UNIVERSITY OF CALIFORNIA SAN DIEGO

Numerical Analysis of Curves on the Wasserstein Manifold

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

in

Mathematics

by

Nicholas Karris

Committee in charge:

Professor Alexander Cloninger, Chair
Professor Melvin Leok
Professor Yian Ma
Professor Rayan Saab

2026

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

University of California San Diego

2026

DEDICATION

To my parents, sister, family, friends, teachers, and
everyone else who cultivated my love of learning.

EPIGRAPH

Don't ever permit the pressure to exceed the pleasure.
Joe Maddon

Achieving the end of the exercise was never the point
of the exercise to begin with, was it?
Adam Savage

TABLE OF CONTENTS

Dissertation Approval Page iii
Dedication iv
Epigraph v
List of Figures xi
List of Tables xii
Acknowledgements xiii
Vita xviii
Publications xviii
Abstract of the Dissertation xix
Chapter 1 Introduction 1
1.1 Preliminaries 2
1.1.1 Optimal Transport Background 2
1.1.2 Tangent Fields and Linearized Optimal Transport 4
1.2 Overview 7
1.2.1 Euler's Method for Predicting Measure Evolution 7
1.2.2 Choosing an Appropriate Time Step 8
1.2.3 Newton's Method for Approximating Steady States 9
1.2.4 Image Interpolation Using Wasserstein Geodesics 10
Chapter 2 Euler's Method for Predicting Measure Evolution 12
2.1 Introduction 13
2.1.1 Practical Utility for Agent-based Models 16
2.1.2 Major Contributions 20
2.2 Euler-type Methods on the Wasserstein Manifold 21
2.2.1 Motivation from Euclidean Space 21
2.2.2 Euler's Method on the Wasserstein Manifold 22
2.2.3 Euler's Method with Imperfect Information 30
2.3 Euler-type Method for Empirical Measures of Particle Systems 32
2.3.1 Choosing an Appropriate Step Size 34
2.3.2 Additional Considerations 38
2.3.3 Detailed Description of Our Algorithm 39
2.4 Simulation Study I: Biological Agent-Based Model 41

2.4.1 Micro-scale Model Description . . . 41
2.4.2 Experimental Results . . . 43
2.4.3 Computational Improvement . . . 48
2.4.4 Comparison with Alternative Methods . . . 48
2.4.5 Comparison with Different Re-initializations . . . 53
2.5 Simulation Study II: Partial Differential Equation . . . 54
2.6 Simulation Study III: Langevin Dynamics . . . 59

Chapter 3 Choosing an Appropriate Time Step . . . 64
3.1 Introduction . . . 64
3.2 Orthogonality of the Model and Statistical Error . . . 66
3.3 Model Error . . . 70
3.4 Statistical Error . . . 73
3.4.1 Statistical Error with Independent Samples in 1D . . . 75
3.4.2 Statistical Error with Particle Timesteppers . . . 78
3.5 Consecutive Finite Difference Error . . . 85

Chapter 4 Newton's Method for Approximating Steady States . . . 89
4.1 Introduction . . . 90
4.2 Stochastic Particle Steady States and Optimal Transport . . . 98
4.2.1 Optimal Transport Background . . . 99
4.2.2 Evolving Measures as Curves in Wasserstein Space . . . 100
4.2.3 Wasserstein Formulation of Particle Steady States . . . 101
4.2.4 Three Examples . . . 104
4.2.4.1 Bacterial Chemotaxis . . . 104
4.2.4.2 Economic Agents . . . 106
4.2.4.3 The Half-Moon Potential . . . 107
4.3 Newton–Krylov Framework . . . 110
4.3.1 Newton–Krylov for Mean-Field Equations . . . 111
4.3.1.1 Bacterial chemotaxis model . . . 113
4.3.1.2 Economic Agents Model . . . 114
4.4 From Particles to Smooth Representations . . . 116
4.4.1 The ICDF-to-ICDF Timestepper . . . 117
4.4.2 Three Numerical Illustrations . . . 119
4.4.2.1 The Bimodal Distribution . . . 120
4.4.2.2 Bacterial Chemotaxis . . . 121
4.4.2.3 Economic Agents . . . 122
4.5 Extending Smooth Representations to Multiple Dimensions . . . 124
4.5.1 The CDF-to-CDF Timestepper . . . 126
4.5.2 The Sliced Wasserstein Timestepper . . . 130
4.6 Discussion and Outlook . . . 135

Chapter 5 Image Interpolation Using Wasserstein Geodesics . . . 138
5.1 Introduction . . . 139
5.2 Preliminaries and theoretical motivation . . . 141
5.2.1 Diffusion models, stable diffusion, and CLIP . . . 141

5.2.2 Permutation invariance . . . 143
5.2.3 Optimal transport and Wasserstein space . . . 145
5.3 Geometry-informed Image Interpolation Approach . . . 147
5.4 Experimental Design and Results . . . 149
5.4.1 Experimental setup . . . 149
5.4.2 Dataset . . . 150
5.4.3 Results . . . 151
5.5 Conclusions and Future Directions . . . 153
Bibliography . . . 156

LIST OF FIGURES

Figure 2.1 Example of the difference between the optimal transport map and the particle-wise map. 34

Figure 2.2 Vector field approximation error $\|\mathbf{v}_t - \mathbf{v}_t^{h,N}\|^2_{L^2(\mu_t^N)}$ versus Euler step approximation error $W_2(\mu_{t+H}^N, \widetilde{\mu}_{t+H}^N)^2$ for Brownian motion with $t = 1$ and $H = 99$ (h and N vary). 35

Figure 2.3 Results of vector field error computations for Brownian motion and comparison with an estimated error using the difference between consecutive approximations. 36

Figure 2.4 Histograms of bacteria locations at various times t from $t = 0$ to $t = 4000$ for the control simulation of chemotaxis. 43

Figure 2.5 Histograms of bacteria locations after first two Euler steps (bottom) compared with corresponding times from control simulation (top). . 45

Figure 2.6 Histograms of bacteria locations after final two Euler steps (bottom) compared with corresponding times from control simulation (top). . 45

Figure 2.7 Comparisons between control distributions and approximations. . . 51

Figure 2.8 Histograms of particle locations at various times t from $t = 0$ to $t = 2$ for the control simulation of the Burgers' model. 56

Figure 2.9 Histograms of particle locations after first two Euler steps (bottom) compared with corresponding times from control simulation (top). . 57

Figure 2.10 Histograms of particle locations after final two Euler steps (bottom) compared with corresponding times from control simulation (top). . 57

Figure 2.11 Scatter plots of particle locations at various times t from $t = 0$ to $t = 2$ for the control simulation of the half-moon Langevin dynamics model. 61

Figure 2.12 Scatter plots of particle locations after first two Euler steps (bottom) compared with corresponding times from control simulation (top). . 61

Figure 2.13 Comparisons between control distributions and approximations. The blue is our algorithm, and the orange is using the JKOnet* method for learning and simulating an SDE. 62

Figure 3.1 Average total error $E_1^{h,N}$ for different values of h, N, averaged over 32 independent simulations. Actual values of the model error $M_1^{h,N}$ and the statistical error $S_1^{h,N}$ are shown in gray, for comparison. . . . 71

Figure 3.2 Average sum of model and statistical error $M_1^{h,N} + S_1^{h,N}$ for different values of h, N, averaged over 32 independent simulations. Actual values of the model error $M_1^{h,N}$ and the statistical error $S_1^{h,N}$ are shown in gray, for comparison. 71
Figure 3.3 Average absolute value of the inner product $|\langle \mathbf{m}, \mathbf{s} \rangle|_{L^2(\mu_1^N)}$ for different values of h, N, averaged over 32 independent simulations. 72
Figure 3.4 Average value of the model error $M_1^{h,N}$ for different values of h, N, averaged over 32 independent samples. 74
Figure 3.5 Average value of the optimal transport estimation error $h^2 S_1^{h,N}$ for different values of h, N, averaged over 32 independent trials per pair, in the case of independent samples from $\mu_1 = \mathcal{N}(0, 1)$ and $\mu_{1+h} = \mathcal{N}(0, 1 + h)$. 78
Figure 3.6 Average value of the statistical error $S_1^{h,N}$ for different values of h, N, averaged over 32 independent trials per pair, in the case of independent samples from $\mu_1 = \mathcal{N}(0, 1)$ and $\mu_{1+h} = \mathcal{N}(0, 1 + h)$. . . . 79
Figure 3.7 Average value of the optimal transport estimation error $h^2 S_1^{h,N}$ for different values of h, N, averaged over 32 independent trials per pair, in the case of an Euler-Maruyama particle timestepper. 81
Figure 3.8 Average value of the statistical error $S_1^{h,N}$ for different values of h, N, averaged over 32 independent trials per pair, in the case of an Euler-Maruyama particle timestepper. 82
Figure 3.9 Average value of the consecutive error $\hat{E}_1^{h,N}$ for different values of h, N, averaged over 32 independent trials per pair, in the case of an Euler-Maruyama particle timestepper. 86
Figure 3.10 Average value of the consecutive statistical error estimate $\hat{S}_1^{h,N}$ compared with the average of two statistical errors $\frac{1}{2}(S_1^{h,N} + S_1^{2h,N})$ for different values of h, N, averaged over 32 independent trials per pair, in the case of an Euler-Maruyama particle timestepper. 88

Figure 4.1 Numerical results for the Wasserstein-Adam optimizer on the chemotaxis model. 106
Figure 4.2 Numerical results for the Wasserstein-Adam optimizer on the economic agents model. 107
Figure 4.3 A color plot of the half-moon potential, see equation (4.2.4.2) for more details. 109
Figure 4.4 Numerical results for the Wasserstein-Adam optimizer on the half-moon potential. 110
Figure 4.5 (Left) Steady-state distribution of the Keller-Segel model: Newton–Krylov (blue) vs. analytic solution (dashed orange). (Right) Newton–Krylov residual per iteration. 114
Figure 4.6 Steady-state distribution from Newton–Krylov versus time evolution of stochastic agents. 115

Figure 4.7 Numerical results for the Newton–Krylov optimizer for bacterial chemotaxis. 116
Figure 4.8 Schematic of the effective ICDF–to-ICDF timestepper. 119
Figure 4.9 (Left) Initial ICDF (green), true steady-state (orange) and steady-state ICDF computed by Newton–Krylov (dashed blue). (Right) Particles sampled from the Newton–Krylov ICDF (blue) and corresponding steady-state bimodal density. 121
Figure 4.10 (Left) Initial ICDF(green), true steady-state (orange) and steady-state ICDF computed by Newton–Krylov (dashed blue). (Right) Particles sampled from the Newton–Krylov ICDF (blue) and corresponding steady-state density (4.3.1.4) in orange. 122
Figure 4.11 (Left) Initial ICDF (green), true steady-state (orange) and steady-state ICDF computed by Newton–Krylov (dashed blue). (Right) Particles sampled from the Newton–Krylov ICDF (blue) and steady-state density corresponding to the mean-field PDE. 123
Figure 4.12 Newton–Krylov ICDF residual $\Psi\left((F^{-1})^{(k)}\right)$ per iteration for the economics agents model (blue); Second-order convergence rate (green). 124
Figure 4.13 Schematic of the effective two-dimensional CDF-to-CDF timestepper. 129
Figure 4.14 Histogram heatmaps of particle density: (left) initial Gaussian condition; (right) distribution after Newton–Krylov optimization using the 2D CDF smooth representation. 131
Figure 4.15 Newton–Krylov loss $\left\|\Psi\left(F^{(k)}_{X,Y}\right)\right\|$ per iteration k. The loss decreases up until a noise level induced by the finite number of particles. . . . 132
Figure 4.16 Schematic of the effective sliced Wasserstein to sliced Wasserstein timestepper. 134
Figure 4.17 (left) Loss $\left\|\Psi(F^{(k)}_{X,Y}\right\|$ per iteration; (right) Histogram heatmap of resulting steady-state particle density of the Newton–Krylov method applied to the Sliced Wasserstein representation. 135

Figure 5.1 Embedding interpolation workflow where SD indicates use of the Stable Diffusion model to produce the associated image. 149
Figure 5.2 Image interpolations for each method across four selected prompt pairs of varying similarity. 151
Figure 5.3 PPL (left) and coupling cost (right) by interpolation method and similarity group. 153

LIST OF TABLES

Table 2.1 Parameter definitions. 39
Table 2.2 Parameters for chemotaxis micro-scale simulation. 42
Table 2.3 Parameters for approximation algorithm with chemotaxis micro-scale simulation. 44
Table 2.4 Parameters of different re-initialization schemes. 53
Table 2.5 Parameters for Burgers' model micro-scale simulation. 55
Table 2.6 Additional parameters for approximation algorithm with Burgers' model micro-scale simulation. 55
Table 2.7 Parameters for 2-D half-moon micro-scale simulation. 60
Table 2.8 Additional parameters for approximation algorithm with chemotaxis micro-scale simulation. 60

Table 5.1 Significance of p-values for the Wilcoxon test comparing the median of the PPL scores. ($p < 0.05^{*}$, $p < 0.01^{**}$, $p < 0.001^{***}$) 153

ACKNOWLEDGEMENTS

If I were to give appropriate thanks to everyone who has helped shape my mathematical journey, then this section would need to be longer than the rest of the dissertation. There are too many people to individually name and their impacts too great to adequately express, so what follows is an incomplete and insufficient accounting of the people I most want to thank for helping me reach the end of this long and wonderful educational road.

