t)}{\partial p_T} \frac{\partial H}{\partial q}(q(t), p(t)) - \frac{\partial p(t)}{\partial p_T} \frac{\partial H}{\partial p}(q(t), p(t)) \right] dt \\
&= q_T + \int_0^T \frac{\partial q(t)}{\partial p_T} \left(\dot{p}(t) + \frac{\partial H}{\partial q}(q(t), p(t)) + \langle \lambda(t), \nabla \mathcal{C}(q(t)) \rangle \right) dt \\
&\qquad - \int_0^T \frac{\partial p(t)}{\partial p_T} \left(\dot{q}(t) - \frac{\partial H}{\partial p}(q(t), p(t)) \right) dt + \int_0^T \langle \frac{\partial \lambda(t)}{\partial p_T}, \mathcal{C}(q(t)) \rangle dt.
\end{aligned}$$

By Theorem 2.10, the extremum of the action functional $\mathfrak{S}$ is achieved when the curve (q, p, λ) satisfies Hamilton's constrained equations (2.81), so the integrands vanish, and thus $p_0 = \frac{\partial \mathcal{S}}{\partial q_0}(q_0, p_T) = D_1\mathcal{S}(q_0, p_T)$ and $q_T = \frac{\partial \mathcal{S}}{\partial p_T}(q_0, p_T) = D_2\mathcal{S}(q_0, p_T)$. □

A.2.6 Theorem 2.13: Type III Generating Function

Theorem A.8. *The exact time-T flow map of Hamilton's equations $(q_0, p_0) \mapsto (q_T, p_T)$ is implicitly given by the following relations:*

$$q_0 = -D_2\mathcal{S}(q_T, p_0), \qquad p_T = -D_1\mathcal{S}(q_T, p_0). \tag{A.9}$$

In particular, $\mathcal{S}(q_T, p_0)$ is a Type III generating function that generates the exact flow of Hamilton's constrained equations (2.83)*.*

Proof. Integrating by parts and simplifying yields

$$\begin{aligned}
\frac{\partial \mathcal{S}}{\partial q_T}(q_T, p_0) &= -\frac{\partial q_0}{\partial q_T}p_0 + \int_0^T \left[\langle \lambda(t), \frac{\partial q(t)}{\partial q_T}\nabla \mathcal{C}(q(t))\rangle + \langle \frac{\partial \lambda(t)}{\partial q_T}, \mathcal{C}(q(t))\rangle\right] dt \\
&\quad - \int_0^T \left[\frac{\partial p(t)}{\partial q_T}\dot{q}(t) + \frac{\partial \dot{q}(t)}{\partial q_T}p(t) - \frac{\partial q(t)}{\partial q_T}\frac{\partial H}{\partial q}(q(t), p(t)) - \frac{\partial p(t)}{\partial q_T}\frac{\partial H}{\partial p}(q(t), p(t))\right] dt \\
&= -p_T + \int_0^T \frac{\partial q(t)}{\partial q_T}\left(\dot{p}(t) + \frac{\partial H}{\partial q}(q(t), p(t)) + \langle \lambda(t), \nabla \mathcal{C}(q(t))\rangle\right) dt \\
&\qquad - \int_0^T \frac{\partial p(t)}{\partial q_T}\left(\dot{q}(t) - \frac{\partial H}{\partial p}(q(t), p(t))\right) dt + \int_0^T \langle \frac{\partial \lambda(t)}{\partial q_T}, \mathcal{C}(q(t))\rangle dt,
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{S}}{\partial p_0}(q_T, p_0) &= -q_0 - \frac{\partial q_0}{\partial p_0}p_0 + \int_0^T \left[\langle \lambda(t), \frac{\partial q(t)}{\partial p_0}\nabla \mathcal{C}(q(t))\rangle + \langle \frac{\partial \lambda(t)}{\partial p_0}, \mathcal{C}(q(t))\rangle\right] dt \\
&\quad - \int_0^T \left[\frac{\partial p(t)}{\partial p_0}\dot{q}(t) + \frac{\partial \dot{q}(t)}{\partial p_0}p(t) - \frac{\partial q(t)}{\partial p_0}\frac{\partial H}{\partial q}(q(t), p(t)) - \frac{\partial p(t)}{\partial p_0}\frac{\partial H}{\partial p}(q(t), p(t))\right] dt \\
&= -q_0 + \int_0^T \frac{\partial q(t)}{\partial p_0}\left(\dot{p}(t) + \frac{\partial H}{\partial q}(q(t), p(t)) + \langle \lambda(t), \nabla \mathcal{C}(q(t))\rangle\right) dt \\
&\qquad - \int_0^T \frac{\partial p(t)}{\partial p_0}\left(\dot{q}(t) - \frac{\partial H}{\partial p}(q(t), p(t))\right) dt + \int_0^T \langle \frac{\partial \lambda(t)}{\partial p_0}, \mathcal{C}(q(t))\rangle dt.
\end{aligned}$$

By Theorem 2.11, the extremum of the action $\mathfrak{S}$ is achieved when the curve (q, p, λ) satisfies Hamilton's constrained equations (2.83), so the integrands vanish, and thus $p_T = -\frac{\partial \mathcal{S}}{\partial q_T}(q_T, p_0) = -D_1\mathcal{S}(q_T, p_0)$ and $q_0 = -\frac{\partial \mathcal{S}}{\partial p_0}(q_T, p_0) = -D_2\mathcal{S}(q_T, p_0)$. □

A.2.7 Theorem 2.9: Discrete Constrained EL Equations

Theorem A.9. *The Type I discrete Hamilton's variational principles*

$$\delta \mathfrak{S}_d^{\pm} \left(\{(q_k, \lambda_k)\}_{k=0}^{N} \right) = 0 \tag{A.10}$$

are equivalent to the discrete constrained Euler–Lagrange equations

$$D_1 L_d(q_k, q_{k+1}) + D_2 L_d(q_{k-1}, q_k) = \langle \lambda_k, \nabla \mathcal{C}(q_k) \rangle, \qquad \mathcal{C}(q_k) = 0, \tag{A.11}$$

where $L_d(q_k, q_{k+1})$ is defined via equation (2.75)*.*

Proof. Using the fact that $\delta q_0 = 0$ and $\delta q_N = 0$, we have

$$\begin{aligned}
\delta \mathfrak{S}_d^{-} &= \delta \left(\sum_{k=0}^{N-1} \left[L_d(q_k, q_{k+1}) - \langle \lambda_k, \mathcal{C}(q_k) \rangle \right] \right) \\
&= \sum_{k=0}^{N-1} \left[D_1 L_d(q_k, q_{k+1}) \delta q_k + D_2 L_d(q_k, q_{k+1}) \delta q_{k+1} \right] \\
&\qquad\qquad - \sum_{k=0}^{N-1} \left(\langle \lambda_k, \nabla \mathcal{C}(q_k) \delta q_k \rangle + \langle \delta \lambda_k, \mathcal{C}(q_k) \rangle \right) \\
&= \sum_{k=1}^{N-1} \left[D_1 L_d(q_k, q_{k+1}) + D_2 L_d(q_{k-1}, q_k) - \langle \lambda_k, \nabla \mathcal{C}(q_k) \rangle \right] \delta q_k - \sum_{k=0}^{N-1} \langle \delta \lambda_k, \mathcal{C}(q_k) \rangle, \\
\delta \mathfrak{S}_d^{+} &= \delta \left(\sum_{k=0}^{N-1} \left[L_d(q_k, q_{k+1}) - \langle \lambda_{k+1}, \mathcal{C}(q_{k+1}) \rangle \right] \right) \\
&= \sum_{k=0}^{N-1} \left[D_1 L_d(q_k, q_{k+1}) \delta q_k + D_2 L_d(q_k, q_{k+1}) \delta q_{k+1} \right] \\
&\qquad\qquad - \sum_{k=0}^{N-1} \left(\langle \lambda_{k+1}, \nabla \mathcal{C}(q_{k+1}) \delta q_{k+1} \rangle + \langle \delta \lambda_{k+1}, \mathcal{C}(q_{k+1}) \rangle \right) \\
&= \sum_{k=1}^{N-1} \left[D_1 L_d(q_k, q_{k+1}) + D_2 L_d(q_{k-1}, q_k) - \langle \lambda_k, \nabla \mathcal{C}(q_k) \rangle \right] \delta q_k - \sum_{k=0}^{N-1} \langle \delta \lambda_{k+1}, \mathcal{C}(q_{k+1}) \rangle.
\end{aligned}$$

If the discrete constrained Euler–Lagrange equations (A.11) are satisfied, then each term vanishes and $\delta \mathfrak{S}_d^{\pm} = 0$.

Conversely, if $\delta \mathfrak{S}_d^{\pm} = 0$, then a discrete fundamental theorem of the calculus of variations yields the discrete constrained Euler–Lagrange equations (A.11). □

A.2.8 Theorem 2.14: Discrete Constrained Right H Equations

Theorem A.10. *The Type II discrete Hamilton's phase space variational principle*

$$\delta \mathfrak{S}_d^+ \left(\{(q_k, p_k, \lambda_k)\}_{k=0}^N \right) = 0 \tag{A.12}$$

is equivalent to the discrete constrained right Hamilton's equations

$$q_{k+1} = D_2 H_d^+(q_k, p_{k+1}), \qquad p_k = D_1 H_d^+(q_k, p_{k+1}) + \langle \lambda_k, \nabla \mathcal{C}(q_k) \rangle, \qquad \mathcal{C}(q_k) = 0, \tag{A.13}$$

where $H_d^+(q_k, p_{k+1})$ is defined via equation (2.91).

Proof. Using the fact that $\delta q_0 = 0$ and $\delta p_N = 0$ since (q_0, p_N) is fixed, we obtain the following expression for the variations of $\mathfrak{S}_d^+$:

$$\begin{aligned}
\delta \mathfrak{S}_d^+ &= \delta \left(p_N q_N - \sum_{k=0}^{N-1} \left[p_{k+1} q_{k+1} - H_d^+(q_k, p_{k+1}) - \langle \lambda_k, \mathcal{C}(q_k) \rangle \right] \right) \\
&= \delta \left(- \sum_{k=0}^{N-2} p_{k+1} q_{k+1} + \sum_{k=0}^{N-1} \left[H_d^+(q_k, p_{k+1}) + \langle \lambda_k, \mathcal{C}(q_k) \rangle \right] \right) \\
&= - \sum_{k=0}^{N-2} \left(q_{k+1} \delta p_{k+1} + p_{k+1} \delta q_{k+1} \right) + \sum_{k=0}^{N-1} \left(D_1 H_d^+(q_k, p_{k+1}) \delta q_k + D_2 H_d^+(q_k, p_{k+1}) \delta p_{k+1} \right) \\
&\qquad + \sum_{k=0}^{N-1} \left(\langle \lambda_k, \nabla \mathcal{C}(q_k) \delta q_k \rangle + \langle \delta \lambda_k, \mathcal{C}(q_k) \rangle \right) \\
&= - \sum_{k=1}^{N-1} \left(q_k \delta p_k + p_k \delta q_k \right) + \sum_{k=1}^{N-1} D_1 H_d^+(q_k, p_{k+1}) \delta q_k + \sum_{k=0}^{N-2} D_2 H_d^+(q_k, p_{k+1}) \delta p_{k+1} \\
&\qquad + \sum_{k=0}^{N-1} \left(\langle \lambda_k, \nabla \mathcal{C}(q_k) \delta q_k \rangle + \langle \delta \lambda_k, \mathcal{C}(q_k) \rangle \right) \\
&= - \sum_{k=1}^{N-1} \left(q_k \delta p_k + p_k \delta q_k \right) + \sum_{k=1}^{N-1} D_1 H_d^+(q_k, p_{k+1}) \delta q_k + \sum_{k=1}^{N-1} D_2 H_d^+(q_{k-1}, p_k) \delta p_k \\
&\qquad + \sum_{k=0}^{N-1} \left(\langle \lambda_k, \nabla \mathcal{C}(q_k) \delta q_k \rangle + \langle \delta \lambda_k, \mathcal{C}(q_k) \rangle \right) \\
&= \sum_{k=1}^{N-1} \left[-q_k + D_2 H_d^+(q_{k-1}, p_k) \right] \delta p_k + \sum_{k=0}^{N-1} \langle \delta \lambda_k, \mathcal{C}(q_k) \rangle \\
&\qquad + \sum_{k=1}^{N-1} \left[-p_k + D_1 H_d^+(q_k, p_{k+1}) + \langle \lambda_k, \nabla \mathcal{C}(q_k) \rangle \right] \delta q_k .
\end{aligned}$$

If the discrete constrained right Hamilton's equations (A.13) are satisfied, then each term vanishes so $\delta \mathfrak{S}_d^+ = 0$. Conversely, $\delta \mathfrak{S}_d^+ = 0$ yields the discrete constrained right Hamilton's equations (A.13), by a discrete fundamental theorem of the calculus of variations $\square$

A.2.9 Theorem 2.15: Discrete Constrained Left H Equations

Theorem A.11. *The Type III discrete Hamilton's phase space variational principle*

$$\delta \mathfrak{S}_d^- \left(\{(q_k, p_k, \lambda_k)\}_{k=0}^N \right) = 0 \tag{A.14}$$

is equivalent to the discrete constrained left Hamilton's equations

$$q_k = -D_2 H_d^-(q_{k+1}, p_k), \quad p_{k+1} = -D_1 H_d^-(q_{k+1}, p_k) - \langle \lambda_{k+1}, \nabla \mathcal{C}(q_{k+1}) \rangle, \quad \mathcal{C}(q_k) = 0, \tag{A.15}$$

where $H_d^-(q_{k+1}, p_k)$ *is defined via equation* (2.92).

Proof. Using the fact that $\delta q_N = 0$ and $\delta p_0 = 0$ since (q_N, p_0) is fixed, we obtain the following expression for the variations of $\mathfrak{S}_d^-$:

$$\begin{aligned}
\delta \mathfrak{S}_d^- &= \delta \left(-p_0 q_0 - \sum_{k=0}^{N-1} \left[-p_k q_k - H_d^-(q_{k+1}, p_k) - \langle \lambda_{k+1}, \mathcal{C}(q_{k+1}) \rangle \right] \right) \\
&= \delta \left(\sum_{k=1}^{N-1} p_k q_k + \sum_{k=0}^{N-1} \left[H_d^-(q_{k+1}, p_k) + \langle \lambda_{k+1}, \mathcal{C}(q_{k+1}) \rangle \right] \right) \\
&= \sum_{k=1}^{N-1} (q_k \delta p_k + p_k \delta q_k) + \sum_{k=0}^{N-1} \left(D_1 H_d^-(q_{k+1}, p_k) \delta q_{k+1} + D_2 H_d^-(q_{k+1}, p_k) \delta p_k \right) \\
&\qquad + \sum_{k=0}^{N-1} \left(\langle \lambda_{k+1}, \nabla \mathcal{C}(q_{k+1}) \delta q_{k+1} \rangle + \langle \delta \lambda_{k+1}, \mathcal{C}(q_{k+1}) \rangle \right) \\
&= \sum_{k=0}^{N} (q_k \delta p_k + p_k \delta q_k) + \sum_{k=0}^{N-2} D_1 H_d^-(q_{k+1}, p_k) \delta q_{k+1} + \sum_{k=1}^{N-1} D_2 H_d^-(q_{k+1}, p_k) \delta p_k \\
&\qquad + \sum_{k=0}^{N-1} \left(\langle \lambda_{k+1}, \nabla \mathcal{C}(q_{k+1}) \delta q_{k+1} \rangle + \langle \delta \lambda_{k+1}, \mathcal{C}(q_{k+1}) \rangle \right) \\
&= \sum_{k=1}^{N-1} (q_k \delta p_k + p_k \delta q_k) + \sum_{k=1}^{N-1} D_1 H_d^-(q_k, p_{k-1}) \delta q_k + \sum_{k=1}^{N-1} D_2 H_d^-(q_{k+1}, p_k) \delta p_k \\
&\qquad + \sum_{k=1}^{N} \langle \lambda_k, \nabla \mathcal{C}(q_k) \delta q_k \rangle + \sum_{k=0}^{N-1} \langle \delta \lambda_{k+1}, \mathcal{C}(q_{k+1}) \rangle \\
&= \sum_{k=1}^{N-1} \left[q_k + D_2 H_d^-(q_{k+1}, p_k) \right] \delta p_k + \sum_{k=0}^{N-1} \langle \delta \lambda_{k+1}, \mathcal{C}(q_{k+1}) \rangle \\
&\qquad + \sum_{k=1}^{N-1} \left[p_k + D_1 H_d^-(q_k, p_{k-1}) + \langle \lambda_k, \nabla \mathcal{C}(q_k) \rangle \right] \delta q_k.
\end{aligned}$$

If the discrete constrained left Hamilton's equations (A.15) are satisfied, then each term vanishes and $\delta \mathfrak{S}_d^- = 0$. Conversely, if $\delta \mathfrak{S}_d^- = 0$, then a discrete fundamental theorem of the calculus of variations yields the discrete constrained left Hamilton's equations (A.15). □

A.3 Lagrangian Variational Integrators with Variable Time-Steps

A.3.1 Theorem 2.21: Discrete Extended Euler–Lagrange Equations (I)

Theorem A.12. *The Type I discrete Hamilton's variational principle,*

$$\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=0,$$

where

$$\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=\sum_{k=0}^{N-1}\left[L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k},$$

is equivalent to the discrete extended Euler–Lagrange equations,

$$\mathfrak{q}_{k+1}=\mathfrak{q}_k+(\tau_{k+1}-\tau_k)g(\mathfrak{q}_k),$$

$$\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_1L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})+\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}D_3L_d(q_{k-1},\mathfrak{q}_{k-1},q_k,\mathfrak{q}_k)=0,$$

$$\begin{aligned}&\left[D_2L_{d_k}+\lambda_k\frac{1}{\tau_{k+1}-\tau_k}+\lambda_k\nabla g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}-\frac{1}{\tau_{k+1}-\tau_k}\left[L_{d_k}-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\\&\quad+\left[D_4L_{d_{k-1}}-\lambda_{k-1}\frac{1}{\tau_k-\tau_{k-1}}\right]\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\frac{1}{\tau_k-\tau_{k-1}}\left[L_{d_{k-1}}-\lambda_{k-1}\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\lambda_{k-1}g(\mathfrak{q}_{k-1})\right]=0,\end{aligned}$$

where L_{d_k} *denotes* $L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})$.