First and foremost is my advisor Alex Cloninger. Alex, you're not going to like this, but I'm going to say some nice things about you. I am so grateful for your mentorship, guidance, support, and patience over the past five years. You have been amazingly kind while still pushing me to give more than I thought I had, and I would not be where I am today without you. You have also given me so many amazing opportunities and have advocated for me in ways I may never fully understand. Most importantly, you have taught me not just how to do good research but how to be a good *researcher*, managing different projects and understanding how my work fits into the broader research story. I am sad to not be under your wing anymore, but I trust that you've readied me for what's next. Thank you for everything, Alex. It really has been a blast.

I also feel lucky to have coauthored papers with many other wonderful and brilliant people. Thank you to my academic brother Varun Khurana, for introducing me to optimal transport and for shepherding me during my (bizarre) second year; to Vaki Nikitopoulos, for helping me navigate much of grad school, including my first paper; to Yannis Kevrekidis, for your uncanny intuition, which led to many of the core ideas in this dissertation; to Hannes Vandecasteele and Joon Lee, for generously sharing your expertise and patiently working through details with me; to Tegan Emerson, for encouraging me to intern at PNNL last summer and being so supportive of me and my career; to Luke Durell and Javi Flores, for being lovely internship mentors and collaborators; to James Murphy, for many exciting ideas and directions, only one of which we've explored; and to Josh Lentz and Jun Linwu,

for allowing me to collaborate with you on your projects. Thank you all for teaching me so much about so much. I look forward to more collaborations and opportunities to learn from and with all of you.

Of course, none of this work would have happened at all if Rayan Saab and Ioana Dumitriu had not talked me into coming to UCSD all those years ago. Rayan, you have always been so generous with your time, especially in the early days of my grad school journey. I still remember you sending me a Zoom message at the open house inviting me to chat more about math for data science. As it turns out, that's exactly what I wanted to do in grad school, I just didn't know about it yet, and that first conversation with you was what set me on the path. I'm bummed we never did write a paper together (yet, at least!), but I am still so grateful for your informal mentorship. Ioana, you have been a sounding board and a voice of reason on so many things over the past five years. You have always given me good advice, but the best advice came in that short call when you confidently told me that this was the right place for me. You and Rayan were both spot on — San Diego has been spectacular. Thank you both for selling me on it. I'm so glad you did.

Before UCSD, there were many wonderful faculty at Northwestern who kindled my interest in mathematics and turned it into a deep love for the field. Particular thanks to Santiago Cañez, Aaron Peterson, John Alongi, Maria Nastasescu, Julian Gold, and especially my undergrad thesis advisor Ezra Getzler for all encouraging me to pursue grad school. Ezra, I know my mathematical direction has changed dramatically over the last five years, but I hope the occasional references to differential geometry still make you smile (who would've guessed the title of my dissertation would have the word "manifold" in it?). Even though my tastes have changed, I can see that you all prepared me for grad school in ways I didn't fully understand at the time, and I'm deeply grateful for that.

Going back even further, I was lucky to have elementary and high school teachers who nurtured and encouraged my curiosity — mathematical and otherwise — from a young age. I am particularly grateful for Stacy Konger, Nick Ruedig, Steve Omachi,

Dan Janeiro, Vu Nguyen, Tom Kelliher, Eileen Lesch, and Sunny Lenhard for patiently entertaining my many questions and encouraging me to keep asking more. I am also hugely indebted to all of the CBA faculty, but especially Matt Reagan, Bill Frake, Craig Dashkavich, Jeff Matson, Sean Nunan, and Cathy Carroll, for taking my natural curiosity and channeling it into a real interest in academic work. Thank you all for giving me permission to stay curious and to never stop learning.

Outside of work, there are so many people who have made UCSD an amazing place to spend the last five years. Thank you to my academic brothers Varun Khurana, Sawyer Robertson, Dhruv Kohli, and Yiming Zhang for their friendship and for many interesting group discussions, and to Rob Webber, Lijun Ding, Rahul Parhi, Erin George, and everyone else who makes the MINDS seminar something to look forward to every Friday. Sharing ideas with you all has been a real treat. Thanks also to all the grad students who work hard to make the math grad community so special: everyone I've worked with on MGSC, everyone in AWM, everyone who organizes teas and social gatherings, the many Drs. Thought, and so many more. This community has been an invaluable source of mentorship and friendship for me, and I'm so grateful for everyone who provides opportunities for that connection.

Beyond the walls of AP&M and HSS, I owe an enormous and loving thank you to everyone who has kept me sane over the last five years. Thank you to the people of the San Diego Roundnet Club, the Andariegos Baseball Team, and the Mesa Rim climbing gyms for welcoming me into their communities and giving me places to get away from work. Thank you to my dear friends Garrett Pollack, Matthew Price, Adam Downing, Annika Pracher, Matt Kelly, Manny Lazzaro, and Tyler Gentile for your constant support, even from many miles away. Most importantly, thank you to all my friends in San Diego, especially Nathan Wenger, for being such a loyal and dependable friend; Nate Conlon, Johna Massaquoi, Finn Southerland, Lizzy Coda, Freddy Garcia, Delmy Santos, and JD Flynn, for plenty of laughs and antics; and Clay Adams, for teaching me more than I ever thought possible.

Above all, I cannot possibly thank my family enough for everything they've given me. Mom, Dad, Gracie, saying "I could not have gotten here without you" feels wildly inadequate. Your constant support has meant everything to me and has opened up so many wonderful opportunities, and you've always encouraged me to do what I'm most excited about. I know I could never pay back everything you all have done for me; I just hope you know how much I appreciate it. I love you all so much.

Finally, there's one last legalistic point. It is important to acknowledge that most of my dissertation comes from three papers that have been submitted or accepted for publication. Specifically:

Chapter 1, in part, and Chapter 2, in full, are reprints of the material in [KNK+25]: Nicholas Karris, Evangelos A. Nikitopoulos, Ioannis G. Kevrekidis, Seungjoon Lee, and Alexander Cloninger. Using linearized optimal transport to predict the evolution of stochastic particle systems. Preprint, arXiv:2408.01857, 2025. The dissertation author is the primary author and investigator of this paper, which is under revision for publication in the SIAM Journal on Applied Dynamical Systems.

Chapter 4, in full, is a reprint of material in [VKCK25]: Hannes Vandecasteele, Nicholas Karris, Alexander Cloninger, and Ioannis G. Kevrekidis. Optimal transport, timesteppers, Newton-Krylov methods and steady states of collective particle dynamics. Preprint, arXiv:2512.24567, 2025. The dissertation author is a primary author and investigator of this paper, which is currently under revision for publication in the Journal of Computational Physics.

Chapter 5, in full, is a reprint of the material in [KDFE26]: Nicholas Karris, Luke Durell, Javier Flores, and Tegan Emerson. Which way from B to A: The role of embedding geometry in image interpolation for Stable Diffusion. In *Proceedings of the 1st Conference on Topology, Algebra, and Geometry in Data Science (TAG-DS 2025)*, volume 321 of *Proceedings of Machine Learning Research*, pages 114–125. PMLR, 2026. The dissertation author is the primary author and investigator of this work, which was conducted under the

Laboratory Directed Research and Development Program at PNNL, a multi-program national laboratory operated by Battelle for the U.S. Department of Energy under contract DE-AC05-76RL01830.

VITA

2021	Bachelor of Arts, Northwestern University
2024	Master of Arts, University of California San Diego
2026	Doctor of Philosophy, University of California San Diego

PUBLICATIONS

N. Karris, L. Durell, J. Flores, T. Emerson. *Which Way from B to A: The Role of Embedding Geometry in Image Interpolation for Stable Diffusion.* Published in the Proceedings of the 1st Conference on Topology, Algebra, and Geometry in Data Science (TAG-DS 2025), *Proceedings of Machine Learning Research*, 2026.

H. Vandecasteele, **N. Karris**, A. Cloninger, I. Kevrekidis. *Optimal Transport, Timesteppers, Newton–Krylov Methods and Steady States of Collective Particle Dynamics.* Preprint, arXiv:2512.24567, 2025.

J. Lentz, **N. Karris**, A. Cloninger, J. Murphy. *Unbalanced Optimal Transport Dictionary Learning for Unsupervised Hyperspectral Image Clustering.* Published in the Proceedings of the 15th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS 2025), IEEE, 2025.

N. Karris, E. Nikitopoulos, I. Kevrekidis, S. Lee, A. Cloninger. *Using Linearized Optimal Transport to Predict the Evolution of Stochastic Particle Systems.* Revised and Resubmitted, SIAM Journal on Applied Dynamical Systems. Preprint, arXiv:2408.01857, 2025.

J. Linwu, V. Khurana, **N. Karris**, A. Cloninger. *Linearized Optimal Transport pyLOT Library: A Toolkit for Machine Learning on Point Clouds.* Preprint, arXiv:2502.03439, 2025.

ABSTRACT OF THE DISSERTATION

Numerical Analysis of Curves on the Wasserstein Manifold

by

Nicholas Karris

Doctor of Philosophy in Mathematics

University of California San Diego, 2026

Professor Alexander Cloninger, Chair

Many modern problems in data science involve data that is fundamentally measure-valued. In these situations, natural questions arise about how one should perform standard tasks from machine learning and numerical analysis. The tools of optimal transport point to intuitive analogs of these tasks by viewing the measures themselves as single points on the Wasserstein manifold. Then, by computing optimal transport maps from a fixed reference, we can embed each measure into a common linear space (the tangent space of the Wasserstein manifold at the fixed reference), which allows us to apply out-of-the-box methods on the embeddings. When the data is static, these methods can be used to build classifiers, generate new measures from particular classes, efficiently compute clusters, and produce visualizations via dimension reduction. In this dissertation, we extend this technique of "linearized optimal transport" to the case where our measure-valued data is

evolving in time. In this case, we interpret our data as a curve on the Wasserstein manifold, and we use optimal transport to compute tangent vectors to this curve. These tangent vectors allow us to produce geodesics in specified directions, which can be used both for interpolation between measures and for extrapolation to predict future evolution. We rigorously develop this extrapolation as an analog of Euler's method in Euclidean space, and we use example particle simulators to demonstrate how this approach can accurately approximate (sufficiently nice) curves of evolving measures. We further develop the analog of Newton's method for finding steady states of particle models, connect it to our Euler timestepper, and demonstrate how steady-state distributions can be efficiently computed using Newton–Krylov methods. As a final application, we observe that textual prompt embeddings in the Stable Diffusion pipeline can be treated as discrete uniform measures, and we show that using Wasserstein geodesics to interpolate between two prompts gives a more "natural" way to interpolate between output images.

Chapter 1

Introduction

There are many situations in which one would like to apply standard machine learning or numerical analysis techniques, except the available data are measures, not points in Euclidean space. One motivating example is a collection of particles evolving over time according to some micro-scale simulator (e.g., as in a simulation of some biological process). Many standard approaches to understanding such systems attempt to learn the simulator as an operator, for example by tracking individual particles and interpolating the results [LJP+21, GBYK23, GKSK22, Mha23, KLM24]. These approaches inherently treat the collection as an *ordered* vector of particle locations. However, if the simulator is random, then this way of vectorizing the particles leads to substantial statistical noise because individual particles are evolving stochastically. To avoid this issue, we can instead interpret the collection of particle locations as an *unordered* distribution and ask a more general question about how the distribution as a whole is evolving [SGOK05, LPSK23]. In this way, we treat the collection of particles as a (discrete, finite) measure rather than a vector in Euclidean space.

In recent years, "linearized optimal transport" has emerged as an important tool for performing standard data science tasks on static measure-valued data [WSB+13, PKKR18, MC23, Gig11, MDC20, LKKC25]. By computing optimal transport maps from a fixed

reference, we can embed each individual measure into a common linear space, which allows us to apply out-of-the-box machine learning methods like clustering, classification, and dimension reduction. In the setting where our measures are evolving in time, we can use the same LOT approach to unlock standard numerical analysis tools for estimating derivatives, approximating curves with geodesics, and finding steady states. Optimal transport thus gives us a way to make sense of the evolution of a *distribution* of particles as a collective whole, rather than as a list of particles each evolving individually.

To motivate and understand our methods, we turn to the geometry of the Wasserstein manifold. Formally, we can view $\mathcal{P}_2(\mathbb{R}^d)$ (the space of probability measures on $\mathbb{R}^d$ with finite second moment) as a Riemannian manifold with metric given by the Wasserstein distance W_2 [Ott01, AGS08]. The existence of a tangent space, and more importantly the ability to access the logarithm map via optimal transport, is what motivates many of the ideas underlying LOT. In particular, one can now view a time-evolving measure as a curve on the Wasserstein manifold, and the correct derivative object is the tangent vector to that curve, which we can compute by lifting the curve to the tangent space and computing a derivative, just as we would on any manifold. With this formalism, many standard methods from numerical analysis now have natural analogs, like Euler's method for building timesteppers [KNK+25] or Newton's method for finding steady states [VKCK25]. Similarly, we can use the optimal transport map between two measures to construct a geodesic between them by linearly interpolating in the tangent space and then projecting back to the Wasserstein manifold using the exponential map.

1.1 Preliminaries

1.1.1 Optimal Transport Background

Let $\mathcal{P}(\mathbb{R}^d)$ be the set of Borel probability measures on $\mathbb{R}^d$, and let $\mathcal{P}_2(\mathbb{R}^d)$ be the set of measures $\mu \in \mathcal{P}(\mathbb{R}^d)$ such that $\int_{\mathbb{R}^d} \|x\|_2^2 \, d\mu(x) < \infty$. For a measurable function $T : \mathbb{R}^d \to \mathbb{R}^d$

and a measure $\sigma \in \mathcal{P}(\mathbb{R}^d)$, the pushforward of σ by T is the measure $T_\#\sigma$ defined by

$$(T_\#\sigma)(A) := \sigma(T^{-1}(A))$$

where $A \subset \mathbb{R}^d$ is measurable and $T^{-1}(A)$ is the preimage of A under T.

Given $\sigma, \mu \in \mathcal{P}_2(\mathbb{R}^d)$, a classical problem (the "Monge Problem") is to seek an "optimal" map T transporting σ to μ in the sense that $T_\#\sigma = \mu$. However, there does not always exist such a map, which leads to the more general "Kantorovich formulation":

Definition 1.1.1.1. *For $\sigma, \mu \in \mathcal{P}_2(\mathbb{R}^d)$, the (2-) Wasserstein distance between σ and μ is*

$$W_2(\sigma, \mu) := \min_{\gamma \in \Gamma_{\sigma,\mu}} \left(\int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 \, d\gamma(x, y) \right)^{\frac{1}{2}}, \tag{1.1.1.2}$$

where $\Gamma_{\sigma,\nu}$ is the set of all couplings between σ and μ, i.e., the set of all $\gamma \in \mathcal{P}_2(\mathbb{R}^d \times \mathbb{R}^d)$ such that

$$\gamma(A \times \mathbb{R}^d) = \sigma(A), \qquad \gamma(\mathbb{R}^d \times A) = \mu(A)$$

for all measurable $A \subseteq \mathbb{R}^d$.

The coupling γ that satisfies (1.1.1.2) *is called the "optimal coupling" or the "optimal plan."*

It is well known that W_2 defines a metric on $\mathcal{P}_2(\mathbb{R}^d)$ (see, e.g., [PC19]).

We will only refer to "optimal couplings" a handful of times in this dissertation (and only to handle technical details). Instead, we usually deal with optimal "maps" as in the Monge problem, and Brenier's Theorem is the standard result that guarantees the existence of such maps under more hypotheses:

Theorem 1.1.1.3 (Brenier [Bre91])**.** *Let $\sigma, \mu \in \mathcal{P}_2(\mathbb{R}^d)$, and suppose σ has a density with respect to the Lebesgue measure $\mathcal{L}^d$. The optimal coupling γ that satisfies* (1.1.1.2) *is unique, and there*

exists a σ-a.e. unique map $T : \mathbb{R}^d \to \mathbb{R}^d$ such that $\gamma = (\mathrm{id}, T)_{\#}\sigma$. That is,

$$W_2(\sigma, \mu) = \left(\int_{\mathbb{R}^d} \|x - T(x)\|^2 \, d\mu(x) \right)^{\frac{1}{2}}.$$

Moreover, there exists a convex function $\phi : \mathbb{R}^d \to \mathbb{R}$, σ-a.e. unique up to an additive constant, such that $T = \nabla\phi$.

Definition 1.1.1.4. *The map T given in Theorem 1.1.1.3 is the optimal transport map from σ to μ, and we denote it T_σ^μ.*

Optimal transport is also possible between discrete measures. In general, there is not an optimal transport map between discrete measures, but the following result guarantees that one exists in the case of uniform measures on the same finite number of points.

Proposition 1.1.1.5 (Proposition 2.1 in [PC19])**.** *If $\sigma = \frac{1}{N}\sum_{i=1}^N \delta_{x_i}$ and $\mu = \frac{1}{N}\sum_{i=1}^N \delta_{y_i}$ are uniform measures on the same number of distinct points, then there exists an optimal transport map T_σ^μ in the sense that*

$$W_2(\sigma, \mu) = \left(\frac{1}{N} \sum_{i=1}^N \|x_i - T_\sigma^\mu(x_i)\|^2 \right)^{\frac{1}{2}}$$

Moreover, if T_σ^μ is such a map, then there exists a permutation $\tau \in S_N$ (the symmetric group on N elements) such that $T_\sigma^\mu(x_i) = y_{\tau(i)}$ for all $i = 1, \ldots, N$.

Unlike in Theorem 1.1.1.3, optimal transport maps as in Proposition 1.1.1.5 are not necessarily unique, even σ-a.e. In this discrete case, the notation T_σ^μ will denote a particular choice of an optimal map.

1.1.2 Tangent Fields and Linearized Optimal Transport

As Otto first observed in [Ott01], $\mathcal{P}_2(\mathbb{R}^d)$ has a formal infinite-dimensional Riemannian manifold structure. Because of this, the language and philosophy of differential geometry are often used in optimal transport theory and its applications. The key examples

of interest to us and, more generally, the subject of linearized optimal transport (LOT) are the descriptions of the "tangent space(s)" and "exponential map" of the "Wasserstein manifold." Below, we present the precise definition of the tangent space and a result that justifies its definition and underlies the philosophy and results of the dissertation.

Definition 1.1.2.1 ((8.0.2) in [AGS08]). *For $\sigma \in \mathcal{P}_2(\mathbb{R}^d)$, define the tangent space $T_\sigma \mathbb{W}_2$ of the "Wasserstein manifold" $\mathbb{W}_2 := (\mathcal{P}_2(\mathbb{R}^d), W_2)$ at σ to be the closure in $L^2(\mathbb{R}^d, \sigma; \mathbb{R}^d)$ of $\{\nabla\phi : \phi \in C_c^\infty(\mathbb{R}^d)\}$.*

Theorem 1.1.2.2 (Theorem 8.3.1, Proposition 8.4.5, and Proposition 8.4.6 in [AGS08]). *Let $I \subseteq \mathbb{R}$ be an open interval, let $\mu : I \to \mathcal{P}_2(\mathbb{R}^d)$ be an absolutely continuous curve in $\mathbb{W}_2$ (written $t \mapsto \mu_t$), and let $|\mu'|$ be the metric derivative of μ.*

(i) *There exists a Borel vector field $I \times \mathbb{R}^d \ni (t, x) \mapsto \mathbf{v}(t, x) = \mathbf{v}_t(x) \in \mathbb{R}^d$ such that*

$$\|\mathbf{v}_t\|_{L^2(\mu_t)} := \left(\int_{\mathbb{R}^d} \|\mathbf{v}_t(x)\|_2^2 \, d\mu_t(x) \right)^{\frac{1}{2}} \leq |\mu'|(t) \quad \text{for } \mathcal{L}^1\text{-a.e. } t \in I \tag{1.1.2.3}$$

and the continuity equation

$$\partial_t \mu_t + \nabla \cdot (\mathbf{v}_t \mu_t) = 0 \tag{1.1.2.4}$$

holds in the sense of distributions, i.e.,

$$\int_I \int_{\mathbb{R}^d} \left[\partial_t \phi(x, t) + \langle \mathbf{v}_t(x), \nabla_x \phi(x, t) \rangle \right] d\mu_t(x) dt = 0 \quad \text{for all } \phi \in C_c^\infty(\mathbb{R}^d \times I). \tag{1.1.2.5}$$

The map $t \mapsto \mathbf{v}_t$ is $\mathcal{L}^1$-a.e. uniquely determined by (1.1.2.3) *and* (1.1.2.4) *and is called the* tangent field *of μ.*

(ii) *If $\mathbf{v} \colon I \times \mathbb{R}^d \to \mathbb{R}^d$ is any Borel vector field satisfying* (1.1.2.4)*, then* (1.1.2.3) *is satisfied if and only if $\mathbf{v}_t \in T_{\mu_t} \mathbb{W}_2$ for $\mathcal{L}^1$-a.e. $t \in I$.*

(iii) *If* $\mathbf{v}$ *is the tangent field of* μ*, then*

$$\lim_{h\to 0} \frac{W_2(\mu_{t+h}, (\mathrm{id} + h\mathbf{v}_t)_{\#}\mu_t)}{|h|} = 0 \quad \text{for } \mathcal{L}^1\text{-a.e. } t \in I. \tag{1.1.2.6}$$

In particular, if $\mu_t \ll \mathcal{L}^d$ *for all* $t \in I$*, then for* $\mathcal{L}^1$*-a.e.* $t \in I$*,*

$$\lim_{h\to 0} \frac{1}{h}(T_{\mu_t}^{\mu_{t+h}} - \mathrm{id}) = \mathbf{v}_t \quad \text{in } L^2(\mathbb{R}^d, \mu_t; \mathbb{R}^d), \tag{1.1.2.7}$$

where $T_{\mu_t}^{\mu_{t+h}}$ *is the optimal transport map from* μ_t *to* μ_{t+h}*.*

Remark 1.1.2.8. *For a general choice of* $\sigma \in \mathcal{P}_2(\mathbb{R}^d)$*, the best one can hope for is that the map* $t \mapsto T_\sigma^{\mu_t} \in L^2(\mathbb{R}^d, \sigma; \mathbb{R}^d)$ *is at most* $\frac{1}{2}$*-Hölder continuous, even if the curve* μ *is Lipschitz [Gig11, MDC20]. However,* (1.1.2.7) *says that by fixing* t *and choosing* $\sigma = \mu_t$*, the curve* $h \mapsto T_{\mu_t}^{\mu_{t+h}}$ *becomes "differentiable" at* $h = 0$*.*

The exponential map can now be computed by finding the tangent field of a geodesic between two points $\mu_0, \mu_1 \in \mathcal{P}_2(\mathbb{R}^d)$. One can show (Section 7.2 of [AGS08]) that if $T_{\mu_0}^{\mu_1}$ is an optimal transport map from μ_0 to μ_1, then the curve

$$t \mapsto \mu_t := ((1-t)\,\mathrm{id} + tT_{\mu_0}^{\mu_1})_{\#}\mu_0$$

is a (constant-speed) geodesic in W_2 from μ_0 to μ_1. Furthermore, the vector field

$$\mathbf{v}_t(x) := (T_{\mu_0}^{\mu_1} - \mathrm{id})\left(((1-t)\,\mathrm{id} + tT_{\mu_0}^{\mu_1})^{-1}(x)\right)$$

is the tangent field of the geodesic μ (Section 5.4 of [San15]). Therefore, $\mathbf{v}_0 = T_{\mu_0}^{\mu_1} - \mathrm{id}$ is the "tangent vector" to μ at $t = 0$. We conclude that the logarithm map, i.e., the inverse of the exponential map, at μ_0 is the map

$$\log_{\mu_0}^{\mathbb{W}_2} : \mathbb{W}_2 \to T_{\mu_0}\mathbb{W}_2, \qquad \log_{\mu_0}^{\mathbb{W}_2}(\mu_1) = \mathbf{v}_0 = T_{\mu_0}^{\mu_1} - \mathrm{id}\,.$$

Since $(\mathbf{v}_0 + \mathrm{id})_{\#}\mu_0 = (T_{\mu_0}^{\mu_1})_{\#}\mu_0 = \mu_1$, it follows that the exponential map at μ_0 is the map

$$\exp_{\mu_0}^{\mathbb{W}_2} \colon T_{\mu_0}\mathbb{W}_2 \to \mathbb{W}_2, \qquad \exp_{\mu_0}^{\mathbb{W}_2}(\mathbf{v}) = (\mathbf{v} + \mathrm{id})_{\#}\mu_0.$$

(Please see [AG13, Lot08] for additional discussion.) In particular, if μ is a general absolutely continuous curve with velocity field $\mathbf{v}$, then Theorem 1.1.2.2(iii) says

$$\mu_{t+h} \approx (\mathrm{id} + h\mathbf{v}_t)_{\#}\mu_t = \exp_{\mu_t}^{\mathbb{W}_2}(h\mathbf{v}_t)$$

for small h, which makes sense from a differential geometric point of view.