Proof. We use the notation $L_{d_k}=L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})$, and we will use the fact that $\delta q_0=\delta q_N=\delta\mathfrak{q}_0=\delta\mathfrak{q}_N=0$ throughout the proof. We have

$$\delta\bar{\mathfrak{S}}_d=\delta\left(\sum_{k=0}^{N-1}\left[L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\right)$$

so

$$\begin{aligned}
\delta\bar{\mathfrak{S}}_d = \sum_{k=1}^{N-1} &\left[D_2 L_{d_k} + \lambda_k \frac{1}{\tau_{k+1}-\tau_k} + \lambda_k \nabla g(\mathfrak{q}_k) \right] \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} \delta\mathfrak{q}_k \\
&- \sum_{k=1}^{N-1} \frac{1}{\tau_{k+1}-\tau_k} \left[L_{d_k} - \lambda_k \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} + \lambda_k g(\mathfrak{q}_k) \right] \delta\mathfrak{q}_k \\
&+ \sum_{k=0}^{N-2} \left[D_4 L_{d_k} - \lambda_k \frac{1}{\tau_{k+1}-\tau_k} \right] \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} \delta\mathfrak{q}_{k+1} \\
&+ \sum_{k=0}^{N-2} \frac{1}{\tau_{k+1}-\tau_k} \left[L_{d_k} - \lambda_k \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} + \lambda_k g(\mathfrak{q}_k) \right] \delta\mathfrak{q}_{k+1} \\
&+ \sum_{k=1}^{N-1} \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} D_1 L_{d_k} \delta q_k + \sum_{k=0}^{N-2} \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} D_3 L_{d_k} \delta q_{k+1} \\
&+ \sum_{k=0}^{N-1} \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} \left(g(\mathfrak{q}_k) - \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} \right) \delta\lambda_k.
\end{aligned}$$

Thus,

$$\begin{aligned}
\delta\bar{\mathfrak{S}}_d = \sum_{k=1}^{N-1} &\left[D_2 L_{d_k} + \lambda_k \frac{1}{\tau_{k+1}-\tau_k} + \lambda_k \nabla g(\mathfrak{q}_k) \right] \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} \delta\mathfrak{q}_k \\
&- \sum_{k=1}^{N-1} \frac{1}{\tau_{k+1}-\tau_k} \left[L_{d_k} - \lambda_k \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} + \lambda_k g(\mathfrak{q}_k) \right] \delta\mathfrak{q}_k \\
&+ \sum_{k=1}^{N-1} \left[D_4 L_{d_{k-1}} - \lambda_{k-1} \frac{1}{\tau_k-\tau_{k-1}} \right] \frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}} \delta\mathfrak{q}_k \\
&+ \sum_{k=1}^{N-1} \frac{1}{\tau_k-\tau_{k-1}} \left[L_{d_{k-1}} - \lambda_{k-1} \frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}} + \lambda_{k-1} g(\mathfrak{q}_{k-1}) \right] \delta\mathfrak{q}_k \\
&+ \sum_{k=1}^{N-1} \left[\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} D_1 L_{d_k} + \frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}} D_3 L_{d_{k-1}} \right] \delta q_k \\
&+ \sum_{k=0}^{N-1} \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} \left(g(\mathfrak{q}_k) - \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} \right) \delta\lambda_k.
\end{aligned}$$

As a consequence, if

$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + (\tau_{k+1}-\tau_k) g(\mathfrak{q}_k),$$

$$\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} D_1 L_d(q_k, \mathfrak{q}_k, q_{k+1}, \mathfrak{q}_{k+1}) + \frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}} D_3 L_d(q_{k-1}, \mathfrak{q}_{k-1}, q_k, \mathfrak{q}_k) = 0,$$

$$\begin{aligned}
&\left[D_2 L_{d_k} + \lambda_k \frac{1}{\tau_{k+1}-\tau_k} + \lambda_k \nabla g(\mathfrak{q}_k) \right] \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} - \frac{1}{\tau_{k+1}-\tau_k} \left[L_{d_k} - \lambda_k \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k} + \lambda_k g(\mathfrak{q}_k) \right] \\
&\quad + \left[D_4 L_{d_{k-1}} - \lambda_{k-1} \frac{1}{\tau_k-\tau_{k-1}} \right] \frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}} + \frac{1}{\tau_k-\tau_{k-1}} \left[L_{d_{k-1}} - \lambda_{k-1} \frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}} + \lambda_{k-1} g(\mathfrak{q}_{k-1}) \right] = 0,
\end{aligned}$$

then $\delta\bar{\mathfrak{S}}_d\left(\{(q_k, \mathfrak{q}_k, \lambda_k)\}_{k=0}^N\right) = 0$. Conversely, if $\delta\bar{\mathfrak{S}}_d\left(\{(q_k, \mathfrak{q}_k, \lambda_k)\}_{k=0}^N\right) = 0$, then a discrete fundamental theorem of the calculus of variations yields the above equations. □

A.3.2 Theorem 2.23: Discrete Extended Euler–Lagrange Equations (II)

Theorem A.13. *The Type I discrete Hamilton's variational principle,*

$$\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=0,$$

where

$$\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=\sum_{k=0}^{N-1}\left\{\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\left[L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})-\lambda_k\right]+\lambda_k g(\mathfrak{q}_k)\right\},$$

is equivalent to the discrete extended Euler–Lagrange equations,

$$\mathfrak{q}_{k+1}=\mathfrak{q}_k+(\tau_{k+1}-\tau_k)g(\mathfrak{q}_k),$$

$$\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_1L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})+\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}D_3L_d(q_{k-1},\mathfrak{q}_{k-1},q_k,\mathfrak{q}_k)=0,$$

$$\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_2L_{d_k}-\frac{1}{\tau_{k+1}-\tau_k}L_{d_k}+\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}D_4L_{d_{k-1}}+\frac{1}{\tau_k-\tau_{k-1}}L_{d_{k-1}}$$
$$=\frac{\lambda_{k-1}}{\tau_k-\tau_{k-1}}-\frac{\lambda_k}{\tau_{k+1}-\tau_k}-\lambda_k\nabla g(\mathfrak{q}_k),$$

where L_{d_k} *denotes* $L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})$.

Proof. We use the notation $L_{d_k}=L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})$, and we will use the fact that $\delta q_0=\delta q_N=\delta\mathfrak{q}_0=\delta\mathfrak{q}_N=0$ throughout the proof. We have

$$\begin{aligned}
\delta\bar{\mathfrak{S}}_d&=\delta\left(\sum_{k=0}^{N-1}\left\{\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\left[L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1})-\lambda_k\right]+\lambda_k g(\mathfrak{q}_k)\right\}\right)\\
&=\sum_{k=1}^{N-1}\left[\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_2L_{d_k}-\frac{1}{\tau_{k+1}-\tau_k}L_{d_k}+\frac{\lambda_k}{\tau_{k+1}-\tau_k}+\lambda_k\nabla g(\mathfrak{q}_k)\right]\delta\mathfrak{q}_k\\
&\quad+\sum_{k=0}^{N-2}\left[\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_4L_{d_k}+\frac{1}{\tau_{k+1}-\tau_k}L_{d_k}-\frac{\lambda_k}{\tau_{k+1}-\tau_k}\right]\delta\mathfrak{q}_{k+1}\\
&\quad+\sum_{k=1}^{N-1}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_1L_{d_k}\delta q_k+\sum_{k=0}^{N-2}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_3L_{d_k}\delta q_{k+1}\\
&\quad+\sum_{k=0}^{N-1}\left(g(\mathfrak{q}_k)-\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\right)\delta\lambda_k.
\end{aligned}$$

Thus,

$$\begin{aligned}\delta\bar{\mathfrak{S}}_d = \sum_{k=1}^{N-1}\left[\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_2L_{d_k} - \frac{1}{\tau_{k+1}-\tau_k}L_{d_k} + \frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}D_4L_{d_{k-1}} + \frac{1}{\tau_k-\tau_{k-1}}L_{d_{k-1}}\right]\delta\mathfrak{q}_k\\ + \sum_{k=1}^{N-1}\left[\frac{\lambda_k}{\tau_{k+1}-\tau_k} + \lambda_k\nabla g(\mathfrak{q}_k) - \frac{\lambda_{k-1}}{\tau_k-\tau_{k-1}}\right]\delta\mathfrak{q}_k\\ + \sum_{k=1}^{N-1}\left[\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_1L_{d_k} + \frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}D_3L_{d_{k-1}}\right]\delta q_k\\ + \sum_{k=0}^{N-1}\left(g(\mathfrak{q}_k) - \frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\right)\delta\lambda_k.\end{aligned}$$

As a consequence, if

$$\mathfrak{q}_{k+1} = \mathfrak{q}_k + (\tau_{k+1}-\tau_k)g(\mathfrak{q}_k),$$

$$\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_1L_d(q_k,\mathfrak{q}_k,q_{k+1},\mathfrak{q}_{k+1}) + \frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}D_3L_d(q_{k-1},\mathfrak{q}_{k-1},q_k,\mathfrak{q}_k) = 0,$$

$$\begin{aligned}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_2L_{d_k} - \frac{1}{\tau_{k+1}-\tau_k}L_{d_k} + \frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}D_4L_{d_{k-1}} + \frac{1}{\tau_k-\tau_{k-1}}L_{d_{k-1}}&\\ = \frac{\lambda_{k-1}}{\tau_k-\tau_{k-1}} - \frac{\lambda_k}{\tau_{k+1}-\tau_k} - \lambda_k\nabla g(\mathfrak{q}_k),&\end{aligned}$$

then

$$\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right) = 0.$$

Conversely, if

$$\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right) = 0,$$

then a discrete fundamental theorem of the calculus of variations yields the above discrete extended Euler–Lagrange equations. □

A.3.3 Theorem 4.8: Discrete Extended Euler–Lagrange Equations in the Lie Group Setting

Theorem A.14. *The Type I discrete Hamilton's variational principle,*

$$\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=0,$$

where,

$$\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=\sum_{k=0}^{N-1}\left[L_d(q_k,z_k,\mathfrak{q}_k,\mathfrak{q}_{k+1})-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k},$$

is equivalent to the discrete extended Euler–Lagrange equations,

$$\mathfrak{q}_{k+1}=\mathfrak{q}_k+(\tau_{k+1}-\tau_k)g(\mathfrak{q}_k),$$

$$\mathrm{Ad}^*_{z_k^{-1}}\left(\mathrm{T}^*_e\mathrm{L}_{z_k}D_2L_{d_k}\right)=\mathrm{T}^*_e\mathrm{L}_{q_k}D_1L_{d_k}+\frac{\tau_{k+1}-\tau_k}{\mathfrak{q}_{k+1}-\mathfrak{q}_k}\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}\mathrm{T}^*_e\mathrm{L}_{z_{k-1}}D_2L_{d_{k-1}},$$

and

$$\begin{aligned}&\left[D_3L_{d_k}+\lambda_k\frac{1}{\tau_{k+1}-\tau_k}+\lambda_k\nabla g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}-\frac{1}{\tau_{k+1}-\tau_k}\left[L_{d_k}-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\\&\quad+\left[D_4L_{d_k}-\lambda_{k-1}\frac{1}{\tau_k-\tau_{k-1}}\right]\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\frac{1}{\tau_k-\tau_{k-1}}\left[L_{d_{k-1}}-\lambda_{k-1}\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\lambda_{k-1}g(\mathfrak{q}_{k-1})\right]=0,\end{aligned}$$

where L_{d_k} denotes $L_d(q_k,z_k,\mathfrak{q}_k,\mathfrak{q}_{k+1})$.

Proof. We will use the notation $L_{d_k}=L_d(q_k,z_k,\mathfrak{q}_k,\mathfrak{q}_{k+1})$ and we will use the boundary conditions

$$\delta q_0=\delta q_N=\delta\mathfrak{q}_0=\delta\mathfrak{q}_N=\eta_0=\eta_N=0$$

throughout the proof. We have

$$\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=\delta\left(\sum_{k=0}^{N-1}\left[L_d(q_k,z_k,\mathfrak{q}_k,\mathfrak{q}_{k+1})-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\right),$$

so

$$\begin{aligned}\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right) = &\sum_{k=1}^{N-1}\left[D_3L_{d_k}+\lambda_k\frac{1}{\tau_{k+1}-\tau_k}+\lambda_k\nabla g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\delta\mathfrak{q}_k\\ &-\sum_{k=1}^{N-1}\frac{1}{\tau_{k+1}-\tau_k}\left[L_{d_k}-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\delta\mathfrak{q}_k\\ &+\sum_{k=0}^{N-2}\left[D_4L_{d_k}-\lambda_k\frac{1}{\tau_{k+1}-\tau_k}\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\delta\mathfrak{q}_{k+1}\\ &+\sum_{k=0}^{N-2}\frac{1}{\tau_{k+1}-\tau_k}\left[L_{d_k}-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\delta\mathfrak{q}_{k+1}\\ &+\sum_{k=0}^{N-1}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\left(g(\mathfrak{q}_k)-\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\right)\delta\lambda_k\\ &+\sum_{k=1}^{N-1}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_1L_{d_k}\delta q_k\\ &+\sum_{k=0}^{N-1}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}D_2L_{d_k}\delta z_k.\end{aligned}$$

We can write δg_k as $\delta g_k = g_k\eta_k$ for some $\eta_k \in \mathfrak{g}$. Then, taking the variation of the discrete kinematics equation $q_{k+1} = q_k z_k$ gives the equation $\delta q_{k+1} = \delta q_k z_k + q_k \delta z_k$ and $z_k = q_k^{-1}q_{k+1}$. Therefore,

$$\delta z_k = q_k^{-1}\delta q_{k+1} - q_k^{-1}\delta q_k z_k = q_k^{-1}q_{k+1}\eta_{k+1} - q_k^{-1}q_k\eta_k z_k = z_k\eta_{k+1} - \eta_k z_k,$$

so

$$\begin{aligned}\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right) = &\sum_{k=1}^{N-1}\left[D_3L_{d_k}+\lambda_k\frac{1}{\tau_{k+1}-\tau_k}+\lambda_k\nabla g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\delta\mathfrak{q}_k\\ &-\sum_{k=1}^{N-1}\frac{1}{\tau_{k+1}-\tau_k}\left[L_{d_k}-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\delta\mathfrak{q}_k\\ &+\sum_{k=1}^{N-1}\left[\left(D_4L_{d_{k-1}}-\lambda_{k-1}\frac{1}{\tau_k-\tau_{k-1}}\right)\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}\right]\delta\mathfrak{q}_k\\ &+\sum_{k=1}^{N-1}\left[\frac{1}{\tau_k-\tau_{k-1}}\left(L_{d_{k-1}}-\lambda_{k-1}\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\lambda_{k-1}g(\mathfrak{q}_{k-1})\right)\right]\delta\mathfrak{q}_k\\ &+\sum_{k=1}^{N-1}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\left(\mathrm{T}_e^*\mathrm{L}_{q_k}D_1L_{d_k}\bullet\eta_k\right)\\ &+\sum_{k=0}^{N-1}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\left(\mathrm{T}_e^*\mathrm{L}_{z_k}D_2L_{d_k}\bullet\left[\eta_{k+1}-z_k^{-1}\eta_k z_k\right]\right)\\ &+\sum_{k=0}^{N-1}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\left(g(\mathfrak{q}_k)-\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\right)\delta\lambda_k.\end{aligned}$$

Then,

$$\begin{aligned}
\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right) = &\sum_{k=1}^{N-1}\left[D_3L_{d_k}+\lambda_k\frac{1}{\tau_{k+1}-\tau_k}+\lambda_k\nabla g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\delta\mathfrak{q}_k\\
&-\sum_{k=1}^{N-1}\frac{1}{\tau_{k+1}-\tau_k}\left[L_{d_k}-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\delta\mathfrak{q}_k\\
&+\sum_{k=1}^{N-1}\left[\left(D_4L_{d_{k-1}}-\lambda_{k-1}\frac{1}{\tau_k-\tau_{k-1}}\right)\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}\right]\delta\mathfrak{q}_k\\
&+\sum_{k=1}^{N-1}\left[\frac{1}{\tau_k-\tau_{k-1}}\left(L_{d_{k-1}}-\lambda_{k-1}\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\lambda_{k-1}g(\mathfrak{q}_{k-1})\right)\right]\delta\mathfrak{q}_k\\
&+\sum_{k=1}^{N-1}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\left(\mathrm{T}_e^*\mathrm{L}_{q_k}D_1L_{d_k}\bullet\eta_k\right)\\
&+\sum_{k=0}^{N-1}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\left(g(\mathfrak{q}_k)-\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\right)\delta\lambda_k\\
&+\sum_{k=0}^{N-1}\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}\left(\mathrm{T}_e^*\mathrm{L}_{z_{k-1}}D_2L_{d_{k-1}}\bullet\eta_k\right)\\
&-\sum_{k=0}^{N-1}\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}\left(\mathrm{T}_e^*\mathrm{L}_{z_k}D_2L_{d_k}\bullet\mathrm{Ad}_{z_k^{-1}}\eta_k\right).
\end{aligned}$$

As a consequence, if

$$\mathfrak{q}_{k+1}=\mathfrak{q}_k+(\tau_{k+1}-\tau_k)g(\mathfrak{q}_k),$$

$$\mathrm{Ad}^*_{z_k^{-1}}\left(\mathrm{T}_e^*\mathrm{L}_{z_k}D_2L_{d_k}\right)=\mathrm{T}_e^*\mathrm{L}_{q_k}D_1L_{d_k}+\frac{\tau_{k+1}-\tau_k}{\mathfrak{q}_{k+1}-\mathfrak{q}_k}\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}\mathrm{T}_e^*\mathrm{L}_{z_{k-1}}D_2L_{d_{k-1}},$$

$$\begin{aligned}
&\left[D_3L_{d_k}+\lambda_k\frac{1}{\tau_{k+1}-\tau_k}+\lambda_k\nabla g(\mathfrak{q}_k)\right]\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}-\frac{1}{\tau_{k+1}-\tau_k}\left[L_{d_k}-\lambda_k\frac{\mathfrak{q}_{k+1}-\mathfrak{q}_k}{\tau_{k+1}-\tau_k}+\lambda_k g(\mathfrak{q}_k)\right]\\
&\quad+\left[D_4L_{d_k}-\lambda_{k-1}\frac{1}{\tau_k-\tau_{k-1}}\right]\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\frac{1}{\tau_k-\tau_{k-1}}\left[L_{d_{k-1}}-\lambda_{k-1}\frac{\mathfrak{q}_k-\mathfrak{q}_{k-1}}{\tau_k-\tau_{k-1}}+\lambda_{k-1}g(\mathfrak{q}_{k-1})\right]=0,
\end{aligned}$$

then $\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=0$.

Conversely, if $\delta\bar{\mathfrak{S}}_d\left(\{(q_k,\mathfrak{q}_k,\lambda_k)\}_{k=0}^N\right)=0$, then a discrete fundamental theorem of the calculus of variations yields the above equations. □

A.4 Derivations of the Euler–Lagrange Equations on Riemannian Manifolds

A.4.1 Theorem 4.1: Convex and Weakly Quasi-Convex Cases

Theorem A.15. *The Euler–Lagrange equation corresponding to the Lagrangian*

$$\mathcal{L}_{\alpha,\beta,\gamma}(X,V,t) = \frac{1}{2}e^{\lambda^{-1}\zeta\gamma_t-\alpha_t}\langle V,V\rangle - e^{\alpha_t+\beta_t+\lambda^{-1}\zeta\gamma_t}f(X),$$

is given by

$$\nabla_{\dot{X}}\dot{X} + \left(\lambda^{-1}\zeta e^{\alpha_t} - \dot{\alpha}_t\right)\dot{X} + e^{2\alpha_t+\beta_t}\mathrm{gradf}(X) = 0,$$

Proof. Consider a path on the manifold $\mathcal{Q}$ described in coordinates by

$$(x(t),\dot{x}(t)) = \left(q^1(t),\dots,q^n(t),v^1(t),\dots,v^n(t)\right).$$

Then, with $\langle\cdot,\cdot\rangle = \sum_{i,j=1}^n g_{ij}dx^i dx^j$, the Bregman Lagrangian $\mathcal{L}_{\alpha,\beta,\gamma}$ can be written as

$$\mathcal{L}_{\alpha,\beta,\gamma}(x(t),\dot{x}(t),t) = \frac{1}{2}e^{\lambda^{-1}\zeta\gamma_t-\alpha_t}\sum_{i,j=1}^n g_{ij}(x(t))v^i(t)v^j(t) - e^{\alpha_t+\beta_t+\lambda^{-1}\zeta\gamma_t}f(x(t)).$$

For $k = 1,\dots n$,

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial\mathcal{L}_{\alpha,\beta,\gamma}}{\partial v^k}(x(t),\dot{x}(t),t)\right) &= e^{\lambda^{-1}\zeta\gamma_t-\alpha_t}\sum_{i=1}^n g_{ik}(x(t))\frac{dv^i}{dt}(t)\\ &\quad + e^{\lambda^{-1}\zeta\gamma_t-\alpha_t}\sum_{i,j=1}^n\frac{\partial g_{kj}}{\partial q^i}(x(t))v^i(t)v^j(t)\\ &\quad + (\lambda^{-1}\zeta\dot{\gamma}_t - \dot{\alpha}_t)e^{\lambda^{-1}\zeta\gamma_t-\alpha_t}\sum_{i=1}^n g_{ik}(x(t))v^i(t),\end{aligned}$$

$$\frac{\partial\mathcal{L}_{\alpha,\beta,\gamma}}{\partial q^k}(x(t),\dot{x}(t),t) = \frac{1}{2}e^{\lambda^{-1}\zeta\gamma_t-\alpha_t}\sum_{i,j=1}^n\frac{\partial g_{ij}}{\partial q^k}(x(t))v^i(t)v^j(t) - e^{\alpha_t+\beta_t+\lambda^{-1}\zeta\gamma_t}\frac{\partial f}{\partial q^k}(x(t)).$$

Multiplying both terms by $e^{\alpha_t-\lambda^{-1}\zeta\gamma_t}$, the Euler–Lagrange equations (2.2) for the Bregman Lagrangian $\mathcal{L}_{\alpha,\beta,\gamma}$ are given, for $k = 1,\dots,n$, by

$$\begin{aligned}0 = \sum_{i=1}^n g_{ik}(x(t))\frac{dv^i}{dt}(t) + \sum_{i,j=1}^n\frac{\partial g_{kj}}{\partial q^i}(x(t))v^i(t)v^j(t) + (\lambda^{-1}\zeta\dot{\gamma}_t - \dot{\alpha}_t)\sum_{i=1}^n g_{ik}(x(t))v^i(t)\\ - \frac{1}{2}\sum_{i,j=1}^n\frac{\partial g_{ij}}{\partial q^k}(x(t))v^i(t)v^j(t) + e^{2\alpha_t+\beta_t}\frac{\partial f}{\partial q^k}(x(t)).\end{aligned}$$

Rearranging terms, and multiplying by the matrix (g^{ij}) which is the inverse of (g_{ij}), we get, for $k = 1, \dots n$, the equation

$$\left(\frac{dv^k}{dt}(t) + \sum_{i,j=1}^{n} \Gamma_{ij}^k(x(t))v^i(t)v^j(t)\right) + \left(\lambda^{-1}\zeta\dot{\gamma}_t - \dot{\alpha}_t\right)v^k(t) + e^{2\alpha_t+\beta_t}\left(\mathrm{gradf}(x(t))\right)^k = 0,$$

where Γ_{ij}^k are the Christoffel symbols given by

$$\Gamma_{ij}^k = \frac{1}{2}\sum_{l=1}^{n} g^{kl}\left[\frac{\partial g_{jl}}{\partial x^i} + \frac{\partial g_{li}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^l}\right].$$

This gives the desired Euler–Lagrange equation once we use the ideal scaling equation $\dot{\gamma}_t = e^{\alpha_t}$, and simplify. □