An important consequence of the previous paragraph's description of the Wasserstein manifold's logarithm map is that $\mu_1 \mapsto T_{\mu_0}^{\mu_1} - \mathrm{id}$ "linearizes" $\mathcal{P}_2(\mathbb{R}^d)$; hence the L in LOT. Specifically, for a fixed reference measure $\sigma \in \mathcal{P}_2(\mathbb{R}^d)$, we can use the map $\mu \mapsto T_\sigma^\mu - \mathrm{id}$ to "embed" the space $\mathcal{P}_2(\mathbb{R}^d)$ into the linear space $L^2(\mathbb{R}^d, \sigma; \mathbb{R}^d)$. There are many advantages to this linear embedding; see, for example, [WSB+13, MC23] for computational speed-up and supervised learning and [CHKM25] for performing PCA.

1.2 Overview

1.2.1 Euler's Method for Predicting Measure Evolution

Chapter 2 is based on [KNK+25], in which we formally develop the measure-valued analog of Euler's method, i.e., Euler's method on the Wasserstein manifold. We prove (under hypotheses about the regularity of the curve) that the method gives a first-order accurate approximation of the true curve with error bounds similar to those of ordinary Euler's method. We also use this intuition to build a macro-scale "timestepper" from a micro-scale one [AK21, FVG+25, GGK09, GKT02, KGH+03, LPSK23, LKGK03]. If we have a model that simulates a collection of particles forward in time for a small time step,

then we can use the optimal transport map from the original to the updated distribution to compute a finite-difference approximation $\mathbf{v}_t^h$ of the true tangent field $\mathbf{v}_t$. We use $\mathbf{v}_t^h$ as a surrogate for $\mathbf{v}_t$ in Euler's method, which gives an Euler-type method that takes a much larger time step than one step of the micro-scale model. This "macro-scale" timestepper gives an accurate way to predict the evolution of the system a long time in the future without needing to run the micro-scale model for the equivalent duration. The power of our approach is that it requires only *local* data to take a large time step, whereas other similar approaches focus on learning the action of the micro-scale model *globally* [LJP+21, GBYK23, GKSK22, Mha23, KLM24, TLGD24, BR25]. We experimentally demonstrate that our method outperforms other approaches for predicting the evolution of example particle systems (a bacterial chemotaxis model, a fluid flow model, and Langevin dynamics) when we have limited data from the micro-scale model. Thus, we have built a macro-scale timestepper which can now be used to do a number of system-level tasks like control and identification, analyzing stability, and approximating steady states [SPK03, FVG+25, GKT02].

Chapter 2, in full, is a reprint of material in [KNK+25]: Nicholas Karris, Evangelos A. Nikitopoulos, Ioannis G. Kevrekidis, Seungjoon Lee, and Alexander Cloninger. Using linearized optimal transport to predict the evolution of stochastic particle systems. Preprint, arXiv:2408.01857, 2025. The dissertation author is the primary author and investigator of this paper, which is under revision for publication in the SIAM Journal on Applied Dynamical Systems.

1.2.2 Choosing an Appropriate Time Step

In Chapter 3, we explore in detail the subtleties of choosing an appropriate time step for our Euler-type method developed in Chapter 2. We detail heuristic arguments for why the error intrinsic to the finite difference approximation and the error intrinsic to the statistical estimate of the true optimal transport map are "orthogonal" and so each

contribute independently to the overall estimation error. We then give detailed analysis for each source of error. We demonstrate that the "model error" coming from the finite difference estimate decreases with the time step h while the "statistical error" coming from the empirical optimal transport map increases with h. This tension results in an optimal choice of h that occurs near where these two quantities are of the same order. We then investigate each of the model error and statistical error in depth for the standard example of Gaussian diffusion, which is represented by the curve of measures $\mu_t = \mathcal{N}(0, t)$. This example is nice because we can explicitly calculate the underlying velocity field $\mathbf{v}_t$, which allows us to directly compute the velocity field approximation error. We give theoretical justification for some of the empirically observed phenomena and discuss how these phenomena impact the optimal choice of step size. Finally, we introduce a heuristic estimate that can be used to choose an optimal step size even when the true underlying dynamics (e.g., the true velocity field $\mathbf{v}_t$) are unknown, as is often the case in our setting with timesteppers.

1.2.3 Newton's Method for Approximating Steady States

Chapter 4 is based on [VKCK25] and extends the measure-valued analogs further than first-order methods. By taking more than one step with a micro-scale model, we can approximate the tangent vector at multiple initial measures, which then allows us to estimate a "second derivative." This opens up the possibility of developing analogous second-order methods for the equivalent of ODEs on the Wasserstein manifold. In Chapter 4, we present a Newton-type method that uses our optimal transport–based timestepper from Chapter 2 to approximate distributions which are relative steady states of the micro-scale model. However, since we only have access to a micro-scale timestepper, we are not able to approximate the Jacobian of the system itself, so we combine our method with a Krylov-subspace approach for solving a linear system [FEC+24, KKQ04, QEKK06]. With this approach, we only need to know the action of the Jacobian on particular perturbations,

which we can approximate by perturbing the initial measure, applying the micro-scale timestepper, and computing a velocity field via optimal transport. In Chapter 4, we detail this approach, and we show how it can be used to construct new, smoother timesteppers using inverse CDFs, which we then use to quickly find steady-state distributions. We show that our method drastically reduces the number of micro-scale simulations steps required to find steady-state distributions of many example particle system models, such as bacterial chemotaxis, economic agents, and Langevin dynamics in both one and two dimensions. This shows that our timestepper is not just good for more efficiently and accurately simulating trajectories of curves of measures — it can be used to do many system-level tasks, and efficiently computing steady states is just one such application.

Chapter 4, in full, is a reprint of material in [VKCK25]: Hannes Vandecasteele, Nicholas Karris, Alexander Cloninger, and Ioannis G. Kevrekidis. Optimal transport, timesteppers, Newton-Krylov methods and steady states of collective particle dynamics. Preprint, arXiv:2512.24567, 2025. The dissertation author is a primary author and investigator of this paper, which is currently under revision for publication in the Journal of Computational Physics.

1.2.4 Image Interpolation Using Wasserstein Geodesics

Finally, in Chapter 5, we demonstrate another application of this theory to diffusion models. One can understand the "forward process" in diffusion models as learning a nice way to push forward the unknown distribution to a known distribution (usually a Gaussian), and similarly the "reverse process" attempts to generate new samples from the known distribution and push them forward [HJA20, RBL+22]. These processes inherently involve distributions "evolving in time" (where time here is the process of (de-)noising), which means one can understand them in terms of the geometry of the Wasserstein manifold. This understanding also extends to the underlying embedding spaces, which can also be understood as spaces of distributions (i.e., point clouds). In Chapter 5, we

explore the geometry of embedding spaces by constructing different interpolating paths between embeddings and investigating how they affect the resulting image interpolations. We experimentally demonstrate that the interpolations given by optimal transport give more "natural looking" image interpolations than other interpolation methods, which suggests that the correct way to understand the geometry of embedding space is as a space of measures. Our work hints at a deeper "distributional" perspective on diffusion models that may be helpful for understanding their behavior, training process, and vulnerabilities.

Chapter 5, in full, is a reprint of the material in [KDFE26]: Nicholas Karris, Luke Durell, Javier Flores, and Tegan Emerson. Which way from B to A: The role of embedding geometry in image interpolation for Stable Diffusion. In *Proceedings of the 1st Conference on Topology, Algebra, and Geometry in Data Science (TAG-DS 2025)*, volume 321 of *Proceedings of Machine Learning Research*, pages 114–125. PMLR, 2026. The dissertation author is the primary author and investigator of this work, which was conducted under the Laboratory Directed Research and Development Program at PNNL, a multi-program national laboratory operated by Battelle for the U.S. Department of Energy under contract DE-AC05-76RL01830.

Acknowledgements

Chapter 1, in part (specifically Section 1.1), is a reprint of material in [KNK+25]: Nicholas Karris, Evangelos A. Nikitopoulos, Ioannis G. Kevrekidis, Seungjoon Lee, and Alexander Cloninger. Using linearized optimal transport to predict the evolution of stochastic particle systems. Preprint, arXiv:2408.01857, 2025. The dissertation author is the primary author and investigator of this paper, which is under revision for publication in the SIAM Journal on Applied Dynamical Systems.

Chapter 2

Euler's Method for Predicting Measure Evolution

We develop an Euler-type method to predict the evolution of a time-dependent probability measure without explicitly learning an operator that governs its evolution. We use linearized optimal transport theory to prove that the measure-valued analog of Euler's method is first-order accurate when the measure evolves "smoothly." In applications of interest, however, the measure is an empirical distribution of a system of stochastic particles whose behavior is only accessible through an agent-based micro-scale simulation. In such cases, this empirical measure does not evolve smoothly because the individual particles move chaotically on short time scales. However, we can still perform our Euler-type method, and when the particles' collective distribution approximates a measure that *does* evolve smoothly, we observe that the algorithm still accurately predicts this collective behavior over relatively large Euler steps, thus reducing the number of micro-scale steps required to step forward in time. In this way, our algorithm provides a "macro-scale timestepper" that requires less micro-scale data to still maintain accuracy, which we demonstrate with three illustrative examples: a biological agent-based model, a model of a PDE, and a model of Langevin dynamics.

2.1 Introduction

In this chapter, we consider the problem of predicting the evolution of a time-dependent probability measure $t \mapsto \mu_t$ on $\mathbb{R}^d$, representing the distribution of particles in some physical system, given very limited information about the system's underlying dynamics. Often, the physics governing the system can be approximated by an "oracle operator" E_h that takes as input the present state μ_t of the system and returns an approximation $E_h(\mu_t) \approx \mu_{t+h}$ after "evolving" according to the underlying dynamics for a short time h. As an example, E_h could be represented by an agent-based micro-scale model, in which individual particles act autonomously but exhibit some collective behavior when viewed as a group.

A natural way to understand such physical systems is to attempt to learn the oracle operator E_h. Many approaches to learning operators focus on tracking a sample of μ_t through the actions of E_h and then interpolating the results [LJP+21, GBYK23, GKSK22, Mha23, KLM24]. However, in many applications, it is not possible to track the action of E_h on individual sample particles (e.g., due to randomness or statistical unreliability). Moreover, these approaches typically attempt to learn the *global* dynamics (both in time and space) of the operator, which in general requires data about the operator's actions on particles at many locations and many times. Other approaches attempt to learn the dynamics by attempting to learn an SDE that approximates the observed evolution [TLGD24, BR25]. We consider a different perspective on tackling this problem. Instead of learning the entire operator E_h, we focus on the minimal data needed to predict the evolution of a particular initial distribution. We ask whether it is possible to predict the measure's evolution *without* needing to understand the underlying dynamics. Explicitly, given some μ_t and $E_h(\mu_t) \approx \mu_{t+h}$ for only some small time step h, can we predict the measure μ_{t+H} for a larger time step H? We should not need data about how E_h acts on other distributions at other times to do this. If we can, then we can build an Euler-type method for approximating the long-term evolution of the distribution by making successive

approximations, and this approximation method would only require *local* knowledge of E_h. There are several potential approaches that attempt to only use local data, each with its own merits and drawbacks. One drawback common to most approaches, though, is a reliance on computing and predicting summary statistics and then reconstructing a distribution with the predicted statistics. We avoid this reliance on summary statistics and instead take a linearized optimal transport (LOT)–based approach informed by the geometry of the Wasserstein manifold that directly uses the measure-valued data to predict μ_{t+H}.

To explain the philosophy of our prediction algorithm, we briefly discuss curves on the Wasserstein manifold. Precise definitions and results are reviewed in Section 1.1. Let $\mathcal{P}_2(\mathbb{R}^d)$ be the space of Borel probability measures on $\mathbb{R}^d$ with finite second moment, and let W_2 be the (2-)Wasserstein distance on $\mathcal{P}_2(\mathbb{R}^d)$. If μ is a sufficiently "smooth" curve, written $t \mapsto \mu_t$, on the "Wasserstein manifold" $\mathbb{W}_2 := (\mathcal{P}_2(\mathbb{R}^d), W_2)$, then there exists a "tangent field" governing its evolution. Explicitly, there is a time-dependent vector field $\mathbf{v}_t : \mathbb{R}^d \to \mathbb{R}^d$ representing the time derivative of μ through the continuity equation

$$\partial_t \mu_t + \nabla \cdot (\mathbf{v}_t \mu_t) = 0.$$

Intuitively speaking, the vector field $\mathbf{v}_t$ describes the "flow of mass" of the evolving measure. Furthermore, if μ always has a density, then its tangent field can be computed using optimal transport maps:

$$\mathbf{v}_t = \lim_{h \to 0} \frac{T_{\mu_t}^{\mu_{t+h}} - \mathrm{id}}{h},$$

where $T_{\mu_t}^{\mu_{t+h}}$ is the optimal transport map from μ_t to μ_{t+h} and $\mathrm{id} = \mathrm{id}_{\mathbb{R}^d}$. This leads to the key insight that one can use optimal transport maps from μ_t to μ_{t+h} for small values of h to understand approximately where each infinitesimal piece of mass of μ_t is moving. Moreover, the interpretation of $t \mapsto \mathbf{v}_t$ as the tangent field to the curve μ on the Wasserstein manifold suggests that the "geodesic starting at μ_t in the direction of $\mathbf{v}_t$," given

by $s \mapsto (s\mathbf{v}_t + \mathrm{id})_{\#}\mu_t$, is a first-order accurate approximation of the true curve. This result is made precise in Theorem 1.1.2.2(iii). It also suggests a measure-valued analog of Euler's method for ODEs on the Wasserstein manifold, which we develop in Section 2.2.2.