A.4.2 Theorem 4.4: Strongly Convex Case

Theorem A.16. *The Euler–Lagrange equation corresponding to the Lagrangian $\mathscr{L}^{SC}$ is given by*

$$\nabla_{\dot{X}}\dot{X} + \eta\dot{X} + \mathrm{gradf}(X) = 0.$$

Proof. Consider a path on the manifold $\mathfrak{Q}$ described in coordinates by

$$(x(t), \dot{x}(t)) = \left(q^1(t), \dots, q^n(t), v^1(t), \dots, v^n(t)\right).$$

Then, with $\langle\cdot,\cdot\rangle = \sum_{i,j=1}^n g_{ij}dx^i dx^j$, the Lagrangian $\mathscr{L}^{SC}$ can be written as

$$\mathscr{L}^{SC}(x(t), \dot{x}(t), t) = \frac{e^{\eta t}}{2}\sum_{i,j=1}^{n} g_{ij}(x(t))v^i(t)v^j(t) - e^{\eta t}f(x(t)).$$

For $k = 1, \dots n$,

$$\frac{d}{dt}\left(\frac{\partial\mathscr{L}^{SC}}{\partial v^k}(x(t), \dot{x}(t), t)\right) = e^{\eta t}\sum_{i=1}^{n} g_{ik}(x(t))\frac{dv^i}{dt}(t) + e^{\eta t}\sum_{i,j=1}^{n}\frac{\partial g_{kj}}{\partial q^i}(x(t))v^i(t)v^j(t) + \eta e^{\eta t}\sum_{i=1}^{n} g_{ik}(x(t))v^i(t),$$

$$\frac{\partial\mathscr{L}^{SC}}{\partial q^k}(x(t), \dot{x}(t), t) = e^{\eta t}\sum_{i,j=1}^{n}\frac{\partial g_{ij}}{\partial q^k}(x(t))v^i(t)v^j(t) - e^{\eta t}\frac{\partial f}{\partial q^k}(x(t)).$$

If we multiply both terms by $e^{-\eta t}$, the Euler–Lagrange equations (2.2) for the Lagrangian $\mathcal{L}^{SC}$ are given, for $k = 1, \ldots, n$, by

$$0 = \sum_{i=1}^{n} g_{ik}(x(t))\frac{dv^i}{dt}(t) + \sum_{i,j=1}^{n} \frac{\partial g_{kj}}{\partial q^i}(x(t))v^i(t)v^j(t) + \eta \sum_{i=1}^{n} g_{ik}(x(t))v^i(t) - \frac{1}{2}\sum_{i,j=1}^{n} \frac{\partial g_{ij}}{\partial q^k}(x(t))v^i(t)v^j(t) + \frac{\partial f}{\partial q^k}(x(t)).$$

Rearranging terms, and multiplying by the matrix (g^{ij}) which is the inverse of (g_{ij}), we get, for $k = 1, \ldots n$, the equation

$$\left(\frac{dv^k}{dt}(t) + \sum_{i,j=1}^{n} \Gamma_{ij}^k(x(t))v^i(t)v^j(t)\right) + \eta v^k(t) + (\text{gradf}(x(t)))^k = 0,$$

where Γ_{ij}^k are the Christoffel symbols given by

$$\Gamma_{ij}^k = \frac{1}{2}\sum_{l=1}^{n} g^{kl}\left[\frac{\partial g_{jl}}{\partial x^i} + \frac{\partial g_{li}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^l}\right],$$

which gives the desired Euler–Lagrange equation. □

A.5 Convergence Rates along the Riemannian Euler–Lagrange Equations (Theorem 4.2)

The proofs of the convergence rates of solutions to the Bregman Euler–Lagrange equations are inspired by those of Theorems 5 and 6 from [Alimisis et al., 2020b], and make use of Lemmas 2 and 12 therein:

Lemma A.5. *Given a Riemannian manifold $\mathcal{Q}$ with sectional curvature bounded above by $K_{\max}$ and below by $K_{\min}$, with ζ given by equation (4.9), and such that*

$$\text{diam}(\mathcal{Q}) < \begin{cases} \frac{\pi}{\sqrt{K_{\max}}} & \text{if } K_{\max} > 0 \\ \infty & \text{if } K_{\max} \leq 0 \end{cases},$$

we have that

$$\langle \nabla_{\dot{X}} \text{Log}_X(p), -\dot{X} \rangle \leq \zeta \|\dot{X}\|^2.$$

Lemma A.6. *Given a point q and a smooth curve $X(t)$ on a Riemannian manifold $\mathcal{Q}$, we have that*

$$\frac{d}{dt}\|\mathrm{Log}_{X(t)}(q)\|^2 \;=\; 2\langle \mathrm{Log}_{X(t)}(q), \nabla_{\dot{X}} \mathrm{Log}_{X(t)}(q)\rangle \;=\; 2\langle \mathrm{Log}_{X(t)}(q), -\dot{X}(t)\rangle.$$

Theorem A.17. *Suppose $f : \mathcal{Q} \to \mathbb{R}$ is λ-weakly quasi-convex function, and Assumption 1 is satisfied. Then, any solution $X(t)$ of the Bregman Euler–Lagrange equation*

$$\nabla_{\dot{X}}\dot{X} + \left(\lambda^{-1}\zeta e^{\alpha_t} - \dot{\alpha}_t\right)\dot{X} + e^{2\alpha_t+\beta_t}\mathrm{gradf}(X) = 0,$$

with $X(0) = x_0$ and $\dot{X}(0) = 0$, converges to a minimizer x^* of f with rate

$$f(X(t)) - f(x^*) \;\le\; \frac{2\lambda^2 e^{\beta_0}\left(f(x_0) - f(x^*)\right) + \zeta\|\mathrm{Log}_{x_0}(x^*)\|^2}{2\lambda^2 e^{\beta_t}}.$$

Proof. Let

$$\mathcal{E}(t) \;=\; \lambda^2 e^{\beta_t}\left(f(X) - f(x^*)\right) + \frac{1}{2}(\zeta-1)\|\mathrm{Log}_X(x^*)\|^2 + \frac{1}{2}\left\|\lambda e^{-\alpha_t}\dot{X} - \mathrm{Log}_X(x^*)\right\|^2.$$

Then, from Lemma A.6,

$$\frac{d}{dt}\|\mathrm{Log}_X(x^*)\|^2 = 2\langle \mathrm{Log}_X(x^*), \nabla_{\dot{X}}\mathrm{Log}_X(x^*)\rangle = 2\langle \mathrm{Log}_X(x^*), -\dot{X}\rangle,$$

so

$$\begin{aligned}
\dot{\mathcal{E}}(t) &= \lambda^2\dot{\beta}_t e^{\beta_t}\left(f(X) - f(x^*)\right) + \lambda^2 e^{\beta_t}\langle \mathrm{gradf}(X), \dot{X}\rangle + (\zeta-1)\langle \mathrm{Log}_X(x^*), -\dot{X}\rangle \\
&\qquad + \langle \lambda e^{-\alpha_t}\dot{X} - \mathrm{Log}_X(x^*), -\dot{\alpha}_t\lambda e^{-\alpha}\dot{X} + \lambda e^{-\alpha_t}\nabla_{\dot{X}}\dot{X} - \nabla_{\dot{X}}\mathrm{Log}_X(x^*)\rangle \\
&= \lambda^2\dot{\beta}_t e^{\beta_t}\left(f(X) - f(x^*)\right) + \lambda^2 e^{\beta_t}\langle \mathrm{gradf}(X), \dot{X}\rangle + (\zeta-1)\langle \mathrm{Log}_X(x^*), -\dot{X}\rangle \\
&\qquad + \langle \lambda e^{-\alpha_t}\dot{X} - \mathrm{Log}_X(x^*), \lambda e^{-\alpha_t}\left(-\dot{\alpha}_t\dot{X} + \nabla_{\dot{X}}\dot{X}\right) - \nabla_{\dot{X}}\mathrm{Log}_X(x^*)\rangle.
\end{aligned}$$

Now, from the Bregman Euler–Lagrange equation, we have that

$$-\dot{\alpha}_t\dot{X} + \nabla_{\dot{X}}\dot{X} \;=\; -\lambda^{-1}\zeta e^{\alpha_t}\dot{X} - e^{2\alpha_t+\beta_t}\mathrm{gradf}(X).$$

As a result,

$$\begin{aligned}
\dot{\mathcal{E}}(t) &= \lambda^2\dot{\beta}_t e^{\beta_t}\left(f(X) - f(x^*)\right) + \lambda^2 e^{\beta_t}\langle \mathrm{gradf}(X), \dot{X}\rangle + (\zeta-1)\langle \mathrm{Log}_X(x^*), -\dot{X}\rangle \\
&\qquad + \langle \lambda e^{-\alpha_t}\dot{X} - \mathrm{Log}_X(x^*), -\zeta\dot{X} - \lambda e^{\alpha_t+\beta_t}\mathrm{gradf}(X) - \nabla_{\dot{X}}\mathrm{Log}_X(x^*)\rangle.
\end{aligned}$$

This can be rewritten as

$$\begin{aligned}\dot{\mathcal{E}}(t) &= \lambda^2\dot{\beta}_t e^{\beta_t}\left(f(X)-f(x^*)\right)+\lambda^2 e^{\beta_t}\langle \operatorname{gradf}(X),\dot{X}\rangle \\ &\quad +(\zeta-1)\langle \operatorname{Log}_X(x^*),-\dot{X}\rangle-\lambda\zeta e^{-\alpha_t}\langle \dot{X},\dot{X}\rangle \\ &\quad +\lambda e^{\alpha_t+\beta_t}\langle \operatorname{Log}_X(x^*),\operatorname{gradf}(X)\rangle+\langle \operatorname{Log}_X(x^*),\nabla_{\dot{X}}\operatorname{Log}_X(x^*)\rangle \\ &\quad -\lambda^2 e^{\beta_t}\langle \dot{X},\operatorname{gradf}(X)\rangle-\lambda e^{-\alpha_t}\langle \dot{X},\nabla_{\dot{X}}\operatorname{Log}_X(x^*)\rangle+\zeta\langle \operatorname{Log}_X(x^*),\dot{X}\rangle.\end{aligned}$$

Canceling the $\langle \operatorname{gradf}(X),\dot{X}\rangle$ and $\langle \operatorname{Log}_X(x^*),-\dot{X}\rangle$ terms out using Lemma A.6, we get

$$\begin{aligned}\dot{\mathcal{E}}(t) &= \lambda^2\dot{\beta}_t e^{\beta_t}\left(f(X)-f(x^*)\right)+\lambda e^{\alpha_t+\beta_t}\langle \operatorname{Log}_X(x^*),\operatorname{gradf}(X)\rangle \\ &\quad -\lambda\zeta e^{-\alpha_t}\langle \dot{X},\dot{X}\rangle-\lambda e^{-\alpha_t}\langle \dot{X},\nabla_{\dot{X}}\operatorname{Log}_X(x^*)\rangle \\ &= \lambda e^{\beta_t}\left[\dot{\beta}_t\lambda\left(f(X)-f(x^*)\right)+e^{\alpha_t}\langle \operatorname{Log}_X(x^*),\operatorname{gradf}(X)\rangle\right] \\ &\quad -\lambda e^{-\alpha_t}\left[\zeta\langle \dot{X},\dot{X}\rangle+\langle \dot{X},\nabla_{\dot{X}}\operatorname{Log}_X(x^*)\rangle\right].\end{aligned}$$

Now, since f is geodesically λ-weakly quasi-convex, we have that

$$\lambda\left(f(X)-f(x^*)\right)+\langle \operatorname{Log}_X(x^*),\operatorname{gradf}(X)\rangle\le 0,$$

so the ideal scaling equation $\dot{\beta}_t\le e^{\alpha_t}$ implies that

$$\lambda e^{\beta_t}\left[\dot{\beta}_t\lambda\left(f(X)-f(x^*)\right)+e^{\alpha_t}\langle \operatorname{Log}_X(x^*),\operatorname{gradf}(X)\rangle\right]\le 0.$$

Moreover, Lemma A.5 yields $\left[\zeta\langle \dot{X},\dot{X}\rangle+\langle \dot{X},\nabla_{\dot{X}}\operatorname{Log}_X(x^*)\rangle\right]\ge 0$, so

$$-\lambda e^{-\alpha_t}\left[\zeta\langle \dot{X},\dot{X}\rangle+\langle \dot{X},\nabla_{\dot{X}}\operatorname{Log}_X(x^*)\rangle\right]\le 0.$$

Therefore, $\dot{\mathcal{E}}(t)\le 0$, and so

$$\begin{aligned}\lambda^2 e^{\beta_t}\left(f(X)-f(x^*)\right) &\le \lambda^2 e^{\beta_t}\left(f(X)-f(x^*)\right)+\frac{1}{2}(\zeta-1)\|\operatorname{Log}_X(x^*)\|^2 \\ &\quad +\frac{1}{2}\left\|\lambda e^{-\alpha_t}\dot{X}-\operatorname{Log}_X(x^*)\right\|^2 \\ &= \mathcal{E}(t)\le\mathcal{E}(0)=\lambda^2 e^{\beta_0}\left(f(x_0)-f(x^*)\right)+\frac{1}{2}\zeta\|\operatorname{Log}_{x_0}(x^*)\|^2,\end{aligned}$$

which gives the desired rate of convergence

$$f(X(t))-f(x^*)\le\frac{2\lambda^2 e^{\beta_0}\left(f(x_0)-f(x^*)\right)+\zeta\|\operatorname{Log}_{x_0}(x^*)\|^2}{2\lambda^2 e^{\beta_t}}.$$

□

A.6 Existence of Solutions to the Riemannian Euler–Lagrange Equations

A.6.1 Theorem 4.3: Convex and Weakly Quasi-Convex Cases

Theorem A.18. *Suppose Assumption 1 is satisfied, and let $C, p > 0$ and $v > 1$ be given constants. Then the differential equation*

$$\nabla_{\dot{X}}\dot{X} + \frac{v}{t}\dot{X} + Ct^{p-2}\mathrm{gradf}(X) = 0,$$

has a global solution $X : [0, \infty) \to \mathcal{Q}$ under the initial conditions $X(0) = x_0 \in \mathcal{Q}$ and $\dot{X}(0) = 0$.

Proof. The proof is similar to that of Lemma 3 in [Alimisis et al., 2020b], which extended Theorem 1 in [Su et al., 2016] to the Riemannian setting. We first define a family of smoothed equations for which we then show existence of a solution for all time. After choosing an equicontinuous and uniformly bounded subfamily of smoothed solutions, we use the Arzela–Ascoli Theorem on the complete Riemannian manifold $\mathcal{Q}$ to obtain a subsequence converging uniformly, and argue that the limit of this subsequence solves the original problem. When $p = 2$, we recover the simpler case considered in Lemma 3 of [Alimisis et al., 2020b], so we assume $p \neq 2$ in this proof. Consider the following families of smoothed equations for $\delta > 0$:

$$\nabla_{\dot{X}}\dot{X} + \frac{v}{\max(\delta, t)}\dot{X} + C(\max(\delta, t))^{p-2}\mathrm{gradf}(X) = 0 \qquad \text{if } p < 2,$$
$$\nabla_{\dot{X}}\dot{X} + \frac{v}{\max(\delta, t)}\dot{X} + Ct^{p-2}\mathrm{gradf}(X) = 0 \qquad \text{if } p > 2.$$

Exp and Log are defined globally on $\mathcal{Q}$ by Assumption 1, so we can choose geodesically normal coordinates $\phi = \psi^{-1}$ around x_0 defined globally on $\mathcal{Q}$ and put $c = \phi \circ X$.

Using the smoothness of f and letting $u = \dot{c}$ gives a system of first-order ODEs defining a local representation for a vector field in $T\mathcal{Q}$, and Section IV.3 of [Lang, 1999] guarantees that the smoothed ODE has a unique solution X_δ locally around 0. Actually, X_δ exists on $[0, \infty)$. Indeed, by contradiction, let $[0, T)$ be the maximal interval of existence of X_δ, for some finite $T > 0$.

Using

$$\frac{d}{dt}f(X_\delta(t)) = \langle \text{gradf}(X_\delta), \dot{X}_\delta \rangle$$

gives

$$\text{if } \delta > t,\ p < 2: \qquad \frac{d}{dt}f(X_\delta) = -\frac{\delta^{2-p}}{C}\langle \nabla_{\dot{X}_\delta}\dot{X}_\delta, \dot{X}_\delta\rangle - \frac{v\delta^{1-p}}{C}\langle \dot{X}_\delta, \dot{X}_\delta\rangle$$
$$= -\frac{\delta^{2-p}}{2C}\frac{d}{dt}\|\dot{X}_\delta\|^2 - \frac{v\delta^{1-p}}{C}\|\dot{X}_\delta\|^2$$

$$\text{if } \delta > t,\ p > 2: \qquad \frac{d}{dt}f(X_\delta) = -\frac{t^{2-p}}{C}\langle \nabla_{\dot{X}_\delta}\dot{X}_\delta, \dot{X}_\delta\rangle - \frac{vt^{2-p}}{C\delta}\langle \dot{X}_\delta, \dot{X}_\delta\rangle$$
$$= -\frac{t^{2-p}}{2C}\frac{d}{dt}\|\dot{X}_\delta\|^2 - \frac{vt^{2-p}}{C\delta}\|\dot{X}_\delta\|^2,$$

$$\text{if } \delta < t: \qquad \frac{d}{dt}f(X_\delta) = -\frac{t^{2-p}}{C}\langle \nabla_{\dot{X}_\delta}\dot{X}_\delta, \dot{X}_\delta\rangle - \frac{vt^{1-p}}{C}\langle \dot{X}_\delta, \dot{X}_\delta\rangle$$
$$= -\frac{1}{2C}\frac{d}{dt}\left(t^{2-p}\|\dot{X}_\delta\|^2\right) - \frac{2v(2-p)-1}{2C(2-p)}t^{1-p}\|\dot{X}_\delta\|^2.$$

Now, define

$$\theta = \frac{2v(2-p)-1}{2C(2-p)}.$$

Integrating and using the Cauchy-Schwarz inequality for the $p < 2$ case gives

$$\int_0^T \sqrt{(\max(\delta,t))^{1-p}}\|\dot{X}_\delta\|dt = \int_0^\delta \sqrt{\delta^{1-p}}\|\dot{X}_\delta\|dt + \int_\delta^T \sqrt{t^{1-p}}\|\dot{X}_\delta\|dt$$
$$\leq \sqrt{\frac{C\delta}{v}(f(x_0) - \inf_u f(u)) + \frac{\delta^{2-p}}{2v}\left(\|\dot{X}_\delta(0)\|^2 - \inf_{t\in[0,T)}\|\dot{X}_\delta(t)\|^2\right)}$$
$$+ \sqrt{\frac{T-\delta}{\theta}(f(X_\delta(\delta)) - \inf_u f(u)) + \frac{T-\delta}{2C\theta}\left(\delta^{2-p}\|\dot{X}_\delta(\delta)\|^2 - \inf_{t\in[0,T)} t^{2-p}\|\dot{X}_\delta(t)\|^2\right)}$$
$$< \infty,$$

since f is bounded below by Assumption 1. If $\delta \geq T$, then $\sqrt{\delta^{1-p}}\dot{X}_\delta$ is integrable on $[0,T)$. If $\delta < T$, then the integrals on $[0,T)$ and $[0,\delta)$ are finite, so the integral on $[\delta,T)$ must also be finite, and thus $\sqrt{t^{1-p}}\dot{X}_\delta$ is integrable on $[\delta,T)$. Now, $\|\int_a^T \dot{X}_\delta dt\| \leq \int_a^T \|\dot{X}_\delta\|dt < \infty$ for $a = 0, \delta$ implies that $\lim_{t\to T} X_\delta(t)$ exists. Since $\mathcal{Q}$ is complete by Assumption 1, the limit is in $\mathcal{Q}$, contradicting the maximality of $[0,T)$. The $p > 2$ case is similar: the integrand is

replaced by $\sqrt{t^{2-p}(\max(\delta,t))^{-1}}\|\dot{X}_\delta\|$, and the integral on $[\delta,T)$ remains unchanged while the integral on $[0,\delta)$ can be bounded by the same expression using $t<\delta$. Thus, in both cases, we can find a solution $X_\delta:[0,\infty)\to\mathcal{Q}$ to the smooth initial-value ODE, and its corresponding solution $X_\delta:[0,\infty)\to\mathbb{R}^n$ in local coordinates.

Now define

$$M_\delta(t)=\sup_{u\in(0,t]}\frac{\|\dot{X}_\delta(u)\|}{u}.$$

When $0<t\le\delta$, the smoothed ODE can be written as

$$\begin{aligned}\nabla_{\dot{X}_\delta}\left(\dot{X}_\delta e^{\frac{v}{\delta}}\right)&=-C\delta^{p-2}\mathrm{gradf}(X_\delta)e^{\frac{v}{\delta}} &&\text{if } p<2,\\ \nabla_{\dot{X}_\delta}\left(\dot{X}_\delta e^{\frac{v}{\delta}}\right)&=-Ct^{p-2}\mathrm{gradf}(X_\delta)e^{\frac{v}{\delta}} &&\text{if } p>2.\end{aligned}$$

Thus, we can use Lemma 4 in [Alimisis et al., 2020b] to get for $p>2$ that

$$\begin{aligned}\Gamma_{X_\delta(t)}^{x_0}\dot{X}_\delta(t)=-e^{-\frac{v}{\delta}t}\int_0^t\left(\Gamma_{X_\delta(u)}^{x_0}\mathrm{gradf}(X_\delta(u))-\Gamma_{X_\delta(u)}^{x_0}\Gamma(X_\delta)_{x_0}^{X_\delta(u)}\mathrm{gradf}(x_0)\right)Cu^{p-2}e^{\frac{v}{\delta}u}du\\ -e^{-\frac{v}{\delta}t}\int_0^t Cu^{p-2}\Gamma_{X_\delta(u)}^{x_0}\Gamma(X_\delta)_{x_0}^{X_\delta(u)}\mathrm{gradf}(x_0)e^{\frac{v}{\delta}u}du.\end{aligned}$$

From the Lipschitz assumption on f, we have that

$$\left\|\mathrm{gradf}(X_\delta(u))-\Gamma_{x_0}^{X_\delta(u)}\mathrm{gradf}(x_0)\right\|\le L\int_0^u\|\dot{X}_\delta(s)\|ds=L\int_0^u s\frac{\|\dot{X}_\delta(s)\|}{s}ds\le\frac{1}{2}LM_\delta(u)u^2.$$

Thus, since parallel transport preserves inner products,

$$\begin{aligned}\frac{\|\dot{X}_\delta(t)\|}{t}&\le\left(\frac{1}{2}CLM_\delta(\delta)\delta^p+C\delta^p\|\mathrm{gradf}(x_0)\|\right)\frac{e^{-\frac{v}{\delta}t}}{t}\int_0^t e^{\frac{v}{\delta}u}du\\ &\le\left(\frac{1}{2}CLM_\delta(\delta)\delta^p+C\delta^p\|\mathrm{gradf}(x_0)\|\right)\frac{\delta}{vt}(1-e^{-\frac{v}{\delta}t})\\ &\le\frac{1}{2}CLM_\delta(\delta)\delta^p+C\delta^p\|\mathrm{gradf}(x_0)\|.\end{aligned}$$

Taking the supremum over $0<t\le\delta$ and rearranging gives for $\delta<\delta_M=\left(\frac{2}{CL}\right)^{\frac{1}{p}}$ that

$$M_\delta(\delta)\le\frac{2C\delta^p\|\mathrm{gradf}(x_0)\|}{2-CL\delta^p}.$$

The case $p<2$ is done exactly in the same way except that we do not need to bound u^{p-2} by δ^{p-2} in the integrals since the t^{p-2} term in the differential equation is already replaced by δ^{p-2}.

Note that when $\delta < \delta_M$ and $\delta < t < t_M = \left(\frac{2(v+p+1)}{CL}\right)^{\frac{1}{p}}$, the smoothed ODE can be rewritten as

$$\frac{d}{dt}\left(t^v \dot{X}_\delta(t)\right) = -Ct^{v+p-2}\mathrm{gradf}(X_\delta).$$

Therefore, we can use Lemma 4 in [Alimisis et al., 2020b] once again to obtain

$$\begin{aligned}
&\Gamma_{X_\delta(t)}^{X_\delta(\delta)} t^v \dot{X}_\delta(t) - \delta^v \dot{X}_\delta(\delta) \\
&\qquad = \int_0^t \left(\Gamma_{X_\delta(u)}^{X_\delta(\delta)}\mathrm{gradf}(X_\delta(u)) - \Gamma_{X_\delta(u)}^{X_\delta(\delta)}\Gamma(X_\delta)_{x_0}^{X_\delta(u)}\mathrm{gradf}(x_0)\right) Cu^{v+p-2}du \\
&\qquad\qquad - \int_0^t Cu^{v+p-2}\Gamma_{X_\delta(u)}^{X_\delta(\delta)}\Gamma(X_\delta)_{x_0}^{X_\delta(u)}\mathrm{gradf}(x_0)du.
\end{aligned}$$

Using the fact that parallel transport preserves inner products, and dividing by t^{v+1} gives

$$\begin{aligned}
\frac{\|\dot{X}_\delta(t)\|}{t} &\leq \frac{\delta^{v+1}}{t^{v+1}}\frac{\|\dot{X}_\delta(\delta)\|}{\delta} + \frac{CL}{2t^{v+1}}\int_\delta^t M_\delta(u)u^{v+p}du + \frac{C}{t^{v+1}}\|\mathrm{gradf}(x_0)\|\int_\delta^t u^{v+p-2}du \\
&\leq \frac{\delta^{v+1}}{t^{v+1}}\frac{2C\delta^p\|\mathrm{gradf}(x_0)\|}{2-CL\delta^p} + \frac{CL}{2(v+p+1)}M_\delta(t)t^p + \frac{C(t^{v+p-1}-\delta^{v+p-1})}{(v+p-1)t^{v+1}}\|\mathrm{gradf}(x_0)\|,
\end{aligned}$$

and since this upper bound is an increasing function of t, we have for any $t' \in (\delta, t)$ that

$$\frac{\|\dot{X}_\delta(t')\|}{t'} \;\leq\; \frac{2C\delta^p\|\mathrm{gradf}(x_0)\|}{2-CL\delta^p} + \frac{CL}{2(v+p+1)}M_\delta(t)t^p + \frac{Ct^{p-2}}{v+p-1}\|\mathrm{gradf}(x_0)\|.$$

Taking the supremum over all $t' \in (0, t)$ gives for $\delta < \delta_M$ and $\delta < t < t_M$,

$$M_\delta(t) \;\leq\; \frac{1}{1-\frac{CL}{2(v+p+1)}t^p}\left(\frac{2C\delta^p}{2-CL\delta^p} + \frac{Ct^{p-2}}{v+p-1}\right)\|\mathrm{gradf}(x_0)\|.$$

Now consider the family of functions

$$\mathcal{F} = \left\{X_\delta : [0,T] \to \mathbb{R} \;\middle|\; \delta = 2^{-n}\tilde{\delta}, n = 0, 1, \dots\right\},$$

where

$$T = \left(\frac{v+p+1}{CL}\right)^{\frac{1}{p}}, \qquad \text{and } \tilde{\delta} = \left(\frac{1}{CL}\right)^{\frac{1}{p}}.$$

By definition of M_δ, we have for $t \in [0,T]$ and $\delta \in (0, \tilde{\delta})$ that

$$\|\dot{X}_\delta\| \;\leq\; TM_\delta(T) \;\leq\; 2CT\left(\tilde{\delta} + \frac{CT^{p-2}}{v+p-1}\right),$$

and

$$d(X_\delta(t), X_\delta(0)) \;\leq\; \int_0^t \|\dot{X}_\delta(u)\|du \;\leq\; t\|\dot{X}_\delta\| \;\leq\; T\|\dot{X}_\delta\|.$$

Therefore, $\mathcal{F}$ is equicontinuous and uniformly bounded, and the Riemannian manifold $\mathcal{Q}$ is complete by Assumption 1. Therefore, by the Arzela–Ascoli Theorem on Riemannian manifolds (Theorem 17 in [Kelley, 1975]), $\mathcal{F}$ contains a subsequence that converges uniformly on $[0,T]$ to some function X^*. The same argument as in part 5 of the proof of Lemma 3 of [Alimisis et al., 2020b] shows that X^* is a solution to the original initial-value ODE on $[0,T]$ which can then be extended to get a global solution on $[0,\infty)$. □

A.6.2 Theorem 4.5: Strongly Convex Case

Theorem A.19. *Suppose that Assumption 1 is satisfied, and that $\eta > 0$ is a given constant. Then, the differential equation*

$$\nabla_{\dot{X}}\dot{X} + \eta\dot{X} + \text{gradf}(X) = 0,$$

has a global solution $X : [0,\infty) \to \mathcal{Q}$ under the initial conditions $X(0) = x_0 \in \mathcal{Q}$ and $\dot{X}(0) = 0$.

Proof. Exp and Log are defined globally on $\mathcal{Q}$ by Assumption 1, so we can choose geodesically normal coordinates $\phi = \psi^{-1}$ around x_0 defined globally on $\mathcal{Q}$ and put $c = \phi \circ X$. As in [Alimisis et al., 2020b], using the smoothness of f and letting $u = \dot{c}$ gives a system of first-order ODEs which defines a local representation for a vector field in $T\mathcal{Q}$, and results from Section IV.3 of [Lang, 1999] guarantee that the initial-value differential equation has a unique solution locally around 0. It remains to show that this solution actually exists on $[0,\infty)$. Towards contradiction, suppose $[0,T)$ is the maximal interval of existence of the solution X, for some finite $T > 0$. Then,

$$\frac{d}{dt}f(X(t)) \;=\; \langle \text{gradf}(X), \dot{X}\rangle \;=\; -\langle \nabla_{\dot{X}}\dot{X}, \dot{X}\rangle - C\langle \dot{X}, \dot{X}\rangle \;=\; -\frac{1}{2}\frac{d}{dt}\|\dot{X}\|^2 - C\|\dot{X}\|^2.$$

Rearranging, integrating both sides and using the Cauchy-Schwarz inequality gives

$$\int_0^T \|\dot{X}\|dt \;=\; \sqrt{T(f(x_0) - \inf_u f(u)) + \frac{T}{2}\left(\|\dot{X}(0)\|^2 - \inf_{t\in[0,T)}\|\dot{X}(t)\|^2\right)} \;<\; \infty,$$

since f is bounded from below by Assumption 1. Therefore, $\lim_{t\to T} X(t)$ exists, and since the Riemannian manifold $\mathcal{Q}$ is complete, the limit is in $\mathcal{Q}$, contradicting the maximality of the interval $[0,T)$. This completes the proof. □

A.7 Time-Invariance Theorem 4.6 on Riemannian Manifolds

Theorem A.20. *Suppose that Assumption 1 is satisfied and that the curve $X(t)$ satisfies the Euler–Lagrange equation* (4.12) *corresponding to the Riemannian Bregman Lagrangian $\mathcal{L}_{\alpha,\beta,\gamma}$. Then the reparametrized curve $X(\tau(t))$ satisfies the Euler–Lagrange equation* (4.12) *corresponding to the modified Riemannian Bregman Lagrangian $\mathcal{L}_{\tilde{\alpha},\tilde{\beta},\tilde{\gamma}}$ where $\tilde{\alpha}_t = \alpha_{\tau(t)} + \log \dot{\tau}(t)$, $\tilde{\beta}_t = \beta_{\tau(t)}$, and $\tilde{\gamma}_t = \gamma_{\tau(t)}$. Furthermore, the parameters α, β, γ satisfy the ideal scaling conditions* (3.7) *if and only if $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$ do.*

Proof. Let $Y(t) = X(\tau(t))$. Then

$$\dot{Y}(t) = \dot{\tau}(t)\dot{X}(\tau(t)), \qquad \text{and} \qquad \nabla_{\dot{Y}(t)}\dot{Y}(t) = \ddot{\tau}(t)\dot{X}(\tau(t)) + \dot{\tau}^2(t)\nabla_{\dot{X}(\tau(t))}\dot{X}(\tau(t)).$$

Inverting these relations gives

$$\dot{X}(\tau(t)) = \frac{1}{\dot{\tau}(t)}\dot{Y}(t), \qquad \text{and} \qquad \nabla_{\dot{X}(\tau(t))}\dot{X}(\tau(t)) = \frac{1}{\dot{\tau}^2(t)}\nabla_{\dot{Y}(t)}\dot{Y}(t) - \frac{\ddot{\tau}(t)}{\dot{\tau}^3(t)}\dot{Y}(t).$$

The Bregman Euler–Lagrange equation (4.12) at time $\tau(t)$ is given by

$$\nabla_{\dot{X}(\tau(t))}\dot{X}(\tau(t)) + \left(\lambda^{-1}\zeta e^{\alpha_{\tau(t)}} - \dot{\alpha}_{\tau(t)}\right)\dot{X}(\tau(t)) + e^{2\alpha_{\tau(t)}+\beta_{\tau(t)}}\mathrm{gradf}(X(\tau(t))) = 0.$$

Substituting the expressions for $X(\tau(t)), \dot{X}(\tau(t))$ and $\nabla_{\dot{X}(\tau(t))}\dot{X}(\tau(t))$ in terms of $Y(t)$ and its derivatives, and multiplying by $\dot{\tau}^2(t)$, we get

$$\nabla_{\dot{Y}(t)}\dot{Y}(t) - \frac{\ddot{\tau}(t)}{\dot{\tau}(t)}\dot{Y}(t) + \left(\lambda^{-1}\zeta e^{\alpha_{\tau(t)}} - \dot{\alpha}_{\tau(t)}\right)\dot{\tau}(t)\dot{Y}(t) + \dot{\tau}^2(t)e^{2\alpha_{\tau(t)}+\beta_{\tau(t)}}\mathrm{gradf}(Y(t)) = 0.$$

Substituting the expressions for α, β, γ in terms of $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$ yields

$$\nabla_{\dot{Y}(t)}\dot{Y}(t) - \frac{\ddot{\tau}(t)}{\dot{\tau}(t)}\dot{Y}(t) + \left(\lambda^{-1}\zeta\frac{1}{\dot{\tau}(t)}e^{\tilde{\alpha}_t} - \frac{1}{\dot{\tau}(t)}\left[\dot{\tilde{\alpha}}(t) + \frac{\ddot{\tau}(t)}{\dot{\tau}(t)}\right]\right)\dot{\tau}(t)\dot{Y}(t) + e^{2\tilde{\alpha}_t+\tilde{\beta}_t}\mathrm{gradf}(Y(t)) = 0.$$

This gives the Bregman Euler–Lagrange equation (4.12) corresponding to $\mathcal{L}_{\tilde{\alpha},\tilde{\beta},\tilde{\gamma}}$,

$$\nabla_{\dot{Y}(t)}\dot{Y}(t) + \left(\lambda^{-1}\zeta e^{\tilde{\alpha}_t} - \frac{1}{\dot{\tau}(t)}\dot{\tilde{\alpha}}(t)\right)\dot{Y}(t) + e^{2\tilde{\alpha}_t+\tilde{\beta}_t}\mathrm{gradf}(Y(t)) = 0.$$

The fact that α, β, γ satisfy the ideal scaling conditions (3.7) if and only if $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$ do is established in the proof of Theorem 1.2 of [Wibisono et al., 2016]. □

A.8 Derivation: SE(3) Forced Variational Integrator

We derive here the forced discrete Euler–Lagrange equations in Lagrangian form (equations (6.40)-(6.42)) and in Hamiltonian form (equations (6.14)-(6.18)) associated to the discrete Lagrangian L_d and discrete control forces $f_d^{\pm}$ on SE(3) presented in Section 6.3.2.

On SE(3), with $g_k = (x_k, R_k) \in \mathrm{SE}(3)$ and $z_k = (y_k, Z_k) \in \mathrm{SE}(3)$, the discrete kinematics equations $g_{k+1} = g_k \star z_k$ are given by

$$R_{k+1} = R_k Z_k \qquad \text{and} \qquad x_{k+1} = x_k + R_k y_k, \tag{A.16}$$

so that the sequence $\{(x_k, R_k)\}_k$ remains on SE(3).

Using the continuous kinematics equation $\dot{R} = RS(\omega)$, the matrix $S(\omega_k)$ can be approximated via

$$S(\omega_k) \;=\; R_k^\top \dot{R}_k \;\approx\; R_k^\top \frac{R_{k+1} - R_k}{h} \;=\; \frac{1}{h}(Z_k - \mathbb{I}_3). \tag{A.17}$$

Recall that the discrete Lagrangian is then chosen to be

$$\begin{aligned} L_d(x_k, R_k, y_k, Z_k) = \frac{m}{2h} y_k^\top y_k + \frac{1}{h}\mathrm{Trace}\left([\mathbb{I}_3 - Z_k] J_d\right) \\ - (1-\alpha) h U(x_k, R_k) - \alpha h U(x_k + R_k y_k, R_k Z_k). \end{aligned} \tag{A.18}$$

We will derive the forced discrete equations of motion directly from the discrete Lagrange–d'Alembert variational principle.

The variations of the discrete position variables x_k and R_k are chosen to respect the geometry of the configuration space SE(3). The variation of x_k is given by $x_k^\epsilon \approx x_k + \epsilon \delta x_k$ where $\delta x_k \in \mathbb{R}^3$ vanishes at the endpoints $k = 0$ and $k = N$. The variation of R_k is given by $\delta R_k = R_k \eta_k$ where $\eta_k \in \mathfrak{so}(3)$ is a variation that vanishes at the endpoints $k = 0$ and $k = N$. Then, from $Z_k = R_k^\top R_{k+1}$, we get

$$\delta Z_k \;=\; R_k^\top \delta R_{k+1} + \delta R_k^\top R_{k+1} \;=\; R_k^\top R_{k+1} \eta_{k+1} - \eta_k R_k + R_k \eta_{k+1} \;=\; Z_k \eta_{k+1} - \eta_k Z_k. \tag{A.19}$$

Using the notation $L_{d_k} = L_d(g_k, z_k)$, $U_k = U(x_k, R_k)$, and $f^{\pm}_{d_k} = f^{\pm}_d(g_k, g_{k+1}, u_k)$, the discrete Lagrange–d'Alembert principle

$$\delta \sum_{k=1}^{N-1} L_d(g_k, z_k) + \sum_{k=1}^{N-1} \left[f^-_d(g_k, g_{k+1}, u_k) \cdot \delta g_k + f^+_d(g_k, g_{k+1}, u_k) \cdot \delta g_{k+1} \right] = 0, \tag{A.20}$$

yields

$$\begin{aligned}
0 &= \delta \sum_k L_{d_k} + \sum_k \left[f^-_{d_k} \cdot \delta g_k + f^+_{d_k} \cdot \delta g_{k+1} \right] \\
&= \frac{1}{h} \sum_k \mathrm{Trace}\left((\eta_k Z_k - Z_k \eta_{k+1}) J_d \right) + \frac{m}{h} \sum_k y_k^\top \delta y_k \\
&\quad - (1-\alpha) h \sum_k \frac{\partial U_k}{\partial x_k}^\top \delta x_k - \alpha h \sum_k \frac{\partial U_{k+1}}{\partial x_{k+1}}^\top \delta x_{k+1} \\
&\quad + (1-\alpha) h \sum_k \mathrm{Trace}\left(\eta_k R_k^\top \frac{\partial U_k}{\partial R_k} \right) + \alpha h \sum_k \mathrm{Trace}\left(\eta_{k+1} R_{k+1}^\top \frac{\partial U_{k+1}}{\partial R_{k+1}} \right) \\
&\quad + \frac{1}{2} \sum_k \mathrm{Trace}\left(\eta_k S(f^{R-}_{d_k}) \right) + \frac{1}{2} \sum_k \mathrm{Trace}\left(\eta_{k+1} S(f^{R+}_{d_k}) \right) \\
&\quad + \sum_k R_k f^{x-}_{d_k} \cdot \delta x_k + \sum_k R_{k+1} f^{x+}_{d_k} \cdot \delta x_{k+1}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
0 &= \frac{1}{h} \sum_k \mathrm{Trace}\left((\eta_k Z_k - Z_k \eta_{k+1}) J_d \right) + \frac{m}{h} \sum_k (x_{k+1} - x_k)^\top R_k^\top R_k (\delta x_{k+1} - \delta x_k) \\
&\quad - (1-\alpha) h \sum_k \frac{\partial U_k}{\partial x_k}^\top \delta x_k - \alpha h \sum_k \frac{\partial U_{k+1}}{\partial x_{k+1}}^\top \delta x_{k+1} \\
&\quad + (1-\alpha) h \sum_k \mathrm{Trace}\left(\eta_k R_k^\top \frac{\partial U_k}{\partial R_k} \right) + \alpha h \sum_k \mathrm{Trace}\left(\eta_{k+1} R_{k+1}^\top \frac{\partial U_{k+1}}{\partial R_{k+1}} \right) \\
&\quad - \frac{1}{2} \sum_k \mathrm{Trace}\left(\eta_k S(f^{R-}_{d_k}) \right) - \frac{1}{2} \sum_k \mathrm{Trace}\left(\eta_{k+1} S(f^{R+}_{d_k}) \right) \\
&\quad + \sum_k R_k f^{x-}_{d_k} \cdot \delta x_k + \sum_k R_{k+1} f^{x+}_{d_k} \cdot \delta x_{k+1}
\end{aligned}$$

Then, since the variations δx_k and η_k vanish at the endpoints $k = 0$ and $k = N$, we can shift the summation indices appropriately, and gather similar terms to obtain

$$\begin{aligned}
0 = &\sum_k \mathrm{Trace}\left(\eta_k \left[\frac{1}{h} (Z_k J_d - J_d Z_{k-1}) + (1-\alpha) h R_k^\top \frac{\partial U_k}{\partial R_k} + \alpha h R_k^\top \frac{\partial U_k}{\partial R_k} - \frac{1}{2} S\left(f^{R-}_{d_k} + f^{R+}_{d_{k-1}} \right) \right] \right) \\
&+ \sum_k \delta x_k^\top \left(\frac{m}{h} (x_{k+1} - 2x_k + x_{k-1}) - (1-\alpha) h \frac{\partial U_k}{\partial x_k} - \alpha h \frac{\partial U_k}{\partial x_k} + R_k (f^{x-}_{d_k} + f^{x+}_{d_{k-1}}) \right).
\end{aligned}$$