Euler's method requires complete knowledge of the dynamics of an ODE, and similarly, standard numerical methods for approximating measure evolution according to the continuity equation often require such knowledge [EGH00, BS08, NFM18]. As discussed in the first paragraph, our setting is fundamentally different because we suppose instead that we can only access dynamics information by querying an oracle E_h for a short time h. In this case, although we cannot know the true tangent field $t \mapsto \mathbf{v}_t$, we can hope to approximate it by

$$\mathbf{v}_t^h \coloneqq \frac{T_{\mu_t}^{E_h(\mu_t)} - \mathrm{id}}{h} \approx \frac{T_{\mu_t}^{\mu_{t+h}} - \mathrm{id}}{h}.$$

We then use this approximate tangent field to approximate μ_{t+H} as $\widetilde{\mu}_{t+H} \coloneqq (H\mathbf{v}_t^h + \mathrm{id})_{\#}\mu_t$, the geodesic starting at μ_t in the direction $\mathbf{v}_t^h$ evaluated at time H. Iterating such approximations leads to an Euler's method "with incomplete information," which we discuss in Sections 2.2.1 and 2.2.3 for sufficiently "smooth" curves with densities at all times. Note well that this Euler-type method allows us to predict the evolution of μ without attempting to learn anything about the operator E_h.

A key feature of the above-described approach is that Euler's method with incomplete information can be formulated in fairly arbitrary situations, e.g., for chaotically evolving discrete measures. In this chapter, we specifically consider cases arising from particle systems. Let $x_1(t), \ldots, x_N(t)$ be a system of N (random) particles in $\mathbb{R}^d$ evolving in time, and define $\mu_t^N \coloneqq N^{-1} \sum_{i=1}^N \delta_{x_i(t)}$. If $\nu \coloneqq \mu_t^N$ and $\rho \coloneqq E_h(\mu_t^N) \approx \mu_{t+h}^N$, then the Euler-type method described above becomes the following algorithm:

1. Compute a (discrete) optimal transport plan from ν to ρ, and compute the barycentric projection to obtain a map T.

2. Use the map from Step 1 to form the difference quotient $\mathbf{v}_t^h \coloneqq \frac{T - \mathrm{id}}{h}$.

3. For each point in the support of ν, take an Euler step using its current location and the velocity given by the "optimal transport vector field" $\mathbf{v}_t^h$.

4. Set $\widetilde{\mu}_{t+H}^N$ to be the uniform distribution on the points obtained by performing the Euler step from Step 3 on each point in the support of ν.

5. Repeat Steps 1–4 using $\widetilde{\mu}_{t+H}^N$ as the "initial measure" ν.

Our actual algorithm, laid out in Section 2.3.3, is more complicated. Steps 1–5 above suffice for the present introductory discussion.

It is important to highlight that *both* regularity properties demanded of μ in the earlier discussion of tangent fields and Euler-type methods are generally false for μ^N, the curve $t \mapsto \mu_t^N$. Certainly, μ^N never has a density because it is always a discrete measure. Moreover, μ^N can fail to be "smooth" because the particles may evolve chaotically. For example, if $x_1, \ldots, x_N$ are Brownian motions, then the sample paths are nowhere differentiable. Therefore, *a priori*, it is not clear the algorithm above actually works in this setting. In many systems of interest, however, there is a "smooth" curve $t \mapsto \mu_t$ of measures with densities such that "$\mu^N \approx \mu$" for large N, e.g., if we start with the curve μ and the x_i's are i.i.d. samples of μ. For certain such systems (Sections 2.4, 2.5, and 2.6), we experimentally demonstrate that the LOT-based algorithm described above can be used to predict the evolution of the particle distribution μ^N without explicitly reconstructing the underlying distribution μ or learning the underlying dynamics described by E_h.

2.1.1 Practical Utility for Agent-based Models

One application for which our approximation algorithm is particularly useful is if E_h is given by a computationally expensive micro-scale particle simulation, as it is in Section 2.4. Then we can think of μ_t^N as the locations of the particles at time t and $E_h(\mu_t^N) \approx \mu_{t+h}^N$ as their locations at the "next time step" in the simulation. The benefit of accurately predicting μ_{t+H}^N for a substantial time step $H \gg h$ is that one can then skip potentially

many micro-scale steps in the simulation and still understand how μ^N evolves in time. Indeed, after predicting μ^N_{t+H}, one can reinitialize the simulation with particles at those locations, take another small step in the simulation to obtain an estimate for μ^N_{t+H+h}, and use the prediction algorithm again to approximate μ^N_{t+2H}. Following strategies of some projective integration techniques [GK03a, GK03b, SGOK05], doing this successively gives an Euler-type method for approximating the system's behavior for a long time without the need to perform as many micro-scale steps.

Approximating distributional trajectories through an Euler-type method is not the only application of our proposed approach, however. At its core, our approach gives a way to step forward in time without needing to compute the equivalent number of steps of the micro-scale model E_h. If computing E_h is expensive, then this means we get a comparatively cheap "macro-scale timestepper," which has many applications beyond merely speeding up simulations. For example, one could use such a macro-scale timestepper to perform system-level tasks such as fixed-point computations (including finding steady-state distributions which are dynamically unstable), bifurcation computations, stability analysis, and even coarse-grained feedback controller design and optimization [SPK03, FVG+25, GKT02]. Thus, while we primarily evaluate our method on its accuracy in estimating distributional trajectories, the introduction of an accurate, cheap macro-scale timestepper has many implications beyond those directly discussed in the present chapter. As such, when evaluating the "computational improvement" of our method, we focus entirely on the number of micro-scale steps needed in its computation rather than the actual speed improvement of the simulation. Even though the micro-scale models we use in our experiments are not *extremely* expensive (and in Sections 2.5 and 2.6 are actually fairly cheap), one should have in mind applications where these micro-scale models are indeed the computational bottleneck as they are in, e.g., other agent-based biological models, climate models, etc.

One particular motivating class of such models are those of so-called "fast-slow"

systems. Such systems are difficult to analyze because each particle has multiple underlying variables impacting its behavior that update on vastly different time scales. In general, the "fast dynamics" update on short time scales and generally manifest as chaotic short-time particle behavior. For example, a particle may quickly change directions many times as a result of its interactions with nearby particles. "Slow dynamics," on the other hand, update on much longer time scales and generally manifest as the eventual emergence of a collective behavior of the particles, e.g., the migration of bacteria toward a higher concentration of a food source. Nominally, this means that in order to simulate such a system, a micro-scale simulation must be run with small time steps to capture the fast dynamics accurately, and *many* such time steps are required for the slow dynamics to emerge. Of course, this can be impractical when each step is computationally expensive. One may hope to circumvent this by increasing the step size for the micro-scale simulation, thus requiring fewer steps to reach the desired end time, but some models computationally break down if the time step is too large (for example, the micro-scale model used in 2.4 has this property — see Remark 2.4.2.1) and others quickly lose accuracy [GK03a, RMGK03]. Thus, a different approach is required to accurately simulate such systems.

Many approaches to understanding and more efficiently simulating these types of systems have been explored. A few examples of so-called "multi-scale methods" are presented in [KGH$^+$03, KGH04, GKT02, VE03, ELVE05, EE03], and other "projective integration methods" are proposed in [GK03a, GGK09] and others. More generally, there are many techniques that can be used to understand similar problems with evolving distributions. The techniques of Neural ODEs and Continuous Normalizing Flows can be used, e.g., in [HMT$^+$22, THW$^+$20], and while these techniques avoid the computational cost of simulating the system, they each do so by restricting attention to specific classes of particle systems. The techniques of Schrodinger-bridge methods and Wasserstein gradient flows also handle similar settings [Léo14, CGP16, AGS08, CNWR25], but they are built for different objectives — Schrodinger bridge methods interpolate between measures, and

Wasserstein gradient flows seek to minimize a known functional. These methods are not appropriate for our goal of extrapolating an unknown trajectory that is not necessarily energy-minimizing.

Another related approach that does work on general classes of particle systems is learning an underlying SDE that models the observed behavior of the micro-scale simulator [TLGD24, BR25]. One could then hope to achieve a similar effect of skipping many expensive micro-scale steps by simulating the learned SDE (which is much cheaper) to the desired end time. However, learning an SDE is prone to errors when only given local data about the behavior of the particles. Moreover, these approaches assume that there is an underlying SDE that can accurately approximate the behavior of the micro-scale model at all. In contrast, the method we propose makes efficient use of the limited local data, and it also assumes essentially nothing about the particular class of the underlying micro-scale simulator, only that the bulk behavior appears "smooth." In particular, it does not require an assumption about the precise form of the dynamics of an agent-based model, nor does it attempt to learn those dynamics explicitly. We avoid the computational cost not by making an assumption about the problem but by using optimal transport to detect the slow dynamics after only a small number of micro-scale steps. The key insight is that, for systems that exhibit this multi-scale behavior, we can use an optimal transport map to effectively ignore the chaotic motion of *individual* particles in favor of focusing on the slower collective behavior.

This insight has been used to a similar effect before: [SGOK05] explores how a micro-scale model of bacterial chemotaxis in 1-D could be sped up by computing Euler-type steps after first sorting the locations of the bacteria. In one dimension, sorting a discrete distribution is exactly the same as computing an optimal transport map, so it turns out that their approach is similar to ours. This observation was the genesis of the present chapter. To demonstrate the practical utility of our algorithm, we perform a similar experiment with the same chemotaxis model (Section 2.4) as well as a model of a partial differential equation

(Section 2.5). We show that our algorithm enables us to skip many micro-scale steps and still approximate the long-term behavior of the bacteria distribution, indicating that our Euler-type "macro-scale timestepper" is indeed reasonably accurate despite requiring fewer micro-scale steps.

Framing the algorithm in terms of optimal transport as opposed to particle sorting not only provides it theoretical motivation, as discussed earlier, but also enables its immediate generalization to particle systems in higher dimensions. To demonstrate its efficacy on 2-D particle systems, we use the algorithm to simulate a model of overdamped Langevin dynamics in 2D (Section 2.6). While standard micro-scale models of Langevin dynamics are computationally cheap, the example is a good proof of concept for higher dimensions because it exhibits the same multi-scale behavior as described above. Specifically, the particles move chaotically on short time scales, yet the overall distribution evolves slowly and predictably.

2.1.2 Major Contributions

- We use linearized optimal transport to describe an analog of Euler's method for differential equations on the Wasserstein manifold (Section 2.2.2).
- We prove that, under sufficient regularity conditions, our measure-valued Euler's method with step size H has local truncation error $O(H^2)$ and corresponding global error $O(H)$ (Lemma 2.2.2.9 and Theorem 2.2.2.3).
- We detail an algorithm that uses linearized optimal transport to approximate the evolution of discrete particle systems (Section 2.3).
- We apply our approximation algorithm to three illustrative examples — bacterial chemotaxis as described in [SGOK05] (Section 2.4), a partial differential equation (Section 2.5), and Langevin dynamics (Section 2.6) — and discuss its computational benefits.

- We demonstrate that our method outperforms equivalent methods (a particle-wise approach and JKOnet* [TLGD24]) for approximating the distribution's evolution when only given access to local data.

2.2 Euler-type Methods on the Wasserstein Manifold

2.2.1 Motivation from Euclidean Space

In the sections following this one, we develop Euler-type methods for approximating the solutions to (nice) "ODEs" in the Wasserstein manifold. To understand these methods, it is helpful to discuss their analogs in the context of ODEs in $\mathbb{R}^d$. These analogs in $\mathbb{R}^d$ themselves are not useful methods for numerically solving ODEs, but they are useful to present to build intuition for what our Euler-type method is doing geometrically. To this end, suppose we have a particle whose location at time t is described by some curve $\gamma(t)$ solving the initial value problem

$$\gamma(0) = x_0, \qquad \gamma'(t) = f(t, \gamma(t)).$$

Our goal is to estimate $\gamma(T)$ for some large T. If we happen to have access to the underlying "dynamics" function f, then we can use ordinary Euler's method: For a time step $H > 0$ and discrete times $t_n = nH$, approximate $\gamma(t_n) \approx \gamma_{(n)}$, where $\gamma_{(n)}$ is defined recursively by

$$\gamma_{(0)} = x_0, \qquad \gamma_{(n+1)} = \gamma_{(n)} + Hf(t_n, \gamma_{(n)}) = \exp^{\mathbb{R}^d}_{\gamma_{(n)}}(Hf(t_n, \gamma_{(n)})),$$

where $\exp^{\mathbb{R}^d}_{\gamma_{(n)}}$ denotes the exponential map of $\mathbb{R}^d$ at $\gamma_{(n)}$. Under sufficient regularity conditions on f, it is well known that Euler's method is first-order accurate. The measure-valued analog of this method is developed in Section 2.2.2, and we prove an analogous accuracy result.