This simplifies to

$$0 = \sum_k \mathrm{Trace}\left(\eta_k\left[\frac{1}{h}(Z_k J_d - J_d Z_{k-1}) + hR_k^\top \frac{\partial U_k}{\partial R_k} - \frac{1}{2}S\left(f_{d_k}^{R-} + f_{d_{k-1}}^{R+}\right)\right]\right) + \sum_k \delta x_k^\top \left(\frac{m}{h}(x_{k+1} - 2x_k + x_{k-1}) - h\frac{\partial U_k}{\partial x_k} + R_k(f_{d_k}^{x-} + f_{d_{k-1}}^{x+})\right).$$

This equation must hold for any possible variations $\delta x_k \in \mathbb{R}^3$ and $\eta_k \in \mathfrak{so}(3)$ that vanish at the endpoints, so each term in the summations must be zero. Thus, the following equations must hold for each k:

$$\frac{m}{h}(x_{k+1} - 2x_k + x_{k-1}) - h\frac{\partial U_k}{\partial x_k} + R_k(f_{d_k}^{x-} + f_{d_{k-1}}^{x+}) = 0, \tag{A.21}$$

$$\mathrm{Trace}\left(\eta_k\left[\frac{1}{h}(Z_k J_d - J_d Z_{k-1}) + hR_k^\top \frac{\partial U_k}{\partial R_k} - \frac{1}{2}S\left(f_{d_k}^{R-} + f_{d_{k-1}}^{R+}\right)\right]\right) = 0. \tag{A.22}$$

Since $\eta_k \in \mathfrak{so}(3)$, the quantity

$$\frac{1}{h}(Z_k J_d - J_d Z_{k-1}^\top) + hR_k^\top \frac{\partial U_k}{\partial R_k} + \frac{1}{2}S\left(f_{d_k}^{R-} + f_{d_{k-1}}^{R+}\right)$$

must be symmetric, so since $S\left(f_{d_k}^{R-} + f_{d_{k-1}}^{R+}\right) \in \mathfrak{so}(3)$, we get the equations

$$x_{k+1} = 2x_k - x_{k-1} + \frac{h^2}{m}\frac{\partial U_k}{\partial x_k} + \frac{h}{m}R_k\left(f_{d_k}^{x-} + f_{d_{k-1}}^{x+}\right), \tag{A.23}$$

$$\frac{1}{h}\left[Z_k J_d - J_d Z_{k-1} - J_d Z_k^\top + Z_{k-1}^\top J_d\right] - h\left[\frac{\partial U_k}{\partial R_k}^\top R_k - R_k^\top \frac{\partial U_k}{\partial R_k}\right] - S\left(f_{d_k}^{R-} + f_{d_{k-1}}^{R+}\right) = 0. \tag{A.24}$$

Thus, defining the quantity ξ_k via

$$S(\xi_k) = \frac{\partial U_k}{\partial R_k}^\top R_k - R_k^\top \frac{\partial U_k}{\partial R_k}, \tag{A.25}$$

we obtain the forced discrete Euler–Lagrange equations

$$\boxed{x_{k+1} = 2x_k - x_{k-1} + \frac{h^2}{m}\frac{\partial U_k}{\partial x_k} + \frac{h}{m}R_k\left(f_{d_k}^{x-} + f_{d_{k-1}}^{x+}\right),} \tag{A.26}$$

$$\boxed{h^2 S(\xi_k) + hS\left(f_{d_k}^{R-} + f_{d_{k-1}}^{R+}\right) + J_d Z_{k-1} - Z_{k-1}^\top J_d = Z_k J_d - J_d Z_k^\top,} \tag{A.27}$$

and

$$\boxed{R_{k+1} = R_k Z_k.} \tag{A.28}$$

Using the discrete Legendre transforms

$$S(\pi_k) = \frac{1}{h}(Z_k J_d - J_d Z_k^\top) - (1-\alpha)hS(\xi_k) - S(f_{d_k}^{R-}), \tag{A.29}$$

$$\gamma_k = \frac{m}{h}(x_{k+1} - x_k) + (1-\alpha)h\frac{\partial U_k}{\partial x_k} - R_k f_{d_k}^{x-}, \tag{A.30}$$

we can rewrite equations (A.26) and (A.27) as

$$S(\pi_{k+1}) = \frac{1}{h}(J_d Z_k - Z_k^\top J_d) + \alpha hS(\xi_{k+1}) + S(f_{d_k}^{R+}), \tag{A.31}$$

$$\gamma_{k+1} = \frac{m}{h}(x_{k+1} - x_k) - \alpha h\frac{\partial U_{k+1}}{\partial x_{k+1}} + R_{k+1} f_{d_k}^{x+}. \tag{A.32}$$

Overall, we obtain the following implicit discrete equations of motion:

$$S(\pi_k) = \frac{1}{h}(Z_k J_d - J_d Z_k^\top) - (1-\alpha)hS(\xi_k) - S(f_{d_k}^{R-}), \tag{A.33}$$

$$\gamma_k = \frac{m}{h}(x_{k+1} - x_k) + (1-\alpha)h\frac{\partial U_k}{\partial x_k} - R_k f_{d_k}^{x-}, \tag{A.34}$$

$$R_{k+1} = R_k Z_k, \tag{A.35}$$

$$S(\pi_{k+1}) = \frac{1}{h}(J_d Z_k - Z_k^\top J_d) + \alpha hS(\xi_{k+1}) + S(f_{d_k}^{R+}), \tag{A.36}$$

$$\gamma_{k+1} = \frac{m}{h}(x_{k+1} - x_k) - \alpha h\frac{\partial U_{k+1}}{\partial x_{k+1}} + R_{k+1} f_{d_k}^{x+}. \tag{A.37}$$

Equations (A.33) and (A.34) can be rewritten as

$$hS(\pi_k) + hS(f_{d_k}^{R-}) + (1-\alpha)h^2 S(\xi_k) = Z_k J_d - J_d Z_k^\top, \tag{A.38}$$

$$x_{k+1} = x_k + \frac{h}{m}\gamma_k - (1-\alpha)\frac{h^2}{m}\frac{\partial U_k}{\partial x_k} - \frac{h}{m}R_k f_{d_k}^{x-}. \tag{A.39}$$

Equation (A.36) can be rewritten using equation (A.33) as

$$S(\pi_{k+1}) = Z_k^\top S(\pi_k) Z_k + (1-\alpha)hZ_k^\top S(\xi_k) Z_k + \alpha hS(\xi_{k+1}) + Z_k^\top S(f_{d_k}^{R-}) Z_k + S(f_{d_k}^{R+}). \tag{A.40}$$

Since $Z^\top S(\zeta) Z = S(Z^\top \zeta)$ for any $Z \in \mathrm{SO}(3)$ and $\zeta \in \mathfrak{so}(3)$, we get

$$S(\pi_{k+1}) = S(Z_k^\top \pi_k) + (1-\alpha)hS(Z_k^\top \xi_k) + \alpha hS(\xi_{k+1}) + S(Z_k^\top f_{d_k}^{R-}) + S(f_{d_k}^{R+}). \tag{A.41}$$

Therefore,

$$\pi_{k+1} = Z_k^\top \pi_k + (1-\alpha) h Z_k^\top \xi_k + \alpha h \xi_{k+1} + Z_k^\top f_{d_k}^{R-} + f_{d_k}^{R+}. \tag{A.42}$$

Finally, equation (A.37) can be rewritten using equation (A.34) as

$$\gamma_{k+1} = \gamma_k - (1-\alpha) h \frac{\partial U_k}{\partial x_k} - \alpha h \frac{\partial U_{k+1}}{\partial x_{k+1}} + R_k f_{d_k}^{x-} + R_{k+1} f_{d_k}^{x+}. \tag{A.43}$$

Overall, this gives the forced variational integrator (6.14)-(6.18):

$$\begin{aligned}
&hS(\pi_k) + hS(f_{d_k}^{R-}) + (1-\alpha)h^2 S(\xi_k) = Z_k J_d - J_d Z_k^\top, \\
&R_{k+1} = R_k Z_k, \\
&\pi_{k+1} = Z_k^\top \pi_k + (1-\alpha) h Z_k^\top \xi_k + \alpha h \xi_{k+1} + Z_k^\top f_{d_k}^{R-} + f_{d_k}^{R+}, \\
&x_{k+1} = x_k + \frac{h}{m}\gamma_k - (1-\alpha)\frac{h^2}{m}\frac{\partial U_k}{\partial x_k} - \frac{h}{m} R_k f_{d_k}^{x-}, \\
&\gamma_{k+1} = \gamma_k - (1-\alpha) h \frac{\partial U_k}{\partial x_k} - \alpha h \frac{\partial U_{k+1}}{\partial x_{k+1}} + R_k f_{d_k}^{x-} + R_{k+1} f_{d_k}^{x+}.
\end{aligned}$$

Alternatively, we can start from the forced discrete Euler–Lagrange equations for a discrete Lagrangian $L_d(g_k, z_k)$ on a Lie group G,

$$g_{k+1} = g_k \star z_k, \tag{A.44}$$

$$\mathrm{T}_e^* \mathrm{L}_{z_{k-1}} D_2 L_{d_{k-1}} - \mathrm{Ad}_{z_k^{-1}}^* \left(\mathrm{T}_e^* \mathrm{L}_{z_k} D_2 L_{d_k} \right) + \mathrm{T}_e^* \mathrm{L}_{g_k} D_1 L_{d_k} + f_{d_k}^- + f_{d_{k-1}}^+ = 0, \tag{A.45}$$

where $L_{d_k} = L_d(g_k, z_k)$ and $f_{d_k}^\pm = f_d^\pm(g_k, g_{k+1}, u_k)$.

On SE(3), with $g_k = (x_k, R_k) \in \mathrm{SE}(3)$ and $z_k = (y_k, Z_k) \in \mathrm{SE}(3)$, the discrete kinematics equations $g_{k+1} = g_k \star z_k$ are given by

$$R_{k+1} = R_k Z_k \qquad \text{and} \qquad x_{k+1} = x_k + R_k y_k, \tag{A.46}$$

so that $\{(x_k, R_k)\}_k$ remains on SE(3). Using the kinematics equation $\dot{R} = RS(\omega)$, the matrix $S(\omega_k)$ can be approximated via

$$S(\omega_k) \;=\; R_k^\top \dot{R}_k \;\approx\; R_k^\top \frac{R_{k+1} - R_k}{h} \;=\; \frac{1}{h}(Z_k - \mathbb{I}_3). \tag{A.47}$$

With the discrete Lagrangian

$$L_d(x_k, R_k, y_k, Z_k) = \frac{m}{2h} y_k^\top y_k + \frac{1}{h}\mathrm{tr}\left([\mathbb{I}_3 - Z_k]J_d\right) - (1-\alpha)hU(x_k, R_k) - \alpha h U(x_k + R_k y_k, R_k Z_k), \tag{A.48}$$

it can be shown by proceeding as in [Lee, 2008] that the Lie group forced discrete Euler–Lagrange equations become

$$\frac{1}{h}(J_d Z_{k-1} - Z_{k-1}^\top J_d) - \frac{1}{h}(Z_k J_d - J_d Z_k^\top) + hS(\xi_k) + S(f_{d_k}^{R-}) + S(f_{d_{k-1}}^{R+}) = 0, \tag{A.49}$$

$$\frac{m}{h} R_k^\top (x_k - x_{k-1}) - \frac{m}{h} R_k^\top (x_{k+1} - x_k) - hR_k^\top \frac{\partial U_k}{\partial x_k} + f_{d_k}^{x-} + f_{d_{k-1}}^{x+} = 0, \tag{A.50}$$

$$R_{k+1} = R_k Z_k, \tag{A.51}$$

where $f_{d_k}^{x\pm}$ and $f_{d_k}^{R\pm}$ denote the x and R components of the discrete forces $f_{d_k}^{\pm}$.

This can be simplified into the forced discrete Euler–Lagrange equations

$$h^2 S(\xi_k) + hS(f_{d_k}^{R-}) + hS(f_{d_{k-1}}^{R+}) + (J_d Z_{k-1} - Z_{k-1}^\top J_d) = Z_k J_d - J_d Z_k^\top, \tag{A.52}$$

$$x_{k+1} = 2x_k - x_{k-1} - \frac{h^2}{m}\frac{\partial U_k}{\partial x_k} + \frac{h}{m} R_k (f_{d_k}^{x-} - f_{d_{k-1}}^{x+}), \tag{A.53}$$

$$R_{k+1} = R_k Z_k. \tag{A.54}$$

Using the discrete Legendre transforms

$$S(\pi_k) = \frac{1}{h}(Z_k J_d - J_d Z_k^\top) - (1-\alpha)hS(\xi_k) - S(f_{d_k}^{R-}), \tag{A.55}$$

$$\nu_k = \frac{m}{h} R_k^\top (x_{k+1} - x_k) + (1-\alpha)hR_k^\top \frac{\partial U_k}{\partial x_k} - f_{d_k}^{x-}, \tag{A.56}$$

with $\gamma = R\nu$, we can recover as before the forced variational integrator (6.14)-(6.18):

$$hS(\pi_k) + (1-\alpha)h^2 S(\xi_k) = Z_k J_d - J_d Z_k^\top - hS(f_{d_k}^{R-}), \tag{A.57}$$

$$R_{k+1} = R_k Z_k, \tag{A.58}$$

$$\pi_{k+1} = Z_k^\top \pi_k + (1-\alpha)hZ_k^\top \xi_k + \alpha h \xi_{k+1} + Z_k^\top f_{d_k}^{R-} + f_{d_k}^{R+}, \tag{A.59}$$

$$x_{k+1} = x_k + \frac{h}{m}\gamma_k - (1-\alpha)\frac{h^2}{m}\frac{\partial U_k}{\partial x_k} - \frac{h}{m} R_k f_{d_k}^{x-}, \tag{A.60}$$

$$\gamma_{k+1} = \gamma_k - (1-\alpha)h\frac{\partial U_k}{\partial x_k} - \alpha h \frac{\partial U_{k+1}}{\partial x_{k+1}} + R_k f_{d_k}^{x-} + R_{k+1} f_{d_k}^{x+}. \tag{A.61}$$

A.9 Transforming the equation $S(a) = ZJ_d - J_dZ^\top$

Plugging the Cayley transform

$$Z = \mathrm{Cay}(\mathfrak{z}) \equiv (\mathbb{I}_3 + S(\mathfrak{z}))(\mathbb{I}_3 - S(\mathfrak{z}))^{-1}, \tag{A.62}$$

into the equation

$$S(a) = ZJ_d - J_dZ^\top, \tag{A.63}$$

and using the fact that $(\mathbb{I}_3 \pm S(\mathfrak{z}))^\top = (\mathbb{I}_3 \mp S(\mathfrak{z}))$ gives

$$S(a) = (\mathbb{I}_3 + S(\mathfrak{z}))(\mathbb{I}_3 - S(\mathfrak{z}))^{-1}J_d - J_d(\mathbb{I}_3 + S(\mathfrak{z}))^{-1}(\mathbb{I}_3 - S(\mathfrak{z})). \tag{A.64}$$

Now, $(\mathbb{I}_3 \pm S(\mathfrak{z}))$ and $(\mathbb{I}_3 \mp S(\mathfrak{z}))^{-1}$ commute, so we can rewrite the previous equation as

$$S(a) = (\mathbb{I}_3 - S(\mathfrak{z}))^{-1}(\mathbb{I}_3 + S(\mathfrak{z}))J_d - J_d(\mathbb{I}_3 - S(\mathfrak{z}))(\mathbb{I}_3 + S(\mathfrak{z}))^{-1}. \tag{A.65}$$

Multiplying both sides of equation (A.65) on the left by $(\mathbb{I}_3 - S(\mathfrak{z}))$ and on the right by $(\mathbb{I}_3 + S(\mathfrak{z}))$ gives

$$(\mathbb{I}_3 - S(\mathfrak{z}))S(a)(\mathbb{I}_3 + S(\mathfrak{z})) = (\mathbb{I}_3 + S(\mathfrak{z}))J_d(\mathbb{I}_3 + S(\mathfrak{z})) - (\mathbb{I}_3 - S(\mathfrak{z}))J_d(\mathbb{I}_3 - S(\mathfrak{z})), \tag{A.66}$$

which can be simplified into

$$S(a) - S(\mathfrak{z})S(a) + S(a)S(\mathfrak{z}) - S(\mathfrak{z})S(a)S(\mathfrak{z}) = 2S(\mathfrak{z})J_d + 2J_dS(\mathfrak{z}). \tag{A.67}$$

Using $S(\mathfrak{z})J_d + J_dS(\mathfrak{z}) = S(J\mathfrak{z})$ and the general formulas

$$-S(y)S(x) + S(x)S(y) = S(S(x)y), \qquad S(x)S(y)S(x) = -(y^\top x)S(x), \tag{A.68}$$

we can simplify equation (A.67) into

$$S(a) + S(S(a)\mathfrak{z}) + (a^\top\mathfrak{z})S(\mathfrak{z}) = 2S(J\mathfrak{z}). \tag{A.69}$$

This can be rewritten in the desired vector form

$$a + a \times \mathfrak{z} + (a^\top\mathfrak{z})\mathfrak{z} - 2J\mathfrak{z} = 0. \tag{A.70}$$

➢ The Appendix contains original material from

① "Adaptive Hamiltonian Variational Integrators and Applications to Symplectic Accelerated Optimization" by V. Duruisseaux, J. Schmitt, and M. Leok. *SIAM Journal on Scientific Computing,* Vol.43, No.4, A2949-A2980, 2021

② "A Variational Formulation of Accelerated Optimization on Riemannian Manifolds" by V. Duruisseaux and M. Leok. *SIAM Journal on Mathematics of Data Science*, Vol.4, No.2, pages 649-674, 2022

③ "Accelerated Optimization on Riemannian Manifolds via Discrete Constrained Variational Integrators" by V. Duruisseaux and M. Leok. *Journal of Nonlinear Science*, Vol.32, No.42, 2022

④ "Time-adaptive Lagrangian Variational Integrators for Accelerated Optimization on Manifolds" by V. Duruisseaux and M. Leok. *Journal of Geometric Mechanics*, Vol.15, Issue 1, pages 224-255, 2023.

⑤ "Lie Group Forced Variational Integrator Networks for Learning and Control of Robot Systems" by V. Duruisseaux, T. Duong, M. Leok, and N. Atanasov. *Proceedings of the 5th Learning for Dynamics and Control Conference (L4DC),* PMLR 211:731-744, 2023

The dissertation author was the primary investigator and author of these papers.

B Symbols and Abbreviations

We list here symbols and abbreviations used in this dissertation, together with the section in which they are first introduced:

$\varnothing$	Empty set	
$\mathbb{R}$	Real line	
$u^\top$	Transpose of u	
$\mathrm{Trace}(\cdot)$	Trace operator for matrices	
∇f	Gradient of f	
$\mathrm{Id}_{\mathcal{Q}}$	Identity map on $\mathcal{Q}$	
$\mathbb{I}_n$	$(n \times n)$ Identity matrix	
$\mathbf{1}$	Vector of ones	
ODE/PDE	Ordinary/Partial Differential Equation	
$T_m M$	Tangent space to manifold M at point m	Section 1.1.1
$T_m^* M$	Cotangent space to manifold M at point m	Section 1.1.1
TM	Tangent bundle of manifold M	Section 1.1.1
T^*M	Cotangent bundle of manifold M	Section 1.1.1
$\mathcal{X}(M)$	Set of all vector fields on manifold M	Section 1.1.1
$\mathrm{T}_m f$	Tangent map (or differential) of f	Section 1.1.1
Alt	Alternating operator	Section 1.1.1
$\alpha \otimes \beta$	Tensor product of the differential forms α and β	Section 1.1.1
$\alpha \wedge \beta$	Wedge product of the differential forms α and β	Section 1.1.1
$\mathbf{d}\alpha$	Exterior derivative of the differential form α	Section 1.1.1
$\iota_X \alpha$	Interior product with vector field X of the form α	Section 1.1.1
$\psi^* f$	Pull-back of the function f by the map ψ	Section 1.1.1
$\psi^* X$	Pull-back of the vector field X by the map ψ	Section 1.1.1
$\psi^* \alpha$	Pull-back of the form α by the map ψ	Section 1.1.1
$\psi_* f$	Push-forward of the function f by the map ψ	Section 1.1.1
$\psi_* X$	Push-forward of the vector field X by the map ψ	Section 1.1.1
$\psi_* \alpha$	Push-forward of the form α by the map ψ	Section 1.1.1
$\mathcal{L}_X \alpha$	Lie derivative of form α along the vector field X	Section 1.1.1

$g(\cdot,\cdot) = \langle \cdot, \cdot \rangle$	Riemannian metric	Section 1.1.2
$g^\flat(\cdot)(\cdot)$	Musical isomorphism	Section 1.1.2
$g^\sharp(\cdot)(\cdot)$	Inverse musical isomorphism	Section 1.1.2
$g^*(\cdot,\cdot) = \langle\!\langle \cdot, \cdot \rangle\!\rangle$	Riemannian fiber metric	Section 1.1.2
$\mathrm{grad} f$	Riemannian gradient of f	Section 1.1.2
$\Gamma(\gamma)$	Parallel transport along the geodesic γ	Section 1.1.2
Exp_q	Riemannian Exponential map at point q	Section 1.1.2
Log_q	Riemannian Logarithm map at point q	Section 1.1.2
$\nabla_X Y$	Covariant derivative of Y along X	Section 1.1.2
$[X, Y]$	Lie bracket of X and Y	Section 1.1.2
$\mathbb{S}^{n-1}$	Unit sphere	Section 1.1.4
$\mathrm{St}(m, n)$	Stiefel manifold	Section 1.1.4
L_g	Left translation map on Lie groups	Section 1.2.1
R_g	Left translation map on Lie groups	Section 1.2.1
I_g	Inner automorphism on Lie groups	Section 1.2.1
$\mathfrak{g}$	Lie algebra	Section 1.2.1
Ad_g	Adjoint operator	Section 1.2.1
Ad_g^*	Coadjoint operator	Section 1.2.1
ad_ξ	ad operator	Section 1.2.1
ad_η^*	co-ad operator	Section 1.2.1
$\mathrm{GL}(n, \mathbb{R})$	Real General Linear Group	Section 1.2.2
$\mathrm{O}(n)$	Orthogonal Group	Section 1.2.2
$\mathrm{SO}(n)$	Special Orthogonal Group	Section 1.2.2
$\mathfrak{so}(n)$	Lie algebra of $\mathrm{SO}(n)$	Section 1.2.2
$(\cdot)^\wedge$ or $S(\cdot)$	Hat map on $\mathbb{R}^3$	Section 1.2.2
$(\cdot)^\vee$	Vee map on $\mathfrak{so}(3)$	Section 1.2.2
$\mathrm{SE}(n)$	Special Euclidean Group	Section 1.2.2
$\mathfrak{se}(n)$	Lie algebra of $\mathrm{SE}(n)$	Section 1.2.2
$\mathrm{U}(1)$	Circle Group	Section 1.2.2
$\delta(\cdot)$	Infinitessimal variation	Section 2.1
$\mathfrak{S}$	Action functional	Section 2.1
$\mathbb{F}, \bar{\mathbb{F}}, \mathbb{F}^\pm$	Legendre transform	Section 2.1
$\mathbb{J}$	Symplectic matrix	Section 2.1
AƎƎ	Backward error analysis	Section 2.3
$\mathfrak{S}_d, \mathfrak{S}_d^\pm$	Discrete action functional	Section 2.4
L_d	Discrete Lagrangian	Section 2.4
L_d^E	Exact discrete Lagrangian	Section 2.4
$\tilde{F}_{L_d}$	Discrete Hamiltonian map	Section 2.4
$H_d^\pm$	Discrete Hamiltonian	Section 2.4
$H_d^{\pm,E}$	Exact discrete Hamiltonian	Section 2.4
$\tilde{F}_{H_d^\pm}$	Discrete Hamilton's map	Section 2.4
LTVI	Lagrangian Taylor Variational Integrator	Section 2.4.2
HTVI	Hamiltonian Taylor Variational Integrator	Section 2.4.2

π_Q	Projection map on Q	Section 2.4.2
π_{T^*Q}	Projection map on T^*Q	Section 2.4.2
$f_{d_k}^{\pm}$	Discrete control forces	Section 6.3.2
$\mathcal{C}$	Holonomic constraint function	Section 2.6.1
Λ	Space of Lagrange multipliers	Section 2.6.1
$\mathcal{S}$	Generating function	Section 2.6.1
$g(q,p)$, $g(q,t,p)$	Monitor function	Section 2.7
$\bar{q}$	Extended position variable	Section 2.7.1
$\mathfrak{q}$	Additional position variable	Section 2.7.1
HTVI4	A 4th-Order HTVI	Section 2.7.1
SV	Störmer–Verlet integrator	Section 2.7.1
NAG	Nesterov's Accelerated Gradient method	Section 3.1
$L_{\alpha,\beta,\gamma}$	General Bregman Lagrangian	Section 3.2
$H_{\alpha,\beta,\gamma}$	General Bregman Hamiltonian	Section 3.2
L_p	Polynomial p-Bregman Lagrangian	Section 3.2
H_p	Polynomial p-Bregman Hamiltonian	Section 3.2
$\bar{H}_p$	Direct approach Poincaré transformed p-Hamiltonian	Section 3.3.2
$\bar{H}_{p\to\mathring{p}}$	Adaptive approach Poincaré transformed $p\to\mathring{p}$-Hamiltonian	Section 3.3.3
Direct Splitting	Numerical method based on splitting	Section 3.3.4
Adaptive Splitting	Numerical method based on splitting	Section 3.3.4
CloningStrang	Cloning method with Strang Splitting	Section 3.3.4
CloningY4	Cloning method with Yoshida 4 Splitting	Section 3.3.4
CloningY6	Cloning method with Yoshida 6 Splitting	Section 3.3.4
`ode23`, `ode45`	MATLAB Solvers	Section 3.3.4
ADAM	Popular optimization method	Section 3.3.4
AdaGrad	Popular optimization method	Section 3.3.4
RMSProp	Popular optimization method	Section 3.3.4
TRUST	Trust region steepest descent method	Section 3.3.4
RK	Runge–Kutta method	Section 3.3.4
$\|\cdot\|_F$	Frobenius norm	Section 4.1.2
$\mathscr{L}_{\alpha,\beta,\gamma}$	General Riemannian Bregman Lagrangian	Section 4.2.1
$\mathscr{H}_{\alpha,\beta,\gamma}$	General Riemannian Bregman Hamiltonian	Section 4.2.1
$\mathscr{L}_p$	Riemannian p-Bregman Lagrangian	Section 4.2.1
$\mathscr{H}_p$	Riemannian p-Bregman Hamiltonian	Section 4.2.1
SC	Strongly convex	Section 4.2.2
$\mathscr{L}^{SC}$	Riemannian strongly-convex Bregman Lagrangian	Section 4.2.2
$\mathscr{H}^{SC}$	Riemannian strongly-convex Bregman Hamiltonian	Section 4.2.2

SIRNAG	Semi-Implicit Riemannian NAG	Section 4.3
RGD	Riemannian Gradient Descent	Section 4.3
$\bar{\mathscr{H}}_p$	Direct approach Riemannian Poincaré transformed p-Hamiltonian	Section 4.4
$\bar{\mathscr{H}}_{p\to\mathring{p}}$	Adaptive approach Riemannian Poincaré transformed $p \to \mathring{p}$-Hamiltonian	Section 4.4
EL V1	Euler–Lagrange discretization Version 1	Section 4.5.2
EL V2	Euler–Lagrange discretization Version 2	Section 4.5.2
HTVI AD	Adaptive HTVI	Section 4.5.2
Adaptive LLGVI	Adaptive Lagrangian Lie Group Variational Integrator	Section 4.6
Implicit LGVI	Implicit Lie Group Variational Integrator	Section 4.6
L^η	Exponential η-Bregman Lagrangian	Section 5.2.3
H^η	Exponential η-Bregman Hamiltonian	Section 5.2.3
$\bar{H}_{p\to\mathring{p}}$	Polynomial-p to Polynomial-$\mathring{p}$ Hamiltonian	Section 5.2.5
$\bar{H}^{\eta\to\mathring{\eta}}$	Exponential-η to Exponential-$\mathring{\eta}$ Hamiltonian	Section 5.2.5
$\bar{H}^{\eta}_{\to p}$	Exponential-η to Polynomial-p Hamiltonian	Section 5.2.5
$\bar{H}_{p}^{\to\eta}$	Polynomial-p to Exponential-η Hamiltonian	Section 5.2.5
SLC	Symmetric Leapfrog Composition	Section 5.3.1
PolyHTVI	HTVI for Polynomial Bregman Dynamics	Section 5.3.1
PolyLTVI	LTVI for Polynomial Bregman Dynamics	Section 5.3.1
PolySLC	SLC for Polynomial Bregman Dynamics	Section 5.3.1
PolySV	SV for Polynomial Bregman Dynamics	Section 5.3.1
ExpoHTVI	HTVI for Exponential Bregman Dynamics	Section 5.3.1
ExpoLTVI	LTVI for Exponential Bregman Dynamics	Section 5.3.1
ExpoSLC	SLC for Exponential Bregman Dynamics	Section 5.3.1
ExpoSV	SV for Exponential Bregman Dynamics	Section 5.3.1
ExpoToPolyHTVI	HTVI for Expo-to-Poly Dynamics	Section 5.3.1
ExpoToPolyLTVI	LTVI for Expo-to-Poly Bregman Dynamics	Section 5.3.1
ExpoToPolySLC	SLC for Expo-to-Poly Dynamics	Section 5.3.1
ExpoToPolySV	SV for Expo-to-Poly Dynamics	Section 5.3.1
PolyToExpoHTVI	HTVI for Poly-to-Expo Dynamics	Section 5.3.1
PolyToExpoLTVI	LTVI for Poly-to-Expo Dynamics	Section 5.3.1
PolyToExpoSLC	SLC for Poly-to-Expo Dynamics	Section 5.3.1
PolyToExpoSV	SV for Poly-to-Expo Dynamics	Section 5.3.1
PolySLC-R	PolySLC with Momentum Restarting	Section 5.6
ExpoSLC-R	ExpoSLC with Momentum Restarting	Section 5.6
PolySLC-RTL	PolySLC with Momentum Restarting and Temporal Looping	Section 5.8
ExpoSLC-RTL	ExpoSLC with Momentum Restarting and Temporal Looping	Section 5.8

pBrAVO	Polynomial Bregman Accelerated Variational Optimizer	Section 5.9
eBrAVO	Exponential Bregman Accelerated Variational Optimizer	Section 5.9
SGD	Standard Gradient Descent	Section 5.9
LSTM	Long Short-Term Memory	Section 5.9
MSE	Mean Squared Error	Section 5.9
(F)VINs	(Forced) Variational Integrator Networks	Section 6.1
MPC	Model Predictive Control	Section 6.1
LieFVIN	Lie group Forced Variational Integrator Networks	Section 6.3
LieVIN	Lie group Variational Integrator Networks	Section 6.3.3
$H[V, \eta]$	Hénon-like map	Section 7.2
$H_\varepsilon[V, \eta]$	Near-identity Hénon-like map	Section 7.2
$L[V, \eta]$	Hénon layer	Section 7.2
$\mathscr{H}[\boldsymbol{V}[\boldsymbol{W}], \boldsymbol{\eta}]$	Hénon Network	Section 7.2
$\mathscr{H}_\varepsilon[\boldsymbol{V}[\boldsymbol{W}], \boldsymbol{\eta}]$	Near-identity Hénon Network	Section 7.2

C Data Availability Statement

Simple implementations of some accelerated optimization algorithms in MATLAB, Julia and Python, and more sophisticated Python code implementations which allow the optimizers to be called conveniently within the TensorFlow and PyTorch frameworks can be found at

https://github.com/vduruiss/AccOpt_via_GNI

The Python codes used for the numerical experiments presented in Chapter 6 for the Lie Group Forced Variational Integrator Networks can be found at

https://github.com/thaipduong/LieFVIN

A simplified implementation of the Python codes used to generate some of the numerical results presented in Chapter 7 is published [Duruisseaux et al., 2023b] and available at

https://github.com/vduruiss/SymplecticGyroceptron

Bibliography

R. Abraham and J. E. Marsden. *Foundations of mechanics*. Benjamin/Cummings Publishing Company, second edition, 1978. https://authors.library.caltech.edu/25029/1/FoM2.pdf.

R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications*, volume 75 of *Applied Mathematical Sciences*. Springer, New York, second edition, 1988. https://doi.org/10.1007/978-1-4612-1029-0.

P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. ISBN 978-0-691-13298-3. https://press.princeton.edu/absil.

J. A. Acosta, M. I. Sanchez, and A. Ollero. Robust control of underactuated aerial manipulators via IDA-PBC. In *IEEE Conference on Decision and Control (CDC)*, 2014. https://doi.org/10.1109/CDC.2014.7039459.

K. Ahn and S. Sra. From Nesterov's estimate sequence to Riemannian acceleration. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 84–118. PMLR, 09–12 Jul 2020. https://proceedings.mlr.press/v125/ahn20a.html.

C. D. Alecsa and S. C. László. Tikhonov regularization of a perturbed heavy ball system with vanishing damping. *SIAM Journal on Optimization*, 31(4):2921–2954, 2021. https://doi.org/10.1137/20M1382027.

F. Alimisis, A. Orvieto, G. Bécigneul, and A. Lucchi. Practical accelerated optimization on Riemannian manifolds, 2020a. https://arxiv.org/abs/2002.04144v1.

F. Alimisis, A. Orvieto, G. Bécigneul, and A. Lucchi. A continuous-time perspective for modeling acceleration in Riemannian optimization. In *Proceedings of the 23rd International AISTATS Conference*, volume 108 of *PMLR*, pages 1297–1307, 2020b. http://proceedings.mlr.press/v108/alimisis20a.

F. Alvarez, H. Attouch, J. Bolte, and P. Redont. A second-order gradient-like dissipative dynamical system with Hessian-driven damping: Application to optimization and

mechanics. *Journal de Mathématiques Pures et Appliquées*, 81(8):747–779, 2002. ISSN 0021-7824. https://doi.org/10.1016/S0021-7824(01)01253-3.

B. Amos, I. Jimenez, J. Sacks, B. Boots, and J. Z. Kolter. Differentiable MPC for End-to-end Planning and Control. In *Advances in Neural Information Processing Systems*, 2018. https://papers.nips.cc/paper/2018/hash/ba6d843eb4251a4526ce65d1807a9309-Abstract.html.

V. I. Arnol′d. *Mathematical methods of classical mechanics*, volume 60 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1989. Translated from the Russian by K. Vogtmann and A. Weinstein. https://doi.org/10.1007/978-1-4757-1693-1.

H. Attouch and Z. Chbani. Combining fast inertial dynamics for convex optimization with Tikhonov regularization. *Journal of Mathematical Analysis and Applications*, 457(2):1065–1094, 2018. ISSN 0022-247X. Special Issue on Convex Analysis and Optimization: New Trends in Theory and Applications. https://doi.org/10.1016/j.jmaa.2016.12.017.

H. Attouch and M. Czarnecki. Asymptotic behavior of gradient-like dynamical systems involving inertia and multiscale aspects. *Journal of Differential Equations*, 262(3):2745–2770, 2017. ISSN 0022-0396. https://doi.org/10.1016/j.jde.2016.11.009.

H. Attouch, Z. Chbani, J. Fadili, and H. Riahi. First-order optimization algorithms via inertial systems with Hessian driven damping. *Mathematical Programming*, November 2020. https://doi.org/10.1007/s10107-020-01591-1.

H. Attouch, Z. Chbani, J. M. Fadili, and H. Riahi. Convergence of iterates for first-order optimization algorithms with inertia and Hessian driven damping. *Optimization*, 2021. https://doi.org/10.1080/02331934.2021.2009828.

H. Attouch, A. Balhag, Z. Chbani, and H. Riahi. Fast convex optimization via inertial dynamics combining viscous and Hessian-driven damping with time rescaling. *Evolution Equations and Control Theory*, 11(2):487–514, 2022. https://doi.org/10.3934/eect.2021010.

P. Attri, Y. Sharma, K. Takach, and F. Shah. Timeseries forecasting for weather prediction. *Keras Tutorial*, 2020. https://keras.io/examples/timeseries/timeseries_weather_forecasting/.

R. Barrio. Performance of the Taylor series method for ODEs/DAEs. *Applied Mathematics and Computation*, 163(2):525–545, 2005. ISSN 0096-3003. https://doi.org/10.1016/j.amc.2004.02.015.

A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal-recovery

problems. *Convex Optimization in Signal Processing and Communications*, pages 42–88, 2009. https://doi.org/10.1017/CBO9780511804458.003.

G. Benettin and A. Giorgilli. On the Hamiltonian interpolation of near-to-the identity symplectic mappings with application to symplectic integration algorithms. *Journal of Statistical Physics*, 74:1117–1143, 03 1994. https://doi.org/10.1007/BF02188219.

R. Benito and D. Martín de Diego. Discrete vakonomic mechanics. *Journal of Mathematical Physics*, 46(8):083521, 2005. https://doi.org/10.1063/1.2008214.

T. Bertalan, F. Dietrich, I. Mezić, and I. G. Kevrekidis. On learning Hamiltonian systems from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12):121107, 2019. https://doi.org/10.1063/1.5128231.

D. Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, 2009. http://www.athenasc.com/convexduality.html.

M. Betancourt, M. Jordan, and A. Wilson. On symplectic optimization. 2018. https://arxiv.org/abs/1802.03653.

J. Bettencourt, M. Johnson, and D. Duvenaud. Taylor-mode automatic differentiation for higher-order derivatives in JAX. 2019. https://openreview.net/forum?id=SkxEF3FNPH.

F. Biscani and D. Izzo. Revisiting high-order Taylor methods for astrodynamics and celestial mechanics. *Monthly Notices of the Royal Astronomical Society*, 504(2):2614–2628, 04 2021. ISSN 0035-8711. https://doi.org/10.1093/mnras/stab1032.

S. Blanes and C. J. Budd. Explicit adaptive symplectic (easy) integrators: A scaling invariant generalisation of the Levi-Civita and KS regularisations. *Celestial Mechanics and Dynamical Astronomy*, 89(4):383–405, 2004. https://doi.org/10.1023/B:CELE.0000043572.30802.83.

S. Blanes and F. Casas. *A Concise Introduction to Geometric Numerical Integration*. 2017. ISBN 9781482263442. https://doi.org/10.1201/b21563.

S. Blanes and A. Iserles. Explicit adaptive symplectic integrators for solving Hamiltonian systems. *Celestial Mechanics and Dynamical Astronomy*, 114(3):297–317, 2012. https://doi.org/10.1007/s10569-012-9441-z.

S. Blanes, F. Casas, J. A. Oteo, and J. Ros. The Magnus expansion and some of its applications. *Physics Reports*, 470:151–238, 11 2008. https://doi.org/10.1016/j.physrep.2008.11.001.

V. Boltyanski, H. Martini, V. Soltan, and V. Soltan. *Geometric Methods and Optimization*

Problems. Combinatorial Optimization. Springer US, 1999. https://doi.org/10.1007/978-1-4615-5319-9.

F. Borrelli, A. Bemporad, and M. Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017. https://doi.org/10.1017/9781139061759.

N. Bou-Rabee and J. E. Marsden. Hamilton–Pontryagin integrators on Lie groups part I: Introduction and structure-preserving properties. *Foundations of Computational Mathematics*, 9(2):197–219, 2009. https://doi.org/10.1007/s10208-008-9030-4.

N. Boumal. An introduction to optimization on smooth manifolds, 2020. http://www.nicolasboumal.net/book.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. https://doi.org/10.1017/CBO9780511804441.

J. W. Burby. Slow manifold reduction as a systematic tool for revealing the geometry of phase space. *Phys. Plasmas*, 29:042102, 2022. doi: 10.1063/5.0084543. https://doi.org/10.1063/5.0084543.

J. W. Burby and E. Hirvijoki. Normal stability of slow manifolds in nearly periodic Hamiltonian systems. *Journal of Mathematical Physics*, 62(9):093506, 2021. https://doi.org/10.1063/5.0054323.

J. W. Burby and T. J. Klotz. Slow manifold reduction for plasma science. *Comm. Nonlin. Sci. Numer. Simul.*, 89:105289, 2020. https://doi.org/10.1016/j.cnsns.2020.105289.

J. W. Burby and J. Squire. General formulas for adiabatic invariants in nearly periodic Hamiltonian systems. *Journal of Plasma Physics*, 86(6):835860601, 2020. https://doi.org/10.1017/S002237782000080X.

J. W. Burby, A. J. Brizard, P. J. Morrison, and H. Qin. Hamiltonian gyrokinetic vlasov-maxwell system. *Phys. Lett. A*, 379:2073, 2015. https://doi.org/10.1016/j.physleta.2015.06.051.

J. W. Burby, Q. Tang, and R. Maulik. Fast neural Poincaré maps for toroidal magnetic fields. *Plasma Physics and Controlled Fusion*, 63(2):024001, dec 2020. https://doi.org/10.1088/1361-6587/abcbaa.

J. W. Burby, E. Hirvijoki, and M. Leok. Nearly-periodic maps and geometric integration of noncanonical Hamiltonian systems. *Journal of Nonlinear Science*, 33(38), 2023. https://doi.org/10.1007/s00332-023-09891-4.

J. P. Calvo and J. M. Sanz-Serna. The development of variable-step symplectic integrators,

with application to the two-body problem. *SIAM J. Sci. Comp.*, 14(4):936–952, 1993. https://doi.org/10.1137/0914057.

C. M. Campos, A. Mahillo, and D. M. de Diego. A discrete variational derivation of accelerated methods in optimization. 2021. https://arxiv.org/abs/2106.02700.

F. Casas and B. Owren. Cost efficient Lie group integrators in the RKMK class. *BIT*, 43: 723–742, 11 2003. https://doi.org/10.1023/B:BITN.0000009959.29287.d4.

A. L. Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. *Acad. Sci. Paris*, 25:536–538, 1847. https://doi.org/10.1017/CBO9780511702396.063.

E. Celledoni, A. Marthinsen, and B. Owren. Commutator-free Lie group methods. *Future Gener. Comput. Syst.*, 19:341–352, 2003. https://doi.org/10.1016/S0167-739X(02)00161-9.