Suppose now that we have "imperfect information" about our system's dynamics, i.e., that we do not have explicit access to the underlying function f. Then we cannot perform Euler's method exactly. Instead, suppose we have access to some oracle E_h that, given a time t and a particle location γ, returns an approximation of where that particle would be at time $t + h$ for some small time step $h > 0$. (For us, this oracle will be a micro-scale numerical simulation that is only accurate for very small time steps h and is computationally expensive, so using it to simulate long-term behavior is impractical.) Given this information, we attempt to approximate $f(t, \gamma)$ by

$$f(t, \gamma) \approx \frac{E_h(t, \gamma) - \gamma}{h} =: \hat{f}(t, \gamma).$$

Using this approximation, we then define the Euler-type method

$$\gamma_{(0)} = x_0, \qquad \gamma_{(n+1)} = \gamma_{(n)} + H\hat{f}(t_n, \gamma_{(n)}) = \exp^{\mathbb{R}^d}_{\gamma_{(n)}}\left(H\hat{f}(t_n, \gamma_{(n)})\right).$$

Of course, this method also makes sense for other approximations $\hat{f}$ of f, and the error of the method depends on how well $\hat{f}$ approximates f. While precise statements can be made, we omit the details for brevity. The measure-valued analog of this Euler-type method with a finite-difference approximation of the "dynamics" function is discussed in Section 2.2.3. In Section 2.3, we describe the "discrete case" of this approximation algorithm (in which E_h is a given micro-scale simulator).

2.2.2 Euler's Method on the Wasserstein Manifold

According to Theorem 1.1.2.2's characterization of tangent fields to curves on $\mathbb{W}_2$, the measure-valued analog of the ODE $\gamma'(t) = f(t, \gamma(t))$ is the continuity equation

$$\partial_t \mu_t + \nabla \cdot (F(t, \mu_t)\, \mu_t) = 0, \tag{2.2.2.1}$$

where F is some "(time-dependent) vector field on the Wasserstein manifold." Given such a "vector field" F and an initial measure μ_0, the measure-valued analog of Euler's method is the following: For discrete times $t_n = nH$, set

$$\mu_{(0)} := \mu_0, \qquad \mu_{(n+1)} := \exp^{\mathbb{W}_2}_{\mu_{(n)}}(HF(t_n, \mu_{(n)})) = (\mathrm{id} + HF(t_n, \mu_{(n)}))_{\#}\mu_{(n)}. \tag{2.2.2.2}$$

With this definition, one should expect that $\mu_{(n)} \approx \mu_{t_n}$ is first-order accurate under appropriate assumptions on F. The following result makes this precise.

Theorem 2.2.2.3. *Let $F : [0,T] \times \mathcal{P}_2(\mathbb{R}^d) \to (\mathbb{R}^d)^{\mathbb{R}^d} = \{\text{functions } \mathbb{R}^d \to \mathbb{R}^d\}$ be a map such that $F(t,\mu) \in T_\mu \mathbb{W}_2$ for all $t \in [0,T]$ and $\mu \in \mathcal{P}_2(\mathbb{R}^d)$. Suppose F satisfies the following Lipschitz conditions: There exist $L_1, L_2 \geq 0$ such that for all $t \in [0,T]$, $x_1, x_2 \in \mathbb{R}^d$, and $\mu_1, \mu_2 \in \mathcal{P}_2(\mathbb{R}^d)$,*

$$\|F(t,\mu)(x_1) - F(t,\mu)(x_2)\|_2 \leq L_1 \|x_1 - x_2\|_2 \quad \textit{and}$$

$$\|F(t,\mu_1) - F(t,\mu_2)\|_{L^2(\mu_i)} \leq L_2 W_2(\mu_1,\mu_2) \quad (i = 1,2).$$

If $\mu : [0,T] \to \mathcal{P}_2(\mathbb{R}^d)$ is an absolutely continuous curve solving the continuity equation (2.2.2.1) *on $(0,T)$ in the sense of distributions and the function $(t,x) \mapsto \mathbf{v}_t(x) := F(t,\mu_t)(x)$ is (jointly) C^1, then for μ_0-a.e. $x \in \mathbb{R}^d$, the characteristic ODE*

$$\gamma_x(0) = x, \qquad \frac{d}{dt}\gamma_x(t) = \mathbf{v}_t(\gamma_x(t))$$

admits a unique C^2 solution. Furthermore, if

$$M := \left(\int_{\mathbb{R}^d} \max_{r \in [0,T]} \|\gamma_x''(r)\|_2^2 \, d\mu_0(x) \right)^{\frac{1}{2}} < \infty,$$

then for a time step $H > 0$ and discrete times $t_n := nH$, the sequence of Euler approximations

iteratively defined by

$$\mu_{(0)} := \mu_0, \qquad \mu_{(n+1)} := (\mathrm{id} + HF(t_n, \mu_{(n)}))_{\#}\mu_{(n)}$$

satisfies

$$W_2(\mu_{(n)}, \mu_{t_n}) \leq \frac{HM}{2(L_1 + L_2)} \left(e^{nH(L_1+L_2)} - 1 \right)$$

for all n such that $t_n = nH \leq T$.

The rest of this section is devoted to proving this result. To begin, observe that (1.1.2.6) says the local truncation error of Euler's method on the Wasserstein manifold is $o(H)$. We now show that under our regularity assumptions on F, this method has local truncation error $O(H^2)$. This is the same as the local truncation error of Euler's method for curves in Euclidean space with bounded second (e.g., see (6.2.5) of [Atk89]).

Lemma 2.2.2.4 (Lemma 8.1.4 and Proposition 8.1.8 in [AGS08])**.** *Suppose $\mu : [0, T] \to \mathcal{P}_2(\mathbb{R}^d)$ is an absolutely continuous curve satisfying* (1.1.2.4) *on $(0, T)$ with velocity field $\mathbf{v}$ satisfying*

$$\int_0^T \int_{\mathbb{R}^d} \|\mathbf{v}_t(x)\|_2 \, d\mu_t(x) dt < \infty \tag{2.2.2.5}$$

and

$$\int_0^T \left(\sup_{x \in B} \|\mathbf{v}_t(x)\|_2 + \mathrm{Lip}(\mathbf{v}_t, B) \right) dt < \infty \tag{2.2.2.6}$$

for all compact set $B \subset \mathbb{R}^d$. Then for μ_0-a.e. $x \in \mathbb{R}^d$, the ODE

$$\gamma_x(0) = x, \qquad \frac{d}{dt}\gamma_x(t) = \mathbf{v}_t(\gamma_x(t)) \tag{2.2.2.7}$$

admits a unique solution defined on $[0, T]$, and

$$\mu_t = (T_t)_{\#}\mu_0 \qquad \text{for all } t \in [0, T], \tag{2.2.2.8}$$

where $T_t(x) := \gamma_x(t)$.

Lemma 2.2.2.9. *Assume the hypotheses of Lemma 2.2.2.4. Fix* $t \in [0, T]$*, let* $H > 0$ *be such that* $t + H \leq T$*, and let* $\gamma_x : [t, t + H] \to \mathbb{R}^d$ *be the solution to*

$$\gamma_x(t) = x, \qquad \gamma_x'(s) = \frac{d}{ds}\gamma_x(s) = \mathbf{v}_s(\gamma_x(s))$$

If γ_x *is* C^2 *for* μ_t*-a.e.* $x \in \mathbb{R}^d$ *and*

$$\int_{\mathbb{R}^d} \max_{r \in [t,t+H]} \|\gamma_x''(r)\|_2^2 \, d\mu_t(x) < \infty,$$

then

$$W_2(\mu_{t+H}, (\mathrm{id} + H\mathbf{v}_t)_{\#}\mu_t) \leq \frac{H^2}{2}\left(\int_{\mathbb{R}^d} \max_{r \in [t,t+H]} \|\gamma_x''(r)\|_2^2 \, d\mu_t(x)\right)^{\frac{1}{2}}.$$

Proof. Define $T_s : \mathbb{R}^d \to \mathbb{R}^d$ by $T_s(x) = \gamma_x(s)$. By Lemma 2.2.2.4, $\mu_{t+H} = (T_{t+H})_{\#}\mu_t$, so

$$\begin{aligned} W_2^2(\mu_{t+H}, (\mathrm{id} + H\mathbf{v}_t)_{\#}\mu_t) &\leq \int_{\mathbb{R}^d} \|T_{t+H}(x) - (\mathrm{id} + H\mathbf{v}_t)(x)\|_2^2 \, d\mu_t(x) \\ &= \int_{\mathbb{R}^d} \|\gamma_x(t + H) - [x + H\mathbf{v}_t(x)]\|_2^2 \, d\mu_t(x). \end{aligned}$$

If γ_x is C^2 for a particular $x \in \mathbb{R}^d$, then

$$\begin{aligned} \gamma_x(t + H) &= \gamma_x(t) + H\gamma_x'(t) + \int_t^{t+H}\int_t^s \gamma''(r)\, dr ds \\ &= x + H\mathbf{v}_t(x) + \int_t^{t+H}\int_t^s \gamma''(r)\, dr ds. \end{aligned}$$

Hence, if γ_x is C^2 for μ_t-a.e. $x \in \mathbb{R}^d$, then

$$\begin{aligned} W_2^2(\mu_{t+H}, (\mathrm{id} + H\mathbf{v}_t)_{\#}\mu_t) &\leq \int_{\mathbb{R}^d} \left\| \int_t^{t+H}\int_t^s \gamma_x''(r)\, dr ds \right\|_2^2 d\mu_t(x) \\ &\leq \int_{\mathbb{R}^d} \frac{H^4}{4} \max_{r \in [t,t+H]} \|\gamma_x''(r)\|_2^2 \, d\mu_t(x). \end{aligned}$$

□

As in the Euclidean case, the fact that the local truncation error of Euler's method on the Wasserstein manifold is $O(H^2)$ implies that, under our Lipschitz conditions on F, the global error is first order. This is exactly the content of Theorem 2.2.2.3.

Proof of Theorem 2.2.2.3. We first prove that the ODE (2.2.2.7) admits a unique solution on $[0, T]$ for μ_0-a.e. $x \in \mathbb{R}^d$. By Lemma 2.2.2.4, it suffices to show that $\mathbf{v}_t := F(t, \mu_t)$ satisfies (2.2.2.5) and (2.2.2.6). For the first, note that

$$\begin{aligned}\int_0^T \int_{\mathbb{R}^d} \|\mathbf{v}_t(x)\|_2 \, d\mu_t(x)dt &\leq \int_0^T \int_{\mathbb{R}^d} [\|\mathbf{v}_t(0)\|_2 + \|\mathbf{v}_t(x) - \mathbf{v}_t(0)\|_2] \, d\mu_t(x)dt \\ &\leq \int_0^T \int_{\mathbb{R}^d} [\|\mathbf{v}_t(0)\|_2 + L_1\|x\|_2] \, d\mu_t(x)dt \\ &= \int_0^T \|\mathbf{v}_t(0)\|_2 \, dt + L_1 \int_0^T \int_{\mathbb{R}^d} \|x\|_2 \, d\mu_t(x)dt.\end{aligned}$$

We assume $(t, x) \mapsto \mathbf{v}_t(x)$ is C^1, so

$$\int_0^T \|\mathbf{v}_t(0)\|_2 \, dt < \infty,$$

and $t \mapsto \int_{\mathbb{R}^d} \|x\|_2 \, d\mu_t(x)$ is continuous (because $t \mapsto \mu_t$ is continuous with respect to W_2 by hypothesis and $\mu \mapsto \int_{\mathbb{R}^d} \|x\|_2 \, d\mu(x)$ is continuous with respect to W_2 because $\|x\|$ grows less than quadratically (see Remark 7.1.11 in [AGS08])), so

$$\int_0^T \int_{\mathbb{R}^d} \|x\|_2 \, d\mu_t(x)dt < \infty.$$

Hence, we have (2.2.2.5). For (2.2.2.6), if $B \subset \mathbb{R}^d$ is compact, then

$$\int_0^T \left(\sup_{x \in B} \|\mathbf{v}_t(x)\|_2 + \mathrm{Lip}(\mathbf{v}_t, B) \right) dt \leq T \left(\sup_{t \in [0,T], x \in B} \|\mathbf{v}_t(x)\|_2 + L_1 \right) < \infty$$

again because $(t, x) \mapsto \mathbf{v}_t(x)$ is C^1. By Lemma 2.2.2.4, the ODE (2.2.2.7) has a unique globally defined solution γ_x for μ_0-a.e. $x \in \mathbb{R}^d$, and since $(x, t) \mapsto \mathbf{v}_t(x)$ is C^1, the map

$t \mapsto \gamma_x(t)$ is C^2 for such x (since its second derivative is $\gamma_x''(t) = \frac{d}{dt}\mathbf{v}_t(\gamma_x(t))$, which is continuous by C^1 of $\mathbf{v}_t(x)$).