E. Celledoni, H. Marthinsen, and B. Owren. An introduction to Lie group integrators – basics, new developments and applications. *Journal of Computational Physics*, 257: 1040–1061, 2014. ISSN 0021-9991. https://doi.org/10.1016/j.jcp.2012.12.031.

E. Celledoni, E. Çokaj, A. Leone, D. Murari, and B. Owren. Lie group integrators for mechanical systems. *International Journal of Computer Mathematics*, 99(1):58–88, 2022. https://doi.org/10.1080/00207160.2021.1966772.

G. Chen and L. Chacón. A multi-dimensional, energy-and charge-conserving, nonlinearly implicit, electromagnetic Vlasov–Darwin particle-in-cell algorithm. *Computer Physics Communications*, 197:73–87, 2015. https://doi.org/10.1016/j.cpc.2015.08.008.

G. Chen, L. Chacón, and D. C. Barnes. An energy-and charge-conserving, implicit, electrostatic particle-in-cell algorithm. *Journal of Computational Physics*, 230(18): 7018–7036, 2011. https://doi.org/10.1016/j.jcp.2011.05.031.

Y. Chen, T. Matsubara, and T. Yaguchi. Neural symplectic form: learning Hamiltonian equations on general coordinate systems. In *Advances in Neural Information Processing Systems*, 2021. https://openreview.net/pdf?id=4h4oqp-ATxb.

Z. Chen, J. Zhang, M. Arjovsky, and L. Bottou. Symplectic Recurrent Neural Networks. *International Conference on Learning Representations*, 2020. https://openreview.net/pdf?id=BkgYPREtPr.

S. H. Christiansen, H. Z. Munthe-Kaas, and B. Owren. Topics in structure-preserving discretization. *Acta Numerica*, 20:1–119, 2011. https://doi.org/10.1017/S096249291100002X.

B. Christianson. Automatic Hessians by reverse accumulation. *IMA Journal of Numerical Analysis*, 12(2):135–150, 04 1992. ISSN 0272-4979. https://doi.org/10.1093/imanum/12.2.135.

O. B. Cieza and J. Reger. IDA-PBC for underactuated mechanical systems in implicit Port-Hamiltonian representation. In *European Control Conference (ECC)*, 2019. https://doi.org/10.23919/ECC.2019.8795994.

J. Cortés and S. Martínez. Non-holonomic integrators. *Nonlinearity*, 14(5):1365–1392, 2001. https://doi.org/10.1088/0951-7715/14/5/322.

J. Cortés, M. de León, D. Martín de Diego, and S. Martínez. Geometric description of vakonomic and nonholonomic dynamics. comparison of solutions. *SIAM Journal on Control and Optimization*, 41(5):1389–1412, 2002. https://doi.org/10.1137/S036301290036817X.

C. J. Cotter and S. Reich. Adiabatic invariance and applications: From molecular dynamics to numerical weather prediction. *BIT Numer. Math.*, 44:439, 2004. https://doi.org/10.1023/B:BITN.0000046816.68632.49.

M. Cranmer, S. Greydanus, S. Hoyer, P. W. Battaglia, D. N. Spergel, and S. Ho. Lagrangian neural networks. *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020. https://greydanus.github.io/2020/03/10/lagrangian-nns.

P. E. Crouch and R. L. Grossman. Numerical integration of ordinary differential equations on manifolds. *Journal of Nonlinear Science*, 3:1–33, 1993. https://doi.org/10.1007/BF02429858.

Y.-H. Dai, L.-Z. Liao, and D. Li. On restart procedures for the conjugate gradient method: Theory and practice in optimization. *Numerical Algorithms*, 35, 04 2004. https://doi.org/10.1023/B:NUMA.0000021761.10993.6e.

Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems*, volume 27, 2014. https://dl.acm.org/doi/10.5555/2969033.2969154.

M. David and F. Méhats. Symplectic learning for Hamiltonian neural networks, 2021. https://arxiv.org/abs/2106.11753.

M. de León, D. Martín de Diego, and A. Santamaría-Merino. Geometric numerical integration of nonholonomic systems and optimal control problems. *European Journal of Control*, 10(5):515–521, 2004. https://doi.org/10.3166/ejc.10.515-521.

M. de León, D. Martín de Diego, and A. Santamaría-Merino. Discrete variational integrators and optimal control theory. *Advances in Computational Mathematics*, 26:251–268, 2007. https://doi.org/10.1007/s10444-004-4093-5.

A. Deaño, D. Huybrechs, and A. Iserles. *Computing Highly Oscillatory Integrals*. SIAM, Philadelphia, January 2018. https://doi.org/10.1137/1.9781611975123.

K. Donghwan and J. Fessler. Adaptive restart of the optimized gradient method for convex optimization. *Journal of Optimization Theory and Applications*, 178, 07 2018. https://doi.org/10.1007/s10957-018-1287-4.

Z. Drezner and H. Hamacher. *Facility Location: Applications and Theory*. Springer Berlin Heidelberg, 2002. ISBN 9783540213451. https://link.springer.com/book/9783540421726.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. https://jmlr.org/papers/v12/duchi11a.html.

T. Duong and N. Atanasov. Hamiltonian-based Neural ODE Networks on the SE(3) Manifold For Dynamics Learning and Control. In *Proceedings of Robotics: Science and Systems*, 2021. https://doi.org/10.15607/RSS.2021.XVII.086.

T. Duong and N. Atanasov. Adaptive control of SE(3) Hamiltonian dynamics with learned disturbance features. *IEEE Control Systems Letters*, 2022. https://doi.org/10.1109/LCSYS.2022.3177156.

V. Duruisseaux and A. Chakraborty. An operator learning framework for spatiotemporal super-resolution of scientific simulations. 2023. https://arxiv.org/abs/2311.02328.

V. Duruisseaux and A. R. Humphries. Bistability, bifurcations and chaos in the Mackey-Glass equation. *Journal of Computational Dynamics*, 9(3):421–450, 2022. http://doi.org/10.3934/jcd.2022009.

V. Duruisseaux and M. Leok. Accelerated optimization on Riemannian manifolds via discrete constrained variational integrators. *Journal of Nonlinear Science*, 32(42), 2022a. https://doi.org/10.1007/s00332-022-09795-9.

V. Duruisseaux and M. Leok. Variational accelerated optimization on Riemannian manifolds. *IEICE Proceedings Series 71 (B1L-D-03). International Symposium on Nonlinear Theory and its Applications (NOLTA)*, 2022b. https://doi.org/10.34385/proc.71.B1L-D-03.

V. Duruisseaux and M. Leok. Accelerated optimization on Riemannian manifolds via

projected variational integrators. 2022c. https://arxiv.org/abs/2201.02904.

V. Duruisseaux and M. Leok. A variational formulation of accelerated optimization on Riemannian manifolds. *SIAM Journal on Mathematics of Data Science*, 4(2):649–674, 2022d. https://doi.org/10.1137/21M1395648.

V. Duruisseaux and M. Leok. Time-adaptive Lagrangian variational integrators for accelerated optimization on manifolds. *Journal of Geometric Mechanics*, 15(1):224–255, 2023a. ISSN 1941-4889. https://www.aimspress.com/article/doi/10.3934/jgm.2023010.

V. Duruisseaux and M. Leok. Practical perspectives on symplectic accelerated optimization. *Optimization Methods and Software*, 38(6):1230–1268, 2023b. https://doi.org/10.1080/10556788.2023.2214837.

V. Duruisseaux, J. Schmitt, and M. Leok. Adaptive Hamiltonian variational integrators and applications to symplectic accelerated optimization. *SIAM Journal on Scientific Computing*, 43(4):A2949–A2980, 2021. https://doi.org/10.1137/20M1383835.

V. Duruisseaux, J. W. Burby, and Q. Tang. Approximation of nearly-periodic symplectic maps via structure-preserving neural networks. *Scientific Reports*, 13(8351), 2023a. https://doi.org/10.1038/s41598-023-34862-w.

V. Duruisseaux, J. W. Burby, and Q. Tang. Code demonstration: Approximation of nearly-periodic symplectic maps via structure-preserving neural networks. 2023b. doi: 10.2172/1972078. URL https://www.osti.gov/biblio/1972078.

V. Duruisseaux, T. Duong, M. Leok, and N. Atanasov. Lie group forced variational integrator networks for learning and control of robot systems. In *Proceedings of the 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pages 731–744. PMLR, 2023c. https://proceedings.mlr.press/v211/duruisseaux23a.html.

L. Eldén and H. Park. A Procrustes problem on the Stiefel manifold. *Numerische Mathematik*, 82(4):599–619, 1999. https://doi.org/10.1007/s002110050432.

W. M. Farr. Variational integrators for almost-integrable systems. *Celestial Mechanics and Dynamical Astronomy*, 102(2):105–118, 2009. https://doi.org/10.1007/s10569-008-9172-3.

Y. N. Fedorov and D. V. Zenkov. Discrete nonholonomic LL systems on Lie groups. *Nonlinearity*, 18:2211–2241, 2005. https://doi.org/10.1088/0951-7715/18/5/017.

O. Fercoq and Z. Qu. Restarting accelerated gradient methods with a rough strong

convexity estimate. Research Report 1609.07358, Télécom ParisTech, 2016. https://hal.telecom-paris.fr/hal-02287730.

O. Fercoq and Z. Qu. Adaptive restart of accelerated gradient methods under local quadratic growth condition. *IMA Journal of Numerical Analysis*, Mar. 2019. https://doi.org/10.1093/imanum/drz007.

L. N. G. Filon. On a quadrature formula for trigonometric integrals. *Proceedings of the Royal Society of Edinburgh*, 49:38–47, 1930. https://doi.org/10.1017/S0370164600026262.

J. Gallier and J. Quaintance. *Differential Geometry and Lie Groups: A Computational Perspective*. Geometry and Computing. Springer International Publishing, 2020. ISBN 9783030460402. https://doi.org/10.1007/978-3-030-46040-2.

Z. Ge and J. E. Marsden. Lie–Poisson Hamilton–Jacobi theory and Lie-Poisson integrators. *Physics Letters A*, 133(3):134–139, 1988. https://doi.org/10.1016/0375-9601(88)90773-6.

P. Giselsson and S. Boyd. Monotonicity and restart in fast gradient methods. *Proceedings of the IEEE Conference on Decision and Control*, 2015:5058–5063, 02 2015. https://doi.org/10.1109/CDC.2014.7040179.

B. Gladman, M. Duncan, and J. Candy. Symplectic integrators for long-time integrations in celestial mechanics. *Celestial Mech. Dynamical Astronomy*, 52:221–240, 1991. https://doi.org/10.1007/BF00048485.

L. Godinho and J. Natário. *An Introduction to Riemannian Geometry: With Applications to Mechanics and Relativity*. Universitext. Springer International Publishing, 2014. ISBN 9783319086668. https://doi.org/10.1007/978-3-319-08666-8.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013. ISBN 9781421407944. https://www.press.jhu.edu/books/title/10678/matrix-computations.

S. l. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. https://greydanus.github.io/2019/05/15/hamiltonian-nns/.

L. Grüne and J. Pannek. *Nonlinear model predictive control*. Springer, 2017. https://doi.org/10.1007/978-0-85729-501-9.

E. Hairer. Variable time step integration with symplectic methods. *Applied Numerical Mathematics*, 25(2-3):219–227, 1997. https://doi.org/10.1016/S0168-9274(97)00061-5.

E. Hairer, C. Lubich, and G. Wanner. Geometric numerical integration illustrated by the

Störmer–Verlet method. *Acta Numerica*, 12:399 – 450, 2003. https://doi.org/10.1017/S0962492902000144.

E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. https://doi.org/10.1007/3-540-30666-8.

B. Hall. *Lie Groups, Lie Algebras, and Representations*. Graduate Texts in Mathematics. Springer Cham, second edition, 2015. https://doi.org/10.1007/978-3-319-13467-3.

J. Hall and M. Leok. Spectral Variational Integrators. *Numer. Math.*, 130(4):681–740, 2015. https://doi.org/10.1007/s00211-014-0679-0.

A. Havens and G. Chowdhary. Forced variational integrator networks for prediction and control of mechanical systems. 2021. http://proceedings.mlr.press/v144/havens21a/havens21a.pdf.

Q. Hernandez, A. Badías, D. González, F. Chinesta, and E. Cueto. Deep learning of thermodynamics-aware reduced-order models from data. *Computer Methods in Applied Mechanics and Engineering*, 379:113763, 2021. ISSN 0045-7825. https://doi.org/10.1016/j.cma.2021.113763.

Q. Hernández, A. Badias, F. Chinesta, and E. Cueto. Port-metriplectic neural networks: thermodynamics-informed machine learning of complex physical systems. *Computational Mechanics*, pages 1–9, 03 2023. https://doi.org/10.1007/s00466-023-02296-w.

M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010. https://doi.org/10.1017/S0962492910000048.

D. Holm, T. Schmah, and C. Stoica. *Geometric Mechanics and Symmetry: From Finite to Infinite Dimensions*. Oxford Texts in Applied and Engineering Mathematics. OUP Oxford, 2009. ISBN 9780199212910. https://global.oup.com/academic/product/geometric-mechanics-and-symmetry-9780199212903.

S. Huang, Z. He, B. Chem, and C. Reina. Variational Onsager Neural Networks (VONNs): A thermodynamics-based variational learning strategy for non-equilibrium PDEs. *Journal of the Mechanics and Physics of Solids*, 163:104856, 2022. ISSN 0022-5096. https://doi.org/10.1016/j.jmps.2022.104856.

I. I. Hussein, M. Leok, A. K. Sanyal, and A. M. Bloch. A discrete variational integrator for optimal control problems on SO(3). *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6636–6641, 2006. https://doi.org/10.1109/CDC.2006.377818.

A. Iserles and S. P. Nørsett. On the solution of linear differential equations in Lie groups. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 357(1754): 983–1019, 1999. ISSN 1364503X. https://doi.org/10.1098/rsta.1999.0362.

A. Iserles and G. Quispel. Why geometric numerical integration ? In *Discrete Mechanics, Geometric Integration and Lie–Butcher Series*, pages 1–28, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01397-4. https://doi.org/10.1007/978-3-030-01397-4_1.

A. Iserles, H. Z. Munthe-Kaas, S. Nørsett, and A. Zanna. Lie group methods. In *Acta Numerica*, volume 9, pages 215–365. Cambridge University Press, 2000. https://doi.org/10.1017/S0962492900002154.

M. Jendoubi and R. May. On an asymptotically autonomous system with Tikhonov type regularizing term. *Archiv der Mathematik*, 95:389–399, 10 2010. https://doi.org/10.1007/s00013-010-0181-6.

F. Jiménez and D. Martín de Diego. Continuous and discrete approaches to vakonomic mechanics. *Revista De La Real Academia De Ciencias Exactas Fisicas Y Naturales Serie A-matematicas*, 106, 03 2012. https://doi.org/10.1007/s13398-011-0028-4.

C. Jin, P. Netrapalli, and M. I. Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1042–1085, 2018. http://proceedings.mlr.press/v75/jin18a.html.

C. Jin, P. Netrapalli, R. Ge, S. M. Kakade, and M. I. Jordan. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *J. ACM*, 68(2), 2021. ISSN 0004-5411. https://doi.org/10.1145/3418526.

P. Jin, Z. Zhang, A. Zhu, Y. Tang, and G. E. Karniadakis. SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. *Neural Networks*, 132(C), 12 2020. https://doi.org/10.1016/j.neunet.2020.08.017.

M. I. Jordan. Dynamical, symplectic and stochastic perspectives on gradient-based optimization. In *Proceedings of the International Congress of Mathematicians (ICM 2018)*, pages 523–549, 2018. https://doi.org/10.1142/9789813272880_0022.

J. Jost. *Riemannian Geometry and Geometric Analysis*. Universitext. Springer, Cham, 7th edition, 2017. https://doi.org/10.1007/978-3-319-61860-9.

O. Junge, J. E. Marsden, and S. Ober-Blöbaum. Discrete mechanics and optimal control. *IFAC Proceedings Volumes*, 38(1):538–543, 2005. https://doi.org/10.3182/20050703-6-CZ-1902.00745.

C. Kane, J. Marsden, and M. Ortiz. Symplectic-energy-momentum preserving variational integrators. *Journal of Mathematical Physics*, 40(7), 1999. https://doi.org/10.1063/1.532892.

G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. https://doi.org/10.1038/s42254-021-00314-5.

J. Kelley. *General Topology*. Graduate Texts in Mathematics. Springer New York, 1975. ISBN 9780387901251. https://link.springer.com/book/9780387901251.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. https://arxiv.org/abs/1412.6980.

G. Kirlinger and G. F. Corliss. On implicit Taylor series methods for stiff ODEs. 1 1991. https://www.osti.gov/biblio/6097907.

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

M. Kruskal. Asymptotic theory of Hamiltonian and other systems with all solutions nearly periodic. *Journal of Mathematical Physics*, 3(4):806–828, 1962. https://doi.org/10.1063/1.1724285.

S. Lall and M. West. Discrete variational Hamiltonian mechanics. *J. Phys. A*, 39(19): 5509–5519, 2006. https://doi.org/10.1088/0305-4470/39/19/S11.

S. Lang. *Fundamentals of Differential Geometry*, volume 191 of *Graduate Texts in Mathematics*. Springer -Verlag, New York, 1999. ISBN 9780387985930. https://doi.org/10.1007/978-1-4612-0541-8.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. https://doi.org/10.1109/5.726791.

J. Lee. *Introduction to Riemannian Manifolds*, volume 176 of *Graduate Texts in Mathematics*. Springer, Cham, second edition, 2018. https://doi.org/10.1007/978-3-319-91755-9.

T. Lee. Computational geometric mechanics and control of rigid bodies. *Ph.D. dissertation, University of Michigan*, 2008. https://deepblue.lib.umich.edu/handle/2027.42/60804.

T. Lee, N. H. McClamroch, and M. Leok. A Lie group variational integrator for the

attitude dynamics of a rigid body with applications to the 3d pendulum. *Proceedings of 2005 IEEE Conference on Control Applications. (CCA 2005)*, pages 962–967. https://doi.org/10.1109/CCA.2005.1507254.

T. Lee, M. Leok, and N. McClamroch. Lie group variational integrators for the full body problem. *Comput. Methods Appl. Mech. Engrg.*, 196(29-30):2907–2924, 2007a. https://doi.org/10.1016/j.cma.2007.01.017.

T. Lee, M. Leok, and N. McClamroch. Lie group variational integrators for the full body problem in orbital mechanics. *Celestial Mechanics and Dynamical Astronomy*, 98(2): 121–144, 2007b. https://doi.org/10.1007/s10569-007-9073-x.

T. Lee, M. Leok, and N. McClamroch. Lagrangian mechanics and variational integrators on two-spheres. *International Journal for Numerical Methods in Engineering*, 79(9): 1147–1174, 2009. https://doi.org/10.1002/nme.2603.

T. Lee, M. Leok, and N. H. McClamroch. *Global Formulations of Lagrangian and Hamiltonian Dynamics on Manifolds: A Geometric Approach to Modeling and Analysis*. Interaction of Mechanics and Mathematics. Springer International Publishing, 2017. ISBN 9783319569536. https://doi.org/10.1007/978-3-319-56953-6.

T. Lee, M. Tao, and M. Leok. Variational symplectic accelerated optimization on Lie groups. In *60th IEEE Conference on Decision and Control (CDC)*, pages 233–240, 2021. https://doi.org/10.1109/CDC45484.2021.9683657.

H. Lei, L. Wu, and E. Weinan. Machine-learning-based non-Newtonian fluid model with molecular fidelity. *Physical Review E*, 102(4):043309, 2020. https://doi.org/10.1103/PhysRevE.102.043309.

B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*, volume 14 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2004. https://doi.org/10.1017/CBO9780511614118.

M. Leok. Generalized Galerkin variational integrators: Lie group, multiscale, and pseudospectral methods. http://arxiv.org/abs/math/0508360, 2004.

M. Leok and T. Ohsawa. Variational and geometric structures of discrete Dirac mechanics. *Found. Comput. Math.*, 11(5):529–562, 2011. https://doi.org/10.1007/s10208-011-9096-2.

M. Leok and T. Shingel. Prolongation-collocation variational integrators. *IMA J. Numer. Anal.*, 32(3):1194–1216, 2012a. https://doi.org/10.1093/imanum/drr042.

M. Leok and T. Shingel. General techniques for constructing variational integrators. *Front.*

Math. China, 7(2):273–303, 2012b. https://doi.org/10.1007/s11464-012-0190-9.

M. Leok and J. Zhang. Discrete Hamiltonian variational integrators. *IMA Journal of Numerical Analysis*, 31(4):1497–1532, 2011. https://doi.org/10.1093/imanum/drq027.

D. Levin. Procedures for computing one- and two-dimensional integrals of functions with rapid irregular oscillations. *Mathematics of Computation*, 38(158):531–538, 1982. https://doi.org/10.2307/2007287.

D. Levin. Fast integration of rapidly oscillatory functions. *Journal of Computational and Applied Mathematics*, 67(1):95–101, 1996. ISSN 0377-0427. https://doi.org/10.1016/0377-0427(94)00118-9.