Now, for

$$M := \left(\int_{\mathbb{R}^d} \max_{r \in [0,T]} \|\gamma_x''(r)\|_2^2 \, d\mu_0(x) \right)^{\frac{1}{2}} < \infty,$$

we will show

$$W_2(\mu_{(n+1)}, \mu_{t_{n+1}}) \leq (1 + H(L_1 + L_2)) W_2(\mu_{(n)}, \mu_{t_n}) + \frac{H^2}{2} M. \tag{2.2.2.10}$$

By the triangle inequality for W_2, we have

$$\begin{aligned}
W_2(\mu_{(n+1)}, \mu_{t_{n+1}}) &= W_2((\mathrm{id} + HF(t_n, \mu_{(n)}))_{\#}\mu_{(n)}, \mu_{t_{n+1}}) \\
&\leq W_2((\mathrm{id} + HF(t_n, \mu_{(n)}))_{\#}\mu_{(n)}, (\mathrm{id} + HF(t_n, \mu_{t_n}))_{\#}\mu_{(n)}) \qquad (*) \\
&\quad + W_2((\mathrm{id} + HF(t_n, \mu_{t_n}))_{\#}\mu_{(n)}, (\mathrm{id} + HF(t_n, \mu_{t_n}))_{\#}\mu_{t_n}) \qquad (\dagger) \\
&\quad + W_2((\mathrm{id} + HF(t_n, \mu_{t_n}))_{\#}\mu_{t_n}, \mu_{t_{n+1}}). \qquad (\ddagger)
\end{aligned}$$

We will handle each term separately. For $(*)$, note that for any $\mu \in \mathcal{P}_2(\mathbb{R}^d)$ and any maps $T, S : \mathbb{R}^d \to \mathbb{R}^d$, the measure $\gamma = (T, S)_{\#}\mu$ is a (generally suboptimal) coupling of $T_{\#}\mu$ and $S_{\#}\mu$, and so

$$W_2(T_{\#}\mu, S_{\#}\mu) \leq \left(\int_{\mathbb{R}^d} \|T(x) - S(x)\|^2 \, d\mu(x) \right)^{\frac{1}{2}}.$$

Thus, we have

$$\begin{aligned}
(*) &\leq \left(\int_{\mathbb{R}^d} \left\| (\mathrm{id} + HF(t_n, \mu_{(n)}))(x) - (\mathrm{id} + HF(t_n, \mu_{t_n}))(x) \right\|^2 \, d\mu_{(n)}(x) \right)^{\frac{1}{2}} \\
&= \|(\mathrm{id} + HF(t_n, \mu_{(n)})) - (\mathrm{id} + HF(t_n, \mu_{t_n}))\|_{L^2(\mu_{(n)})} \\
&= H \|F(t_n, \mu_{(n)}) - F(t_n, \mu_{t_n})\|_{L^2(\mu_{(n)})} \\
&\leq H L_2 W_2(\mu_{(n)}, \mu_{t_n}).
\end{aligned}$$

For (†), note that for any Lipschitz map $T : \mathbb{R}^d \to \mathbb{R}^d$ and any measures $\sigma, \mu \in \mathcal{P}_2(\mathbb{R}^d)$ with optimal coupling γ, we have that $(T \times T)_\# \gamma$ is a coupling of $T_\# \sigma$ and $T_\# \mu$, and so

$$\begin{aligned} W_2(T_\#\sigma, T_\#\mu) &\leq \left(\int_{\mathbb{R}^d \times \mathbb{R}^d} \|T(x) - T(y)\|^2 \, d\gamma(x,y) \right)^{\frac{1}{2}} \\ &\leq \|T\|_{\mathrm{Lip}} \left(\int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\| \, d\gamma(x,y) \right)^{\frac{1}{2}} \\ &= \|T\|_{\mathrm{Lip}} W_2(\sigma, \mu). \end{aligned}$$

Hence

$$\begin{aligned} (\dagger) &\leq \| \operatorname{id} + HF(t_n, \mu_{t_n}) \|_{\mathrm{Lip}} W_2(\mu_{(n)}, \mu_{t_n}) \\ &\leq (1 + HL_1) W_2(\mu_{(n)}, \mu_{t_n}). \end{aligned}$$

Finally, for (‡), we have

$$(\ddagger) \leq \frac{H^2}{2} M$$

by Lemma 2.2.2.9.

Now, we recall the following elementary fact: If $\beta \geq 0$, $\alpha > 0$, and $(a_n)_{n \in \mathbb{N}}$ is a sequence of nonnegative numbers satisfying the recursive property that $a_{n+1} \leq (1+\alpha)a_n + \beta$, then

$$a_n \leq e^{n\alpha} \left(a_0 + \frac{\beta}{\alpha} \right) - \frac{\beta}{\alpha}.$$

Applying this to (2.2.2.10) with $\alpha = H(L_1 + L_2)$ and $\beta = \frac{H^2}{2} M$ gives

$$W_2(\mu_{(n)}, \mu_{t_n}) \leq \frac{HM}{2(L_1 + L_2)} \left(e^{nH(L_1+L_2)} - 1 \right),$$

as desired. □

Before moving on, it is important to discuss the practical utility of Theorem 2.2.2.3 and its similarity to the standard error bound result for ordinary Euler's method in $\mathbb{R}^d$ (e.g., see (6.2.13) of [Atk89]). We note that the role of the Lipschitz constant L_2 is exactly analogous to the role of the Lipschitz constant in the proof of the $\mathbb{R}^d$ Euler result, and the role of L_1 is to ensure that the vector fields returned by F are sufficiently regular that the curves γ_x themselves can be well approximated using Euler's method. This is also the role of the assumption that $(t, x) \mapsto \mathbf{v}_t(x) = F(t, \mu_t)(x)$ is jointly C^1 — this guarantees that the curves γ_x have bounded second derivative, and so the definition of M is sensible. These hypotheses may seem rather restrictive, but there are indeed "natural" examples which satisfy them.

For example, consider an SDE of the form $dX_t = -\nabla\phi(X_t)dt + \sqrt{2D}dW_t$ with potential $\phi : \mathbb{R}^d \to \mathbb{R}$ and diffusion coefficient $D > 0$. It is well known that the density $p(x, t)$ of the law μ_t of X_t evolves according to a parabolic partial differential equation (the Fokker-Planck equation) which we can use [Pav14] to compute the tangent field

$$\mathbf{v}_t = -\nabla(\phi + D \log p_t) \in T_{\mu_t}\mathbb{W}_2.$$

Standard regularity results (e.g., [Fri64, Fri75, LSU68]) show that p is approximately as smooth as ϕ; in particular, if $\phi \in C^\infty$ and satisfies reasonable growth conditions, then $p \in C^\infty$, and hence so is $\mathbf{v}_t$.

As an explicit example, consider the standard SDE of Gaussian diffusion $dX_t = dW_t$ with initial condition $X_0 \sim \mu_0 = \delta_0$, for which the law of X_t is known to be $\mu_t = \mathcal{N}(0, tI_d)$. Since optimal transport maps between Gaussian distributions has a closed form, we can use (1.1.2.7) to compute the minimal-energy velocity flow field $\mathbf{v}_t$ in the continuity equation:

$$\mathbf{v}_t(x) = \lim_{h\to 0} \frac{T_{\mu_t}^{\mu_{t+h}}(x) - x}{h} = \lim_{h\to 0} \frac{\sqrt{\frac{t+h}{t}} - 1}{h} x = \frac{x}{2t}. \tag{2.2.2.11}$$

Note that for $t > 0$, $\mathbf{v}_t$ is jointly C^1 and has Lipschitz constants $L_1 = \frac{1}{t}$ and $L_2 = 0$.

These regularity results are certainly not necessary for $\mathbf{v}_t$ to be jointly C^1 (see [Fri64, Fri75, LSU68] for more detail), and there are surely broader settings in which the hypotheses can be satisfied. However, we wish to be clear that our goal in presenting these theoretical results is not so that we can directly apply them to "real world" examples. Instead, we wish to provide theoretical motivation for the discrete algorithm we present in Section 2.3, to elucidate the analogy to ordinary Euler's method, and to demonstrate that similar guarantees are, at least in theory, possible to obtain. In this way, the primary takeaway is not that our algorithm can be *guaranteed* to work on a particular system *a priori* (indeed, the point of our algorithm is that it does not require any knowledge of the underlying dynamics governing the system), but rather that it empirically performs well on example systems for which it is believable that the appropriate hypotheses hold, even if one cannot check them explicitly.

2.2.3 Euler's Method with Imperfect Information

The above discussion shows that, under sufficient regularity assumptions, Euler's method on the Wasserstein manifold can approximate solutions to the continuity equation with tangent field prescribed by a "vector field" F. However, as discussed in Section 2.2.1, we rarely have access to F in practice, so we cannot explicitly compute the velocity field to be used in Euler's method. Instead, we will only have access to some oracle E_h that, given a time t and a measure μ, will return an approximation of what that measure would be at time $t + h$ if allowed to evolve according to F for some small time step h. That is, if μ is a solution to the continuity equation with $\mathbf{v}_t = F(t, \mu_t)$, then the oracle is such that

$$E_h(t, \mu_t) \approx \mu_{t+h}.$$

To perform an Euler-type method, then, we must first approximate the appropriate velocity field, and (1.1.2.7) hints that we should approximate it with

$$\mathbf{v}_t^h(x) := \frac{T_{\mu_t}^{E_h(t,\mu_t)}(x) - x}{h} \approx \frac{T_{\mu_t}^{\mu_{t+h}}(x) - x}{h}.$$

Unfortunately, even if $E_h(\mu_t) = \mu_{t+h}$ exactly, the L^2 convergence in (1.1.2.7) is not enough to guarantee that $\mathbf{v}_t^h(x) \to \mathbf{v}_t(x)$ pointwise as $h \to 0$. However, the L^2 convergence is still good enough to give a reasonable bound on the local truncation error of using this finite difference approximation in Euler's method.

Corollary 2.2.3.1. *Suppose $\mu_t \ll \mathcal{L}^d$ and $h > 0$. Set*

$$\mathbf{v}_t^h(x) := \frac{T_{\mu_t}^{E_h(t,\mu_t)}(x) - x}{h}.$$

Then

$$W_2(\mu_{t+H}, (\mathrm{id} + H\mathbf{v}_t^h)_\# \mu_t) \leq H\|\mathbf{v}_t - \mathbf{v}_t^h\|_{L^2(\mu_t)} + o(H) \tag{2.2.3.2}$$

as $H \to 0$, and if $E_h(t, \mu_t) = \mu_{t+h}$, then $\|\mathbf{v}_t - \mathbf{v}_t^h\|_{L^2(\mu_t)} \to 0$ as $h \to 0$.

Proof. First note that, since we assume $\mu_t \ll \mathcal{L}^d$, we can apply (1.1.2.7) to give that $\|\mathbf{v}_t - \mathbf{v}_t^h\|_{L^2(\mu_t)} \to 0$ as $h \to 0$ when $E_h(t, \mu_t) = \mu_{t+h}$.

For the inequality, by (1.1.2.6) we have

$$\begin{aligned} W_2(\mu_{t+H}, (\mathrm{id} + H\mathbf{v}_t^h)_\# \mu_t) &\leq W_2(\mu_{t+H}, (\mathrm{id} + H\mathbf{v}_t)_\# \mu_t) + W_2((\mathrm{id} + H\mathbf{v}_t)_\# \mu_t, (\mathrm{id} + H\mathbf{v}_t^h)_\# \mu_t) \\ &\leq o(H) + \|(\mathrm{id} + H\mathbf{v}_t) - (\mathrm{id} + H\mathbf{v}_t^h)\|_{L^2(\mu_t)} \\ &= o(H) + H\|\mathbf{v}_t - \mathbf{v}_t^h\|_{L^2(\mu_t)} \end{aligned}$$

where in the second line we once again use the fact that for any maps $T, S : \mathbb{R}^d \to \mathbb{R}^d$, we have $W_2(T_\# \mu, S_\# \mu) \leq \|T - S\|_{L^2(\mu)}$ because $(T, S)_\# \mu$ gives a (generally suboptimal) coupling between $T_\# \mu$ and $S_\# \mu$. □

This is the local truncation error of the Euler-type method

$$\mu_{(0)} = \mu_0, \qquad \mu_{(n+1)} = (\mathrm{id} + H\mathbf{v}_t^h)_{\#}\mu_{(n)}. \tag{2.2.3.3}$$

As was alluded to in Section 2.2.1, the local truncation error of this method depends on the accuracy of the approximation of the "tangent vector" $\mathbf{v}_t = F(t, \mu_t)$.