D. Lewis and J. C. Simo. Conserving algorithms for the dynamics of Hamiltonian systems on Lie groups. *Journal of Nonlinear Science*, 4(1):253–299, Dec. 1994. https://doi.org/10.1007/BF02430634.

W. Lin, V. Duruisseaux, M. Leok, F. Nielsen, M. E. Khan, and M. Schmidt. Practical structured Riemannian optimization with momentum by using generalized normal coordinates. 2022. https://openreview.net/forum?id=1aybhSfabqh.

W. Lin, V. Duruisseaux, M. Leok, F. Nielsen, M. E. Khan, and M. Schmidt. Simplifying momentum-based positive-definite submanifold optimization with applications to deep learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21026–21050. PMLR, 2023. https://proceedings.mlr.press/v202/lin23c.html.

Y. Liu, F. Shang, J. Cheng, H. Cheng, and L. Jiao. Accelerated first-order methods for geodesically convex optimization on Riemannian manifolds. In *NeurIPS*, volume 30, pages 4868–4877, 2017. https://papers.nips.cc/paper/2017/hash/6ef80bb237adf4b6f77d0700e1255907-Abstract.html.

E. N. Lorenz. On the existence of a slow manifold. *J. Atmos. Sci.*, 43:1547–1557, 1986. https://doi.org/10.1175/1520-0469(1986)043<1547:OTEOAS>2.0.CO;2.

E. N. Lorenz. The slow manifold – what is it? *J. Atmos. Sci.*, 49:2449–2451, 1992. https://doi.org/10.1175/1520-0469(1992)049<2449:TSMII>2.0.CO;2.

E. N. Lorenz and V. Krishnamurthy. On the nonexistence of a slow manifold. *J. Atmos. Sci.*, 44:2940–2950, 1987. https://doi.org/10.1175/1520-0469(1987)044<2940:OTNOAS>2.0.CO;2.

M. Lutter, K. Listmann, and J. Peters. Deep Lagrangian Networks for end-to-end

learning of energy-based control for under-actuated systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019a. https://doi.org/10.1109/IROS40897.2019.8968268.

M. Lutter, C. Ritter, and J. Peters. Deep Lagrangian networks: Using physics as model prior for deep learning. In *International Conference on Learning Representations*, 2019b. https://openreview.net/pdf?id=BklHpjCqKm.

R. S. MacKay. Slow manifolds. In *Energy Localization and Transfer*, volume 22 of *Advanced Series in Nonlinear Dynamics*, pages 149–192. World Scientific, 2004. https://doi.org/10.1142/5458.

J. Marsden and T. Ratiu. *Introduction to mechanics and symmetry*, volume 17 of *Texts in Applied Mathematics*. Springer-Verlag, New York, second edition, 1999. https://doi.org/10.1007/978-0-387-21792-5.

J. E. Marsden and M. West. Discrete mechanics and variational integrators. *Acta Numer.*, 10:357–514, 2001. https://doi.org/10.1017/S096249290100006X.

F. B. Mathiesen, B. Yang, and J. Hu. Hyperverlet: A symplectic hypersolver for Hamiltonian systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4): 4575–4582, June 2022. https://doi.org/10.1609/aaai.v36i4.20381.

A. McInerney. *First Steps in Differential Geometry: Riemannian, Contact, Symplectic*. Undergraduate Texts in Mathematics. Springer New York, 2013. ISBN 9781461477327. https://doi.org/10.1007/978-1-4614-7732-7.

R. I. McLachlan and M. Perlmutter. Integrators for nonholonomic mechanical systems. *Journal of Nonlinear Science*, 16:283–328, 2006. https://doi.org/10.1007/s00332-005-0698-1.

S. T. Miller, E. C. Cyr, J. N. Shadid, R. M. J. Kramer, E. G. Phillips, S. Conde, and R. P. Pawlowski. IMEX and exact sequence discretization of the multi-fluid plasma model. *Journal of Computational Physics*, 397:108806, 2019. https://doi.org/10.1016/j.jcp.2019.05.052.

K. Modin and C. Führer. Time-step adaptivity in variational integrators with application to contact problems. *Z. Angew. Math. Mech.*, 86(10):785–794, 2006. https://doi.org/10.1002/zamm.200610286.

P. J. Morrison. The Maxwell-Vlasov equations as a continuous hamiltonian system. *Physics Letters A*, 80(5):383–386, 1980. ISSN 0375-9601. https://doi.org/10.1016/0375-9601(80)90776-8.

P. J. Morrison. Nonlinear stability of fluid and plasma equilibria. *Rev. Mod. Phys.*, 70:467, 1998. https://doi.org/10.1103/RevModPhys.70.467.

P. J. Morrison and J. M. Greene. Noncanonical Hamiltonian density formulation of hydrodynamics and ideal magnetohydrodynamics. *Phys. Rev. Lett.*, 45:790–794, Sep 1980. https://doi.org/10.1103/PhysRevLett.45.790.

P. J. Morrison and M. Kotschenreuther. The free energy principle, negative energy modes, and stability. In *Presented at the 4th International Workshop on Nonlinear and Turbulent Processes in Physics*, pages 9–22, Oct 1989. https://web2.ph.utexas.edu/~morrison/86IFSR280_morrison.pdf.

P. J. Morrison and J. Vanneste. Weakly nonlinear dynamics in noncanonical hamiltonian systems with applications to fluids and plasmas. *Ann. Phys.*, 368:117, 2016. https://doi.org/10.1016/j.aop.2016.02.003.

M. Muehlebach and M. I. Jordan. A dynamical systems perspective on Nesterov acceleration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *PMLR*, Long Beach, CA, USA, 2019. http://proceedings.mlr.press/v97/muehlebach19a.

H. Z. Munthe-Kaas. Lie-Butcher theory for Runge-Kutta methods. *BIT*, 35:572–587, 1995. https://doi.org/10.1007/BF01739828.

H. Z. Munthe-Kaas. Runge-Kutta methods on Lie groups. *BIT Numerical Mathematics*, 38:92–111, 1998. https://doi.org/10.1007/BF02510919.

H. Z. Munthe-Kaas. High order Runge-Kutta methods on manifolds. *Applied Numerical Mathematics*, 29:115–127, 1999. https://doi.org/10.1016/S0168-9274(98)00030-0.

S. Nair. Time adaptive variational integrators: A space-time geodesic approach. *Physica D: Nonlinear Phenomena*, 241(4):315–325, 2012. https://doi.org/10.1016/j.physd.2011.09.006.

J. Nash. The imbedding problem for Riemannian manifolds. *Annals of Mathematics*, 63 (1):20–63, 1956. ISSN 0003486X. https://doi.org/10.2307/1969989.

R. D. Neidinger. Directions for computing truncated multivariate Taylor series. *Mathematics of Computation*, 74(249):321–340, 2005. https://doi.org/10.1090/S0025-5718-04-01657-6.

R. D. Neidinger. Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming. *SIAM Review*, 52(3):545–563, 2010. https://doi.org/10.1137/080743627.

R. D. Neidinger. Efficient recurrence relations for univariate and multivariate Taylor series coefficients. *Dynamical Systems and Differential Equations*, Special, Proceedings of the 9th AIMS International Conference (Orlando, USA):587–596, 2013. https://doi.org/10.3934/proc.2013.2013.587.

A. Nemirovsky and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley - Interscience series in discrete mathematics. Wiley, 1983. https://www2.isye.gatech.edu/~nemirovs/Nemirovskii_Yudin_1983.pdf.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983. http://mi.mathnet.ru/eng/dan/v269/i3/p543.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, MA, 2004. https://doi.org/10.1007/978-1-4419-8853-9.

Y. Nesterov. Accelerating the cubic regularization of Newton's method on convex problems. *Math. Program.*, 112:159–181, 2008. https://doi.org/10.1007/s10107-006-0089-x.

N. Nordkvist and A. K. Sanyal. A Lie group variational integrator for rigid body motion in SE(3) with applications to underwater vehicle dynamics. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5414–5419, 2010. https://doi.org/10.1109/CDC.2010.5717622.

S. Ober-Blöbaum, O. Junge, and J. E. Marsden. Discrete mechanics and optimal control: An analysis. *ESAIM: Control, Optimisation and Calculus of Variations*, 17(2):322–352, 2011. https://doi.org/10.1051/cocv/2010012.

B. O'donoghue and E. Candès. Adaptive restart for accelerated gradient schemes. *Found. Comput. Math.*, 15(3):715–732, jun 2015. ISSN 1615-3375. https://doi.org/10.1007/s10208-013-9150-3.

C. Offen and S. Ober-Blöbaum. Symplectic integration of learned Hamiltonian systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32(1):013122, 2022. https://doi.org/10.1063/5.0065913.

R. Ortega, M. W. Spong, F. Gómez-Estern, and G. Blankenstein. Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment. *IEEE Transactions on Automatic Control*, 47(8), 2002. https://doi.org/10.1109/TAC.2002.800770.

A. Orvieto and A. Lucchi. Shadowing properties of optimization algorithms. In *Advances in Neural Information Processing Systems*, volume 32, pages 12692–12703, 2019. https:

//papers.nips.cc/paper/2019/file/8471bda5e6201d30893c3582ee131d4d-Paper.pdf.

J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. P. Schöllig. Learning to fly: a PyBullet gym environment to learn the control of multiple nano-quadcopters, 2020. https://github.com/utiasDSL/gym-pybullet-drones.

M. A. Patterson, M. Weinstein, and A. V. Rao. An efficient overloaded method for computing derivatives of mathematical functions in MATLAB. *ACM Trans. Math. Softw.*, 39(3), May 2013. ISSN 0098-3500. https://doi.org/10.1145/2450153.2450155.

S. Paul and S. Rakshit. Large-scale multi-label text classification. *Keras Tutorial*, 2020. https://keras.io/examples/nlp/multi_label_classification/.

B. A. Pearlmutter. Lazy multivariate higher-order forward-mode AD. In *In Proceedings of the 2007 Symposium on Principles of Programming Languages*, 2007. https://doi.org/10.1145/1190216.1190242.

D. L. Phillips. A technique for the numerical solution of certain integral equations of the first kind. *J. ACM*, 9(1):84–97, jan 1962. ISSN 0004-5411. https://doi.org/10.1145/321105.321114.

P. Pihajoki. Explicit methods in extended phase space for inseparable Hamiltonian problems. *Celestial Mechanics and Dynamical Astronomy*, 121:211–231, 2015. https://doi.org/10.1007/s10569-014-9597-9.

H. Poincaré. *Les méthodes nouvelles de la mécanique céleste, Volume 3*. Gauthier-Villars, Paris, 1899. http://hdl.handle.net/1908/3853.

M. J. D. Powell. Restart procedures for the conjugate gradient method. *Mathematical Programming*, 12:241–254, 1977. https://doi.org/10.1007/BF01593790.

H. Qin. Machine learning and serving of discrete field theories. *Scientific Reports*, 10(1): 1–15, 2020. https://doi.org/10.1038/s41598-020-76301-0.

K. Rath, C. G. Albert, B. Bischl, and U. von Toussaint. Symplectic gaussian process regression of maps in hamiltonian systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(5):053121, 2021. https://doi.org/10.1063/5.0048129.

F. Raymond. Classification of the actions of the circle on 3-manifolds. *Transactions of the American Mathematical Society*, 131:51–78, 1968. https://doi.org/10.2307/1994680.

S. Reich. Backward error analysis for numerical integrators. *SIAM J. Numer. Anal.*, 36: 1549–1570, 1999. https://doi.org/10.1137/S0036142997329797.

J. Renegar and B. Grimmer. A simple nearly optimal restart scheme for speeding up first-order methods. *Found. Comput. Math.*, 22(1):211–256, feb 2022. ISSN 1615-3375. https://doi.org/10.1007/s10208-021-09502-2.

V. Roulet and A. d'Aspremont. Sharpness, restart, and acceleration. *SIAM Journal on Optimization*, 30(1):262–289, 2020. https://doi.org/10.1137/18M1224568.

T. K. Rusch and S. Mishra. UnICORNN: A recurrent model for learning very long time dependencies. In *Proceedings of the 38th Int. Conf. on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9168–9178, 2021. https://proceedings.mlr.press/v139/rusch21a.html.

S. Sæmundsson, A. Terenin, K. Hofmann, and M. P. Deisenroth. Variational integrator networks for physically structured embeddings. In *AISTATS*, 2020. http://proceedings.mlr.press/v108/saemundsson20a.

J. A. Sanders, F. Verhulst, and J. Murdock. *Averaging Methods in Nonlinear Dynamical Systems*. Applied Mathematical Sciences. Springer New York, 2007. ISBN 9780387489186. https://doi.org/10.1007/978-0-387-48918-6.

S. Santos, M. Ekal, and R. Ventura. Symplectic momentum neural networks - using discrete variational mechanics as a prior in deep learning. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168 of *Proceedings of Machine Learning Research*, pages 584–595, Jun 2022. https://proceedings.mlr.press/v168/santos22a.html.

J. M. Schmitt and M. Leok. Properties of Hamiltonian variational integrators. *IMA Journal of Numerical Analysis*, 38(1):377–398, 03 2017. https://doi.org/10.1093/imanum/drx010.

J. M. Schmitt, T. Shingel, and M. Leok. Lagrangian and Hamiltonian Taylor variational integrators. *BIT Numerical Mathematics*, 58:457–488, 2018. https://doi.org/10.1007/s10543-017-0690-9.

X. Shen and M. Leok. Geometric exponential integrators. *J. Comput. Phys.*, 382:27–42, 2019. https://doi.org/10.1016/j.jcp.2019.01.005.

S. Smith, P. Kindermans, C. Ying, and Q. V. Le. Don't decay the learning rate, increase the batch size. 2018. https://openreview.net/pdf?id=B1Yy1BxCZ.

G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968. https://doi.org/10.1137/0705041.

J. Struckmeier. Hamiltonian dynamics on the symplectic extended phase space for

autonomous and non-autonomous systems. *Journal of Physics A*, 38:1257–1278, 2005. https://doi.org/10.1088/0305-4470/38/6/006.

W. Su, S. Boyd, and E. Candes. A differential equation for modeling Nesterov's Accelerated Gradient method: theory and insights. *Journal of Machine Learning Research*, 17(153): 1–43, 2016. https://jmlr.org/papers/v17/15-084.html.

Y. B. Suris. Hamiltonian methods of Runge-Kutta type and their variational interpretation. *Mat. Model.*, 2(4):78–87, 1990. http://mi.mathnet.ru/eng/mm/v2/i4/p78.

I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning - Volume 28*, ICML'13, pages 1139–1147, Atlanta, GA, USA, 2013. https://proceedings.mlr.press/v28/sutskever13.html.

M. Tao. Explicit symplectic approximation of nonseparable Hamiltonians: Algorithm and long time performance. *Physical Review E*, 94(4):043303, 2016. https://doi.org/10.1103/PhysRevE.94.043303.

M. Tao and T. Ohsawa. Variational optimization on Lie groups, with examples of leading (generalized) eigenvalue problems. In *Proceedings of the 23rd International AISTATS Conference*, volume 108 of *PMLR*, 2020. https://proceedings.mlr.press/v108/tao20a.html.

Y. Tassa, N. Mansard, and E. Todorov. Control-limited differential dynamic programming. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1168–1175, 2014. https://doi.org/10.1109/ICRA.2014.6907001.

T. Teshima, I. Ishikawa, K. Tojo, K. Oono, M. Ikeda, and M. Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. *Advances in Neural Information Processing Systems*, 33:3362–3373, 2020. https://proceedings.neurips.cc/paper/2020/file/2290a7385ed77cc5592dc2153229f082-Paper.pdf.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

T. Tieleman and G. Hinton. Coursera: Neural Networks for Machine Learning - Lecture 6.5: RMSprop. 2012. https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems*. V. H. Winston & Sons, 1977. https://doi.org/10.2307/2006360.

L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. Other Titles in Applied Mathematics. SIAM, 1997. ISBN 9780898719574. https://people.maths.ox.ac.uk/trefethen/text.html.

D. Turaev. Polynomial approximations of symplectic dynamics and richness of chaos in non-hyperbolic area-preserving maps. *Nonlinearity*, 16(1):123–135, nov 2002. https://doi.org/10.1088/0951-7715/16/1/308.

R. Valperga, K. Webster, D. Turaev, V. Klein, and J. Lamb. Learning reversible symplectic dynamics. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168 of *Proceedings of Machine Learning Research*, pages 906–916. PMLR, Jun 2022. https://proceedings.mlr.press/v168/valperga22a.html.

A. Van Der Schaft and D. Jeltsema. Port-Hamiltonian systems theory: An introductory overview. *Foundations and Trends in Systems and Control*, 1(2-3), 2014. https://doi.org/10.1561/2600000002.

V. Varadarajan. *Lie Groups, Lie Algebras, and Their Representations*. Graduate Texts in Mathematics. Springer New York, 1984. https://doi.org/10.1007/978-1-4612-1126-6.

F. Verhulst. *Nonlinear Differential Equations and Dynamic Systems*. 1996. ISBN 978-3-540-60934-6. https://doi.org/10.1007/978-3-642-61453-8.

G. Wahba. A Least Squares Estimate of Satellite Attitude. *SIAM Review*, 7(3):409, July 1965. https://doi.org/10.1137/1007077.

A. Weinstein. Symplectic manifolds and their Lagrangian submanifolds. *Advances in Mathematics*, 6(3):329–346, 1971. ISSN 0001-8708. https://doi.org/10.1016/0001-8708(71)90020-X.

H. Whitney. The singularities of a smooth n-manifold in $(2n-1)$-space. *Annals of Mathematics*, 45(2):247–293, 1944a. ISSN 0003486X. https://doi.org/10.2307/1969266.

H. Whitney. The self-intersections of a smooth n-manifold in $2n$-space. *Annals of Mathematics*, 45(2):220–246, 1944b. ISSN 0003486X. https://doi.org/10.2307/1969265.

A. Wibisono, A. Wilson, and M. Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016. https://doi.org/10.1073/pnas.1614734113.

J. H. Wilkinson. Error analysis of floating-point computation. *Numerische Mathematik*, 2:

319–340, 1960. https://doi.org/10.1007/BF01386233.

J. D. Willard, X. Jia, S. Xu, M. S. Steinbach, and V. Kumar. Integrating physics-based modeling with machine learning: A survey. 2020. https://beiyulincs.github.io/teach/fall_2020/papers/xiaowei.pdf.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017. https://arxiv.org/abs/1708.07747.

H. Yoshida. Construction of higher order symplectic integrators. *Physics Letters A*, 150(5): 262–268, 1990. ISSN 0375-9601. https://doi.org/10.1016/0375-9601(90)90092-3.

H. Yoshida. Recent progress in the theory and application of symplectic integrators. *Celestial Mechanics and Dynamical Astronomy*, 56(1-2):27–43, Mar. 1993. https://doi.org/10.1007/BF00699717.

H. Yoshimura and J. Marsden. Dirac structures in Lagrangian mechanics Part I: Implicit Lagrangian systems. *Journal of Geometry and Physics*, 57(1):133–156, 2006a. https://doi.org/10.1016/j.geomphys.2006.02.009.

H. Yoshimura and J. Marsden. Dirac structures in Lagrangian mechanics Part II: Variational structures. *Journal of Geometry and Physics*, 57(1):209–250, 2006b. https://doi.org/10.1016/j.geomphys.2006.02.012.

A. Zanna. Collocation and relaxed collocation for the Fer and the Magnus expansions. *SIAM Journal on Numerical Analysis*, 36(4):1145–1182, 1999. ISSN 00361429. https://doi.org/10.1137/S0036142997326616.

K. Zare and V. G. Szebehely. Time transformations in the extended phase-space. *Celestial mechanics*, 11:469–482, 1975. https://doi.org/10.1007/BF01650285.

H. Zhang and S. Sra. First-order methods for geodesically convex optimization. In *29th Annual Conference on Learning Theory*, pages 1617–1638, 2016. http://proceedings.mlr.press/v49/zhang16b.

H. Zhang and S. Sra. An estimate sequence for geodesically convex optimization. In *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1703–1723, Jul 2018. https://proceedings.mlr.press/v75/zhang18a.html.

J. Zhang, A. Mokhtari, S. Sra, and A. Jadbabaie. Direct Runge-Kutta discretization achieves acceleration. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. https://proceedings.neurips.cc/paper/2018/file/44968aece94f667e4095002d140b5896-Paper.pdf.

Y. D. Zhong, B. Dey, and A. Chakraborty. Symplectic ODE-Net: learning Hamiltonian dynamics with control. In *International Conference on Learning Representations (ICLR)*, 2019. https://openreview.net/pdf?id=ryxmb1rKDS.

Y. D. Zhong, B. Dey, and A. Chakraborty. Dissipative SymODEN: Encoding Hamiltonian dynamics with dissipation and control into deep learning. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020. https://openreview.net/pdf?id=knjWFNx6CN.

Y. D. Zhong, B. Dey, and A. Chakraborty. Benchmarking energy-conserving neural networks for learning dynamics from data. In *Learning for Dynamics and Control*, volume 144, pages 1218–1229. PMLR, 2021. https://proceedings.mlr.press/v144/zhong21a.html.

A. Zhu, P. Jin, and Y. Tang. Inverse modified differential equations for discovery of dynamics. 2020. https://arxiv.org/abs/2009.01058.