2.3 Euler-type Method for Empirical Measures of Particle Systems

The results in Section 2.2 show that an Euler-type scheme can effectively predict the evolution of a measure-valued curve (under certain conditions) even with incomplete knowledge of the underlying "dynamics" governing the curve's evolution. In practical applications, one is often interested in predicting the evolution of (random) systems of particles $\{x_i(t)\}_{i=1}^N$ in $\mathbb{R}^d$. In such cases, the measure-valued curve one seeks to predict is the empirical measure

$$t \mapsto \mu_t^N := \frac{1}{N}\sum_{i=1}^N \delta_{x_i(t)},$$

and the oracle-type information one has is a particle-wise micro-scale simulation. Using this particle-wise simulation, we can construct an "oracle operator" E_h on discrete measures. If $\nu = \frac{1}{N}\sum_{i=1}^N \delta_{y_i}$ is a discrete measure, then $E_h(\nu) := \frac{1}{N}\sum_{i=1}^N \delta_{z_i}$, where $\{z_i\}_{i=1}^N$ is the output of running the micro-scale simulation for time h initialized with particles at the locations $\{y_i\}_{i=1}^N$. With this oracle in hand, we may hope to apply the Euler-type method described

in Section 2.2.3 to μ^N. Specifically, if

$$\mathbf{v}_t^{h,N}(x_i(t)) := \frac{T_{\mu_t^N}^{E_h(\mu_t^N)}(x_i(t)) - x_i(t)}{h} \qquad i = 1, \ldots, N, \tag{2.3.0.1}$$

$$\widetilde{\mu}_{t+H}^N := \left(\mathrm{id} + H\mathbf{v}_t^{h,N}\right)_{\#} \mu_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{x_i(t)+H\mathbf{v}_t^{h,N}(x_i(t))}, \tag{2.3.0.2}$$

then we may hope that $\widetilde{\mu}_{t+H}^N \approx \mu_{t+H}^N$. Unfortunately, none of the theory from Section 2.2 applies in this setting: μ_t^N clearly does not have a density, and in many systems of interest, the particles' evolution is too rough in time for the curve μ^N to be absolutely continuous. (Suppose x_i is, e.g., a Brownian motion.) In particular, μ^N has no tangent field, so there is no reason to believe that $\mathbf{v}^{h,N}$ can tell us anything about the true value μ_{t+H}^N. The key idea of our examples is to sidestep this major issue by assuming that μ^N is an empirical estimate of a "nice" curve μ with velocity field $\mathbf{v}$. In this case, $\mu^N \approx \mu$ for large N, and the hope is that $\mathbf{v}^{h,N} \approx \mathbf{v}$ in such a way that ensures $\widetilde{\mu}_{t+H}^N \approx \mu_{t+H}^N$. While work on proving precise results of this form is ongoing, we give empirical evidence that the approach works in Sections 2.4, 2.5, and 2.6. At a heuristic level, even though no tangent field describes the evolution of μ_t^N on short time scales, there is still a notion of a "direction" — determined by μ — in which the *group* of particles is traveling over longer time scales. The idea is that optimal transport maps allow us to access that direction.

As an illustrative example, suppose the x_i are independent, standard d-dimensional Brownian motions, the smooth curve "in the background" is $t \mapsto \mu_t := N(0, tI_d)$, and the oracle is given by an Euler–Maruyama simulation of Gaussian diffusion:

$$E_h(\nu) = \frac{1}{N}\sum_{i=1}^N \delta_{y_i+\sqrt{h}Z_i}, \tag{2.3.0.3}$$

where $\nu = \frac{1}{N}\sum_{i=1}^N \delta_{y_i}$ and $Z_1, \ldots, Z_N \sim \mathcal{N}(0, I_d)$ are i.i.d. By 2.2.2.11, the tangent field $\mathbf{v}$ of μ is $\mathbf{v}_t(x) = \frac{x}{2t}$. Now, see Figure 2.1, in which $d = 2$. In the first image, an arrow is drawn

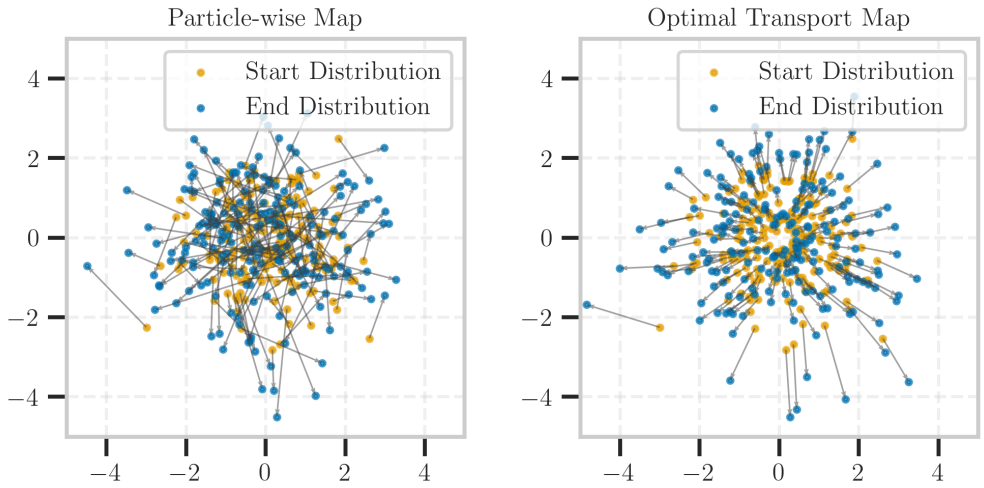


Figure 2.1: Example of the difference between the optimal transport map and the particle-wise map. In the particle-wise image (left), an arrow is drawn to connect each particle's starting position to its ending position. In the optimal transport image (right), we compute the OT map between the starting and ending distributions and draw an arrow connecting each particle's starting location to its corresponding output under the OT map.

from $x_i(1)$ to $x_i(1) + Z_i$, and the arrows are expectedly chaotic. In contrast, the arrows in the second image are drawn from $x_i(1)$ to $x_i(1) + \mathbf{v}_1^{1,N}(x_i(1))$ in accordance with our measure-valued Euler-type method, which reveals the obvious structure of the evolution: The bulk is drifting away from the center in all directions, which agrees with what (2.2.2.11) suggests.

2.3.1 Choosing an Appropriate Step Size

In the "smooth" case, the error of the approximation $\mu_{t+H} \approx (\mathrm{id} + H\mathbf{v}_t^h)_{\#}\mu_t$ generically decreases as h decreases. In the discrete case, however, the roughness of our particles' evolutions introduces a competing need to keep h sufficiently large. To explain why, recall from Proposition 1.1.1.5 that optimal transport maps in the discrete case are given by permutations. Assuming (as we always will) that the particles $\{x_i\}_{i=1}^N$ evolve continuously in time, if h is small enough, then the relevant permutation becomes the identity. More specifically, if h is very small, then running our micro-scale simulation for time h initialized at $x_i(t)$ will result in a particle $y_i^h \approx x_i(t+h)$ that is very close to $x_i(t)$. In this case, the map

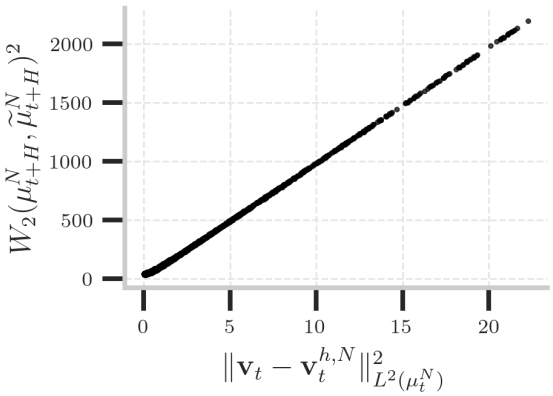


Figure 2.2: Vector field approximation error $\|\mathbf{v}_t - \mathbf{v}_t^{h,N}\|^2_{L^2(\mu_t^N)}$ versus Euler step approximation error $W_2(\mu_{t+H}^N, \widetilde{\mu}_{t+H}^N)^2$ for Brownian motion with $t = 1$ and $H = 99$ (h and N vary).

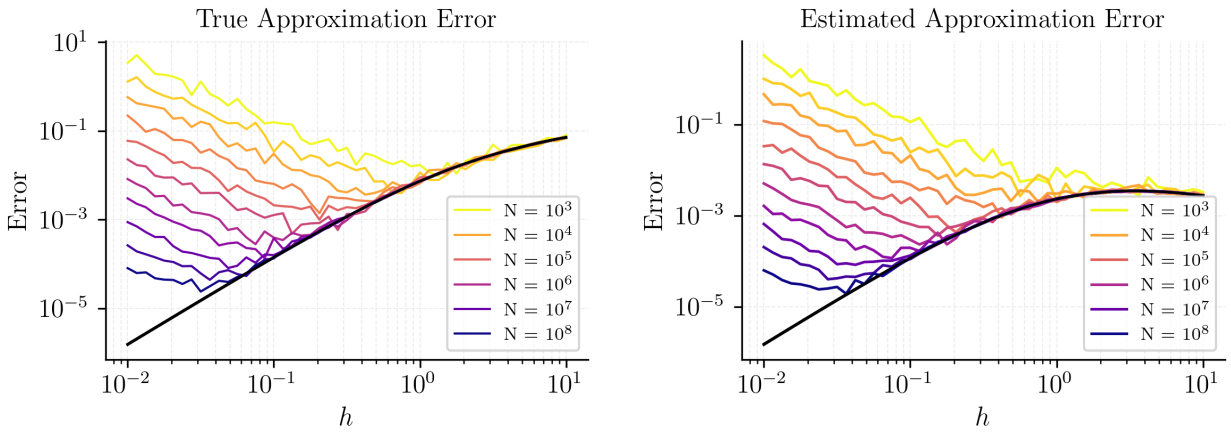
$x_i(t) \mapsto y_i^h$ will optimally transport μ_t^N to $E_h(\mu_t^N)$. Therefore,

$$\mathbf{v}_t^{h,N}(x_i(t)) = \frac{T_{\mu_t^N}^{E_h(\mu_t^N)}(x_i(t)) - x_i(t)}{h} = \frac{y_i^h - x_i(t)}{h} \approx \frac{x_i(t+h) - x_i(t)}{h},$$
$$\widetilde{\mu}_{t+H}^N = \frac{1}{N}\sum_{i=1}^N \delta_{x_i(t)+H\mathbf{v}_t^{h,N}(x_i(t))} \approx \frac{1}{N}\sum_{i=1}^N \delta_{x_i(t)+H\frac{x_i(t+h)-x_i(t)}{h}}.$$

Since x_i is not necessarily differentiable at t, $x_i(t) + H\frac{x_i(t+h)-x_i(t)}{h}$ is not necessarily a good approximation of $x_i(t+H)$, so $\widetilde{\mu}_{t+H}^N$ is not necessarily a good approximation of μ_{t+H}^N if h is too small.

To empirically verify the intuition that small choices of h result in poor approximations, we return to the case of Brownian motion. Specifically, suppose again that the x_i are independent Brownian motions in $\mathbb{R}^d$, $\mu_t = \mathcal{N}(0, tI_d)$, $\mathbf{v}_t(x) = \frac{x}{2t}$, and the oracle is given in terms of an Euler–Maruyama simulation. We have observed experimentally that the approximation error $W_2(\mu_{t+H}^N, \widetilde{\mu}_{t+H}^N)^2$ is strongly correlated with the error $\|\mathbf{v}_t - \mathbf{v}_t^{h,N}\|^2_{L^2(\mu_t^N)}$ in the approximation of the underlying tangent field (Figure 2.2), so we focus the remaining discussion on the latter quantity for computational convenience.

There are two distinct sources of error in the tangent field approximation. The first is the error in the approximation of a derivative by a finite difference with step



(a) True approximation error: The black curve is the theoretical finite difference error $\|\mathbf{v}_t - \mathbf{v}_t^h\|^2$ versus the step size h. The colored curves are the discrete approximation errors $\|\mathbf{v}_t - \mathbf{v}_t^{h,N}\|^2$ for various values of N.

(b) Estimated approximation error: The black curve is the theoretical finite difference error $\|\mathbf{v}_t^h - \mathbf{v}_t^{2h}\|^2$ versus the step size h. The colored curves are the discrete approximation errors $\|\mathbf{v}_t^{h,N} - \mathbf{v}_t^{2h,N}\|^2$ for various values of N.

Figure 2.3: Results of vector field error computations for Brownian motion and comparison with an estimated error using the difference between consecutive approximations.

$h > 0$; the second is the error in using a discrete optimal transport map to approximate a continuous one. As suggested above, the first should dominate for large values of h, and the second should dominate for small values of h. To isolate their effects, we also consider the finite-difference approximation of the tangent field $\mathbf{v}$,

$$\mathbf{v}_t^h(x) = \frac{T_{\mu_t}^{\mu_{t+h}}(x) - x}{h} = \frac{\sqrt{\frac{t+h}{t}} - 1}{h} x,$$

and compute $\|\mathbf{v}_t - \mathbf{v}_t^h\|_{L^2(\mu_t^N)}$ as a proxy for the error caused by the finite-difference approximation. In Figure 2.3a, we plot $\|\mathbf{v}_1 - \mathbf{v}_1^h\|_{L^2(\mu_1^N)}$ in black and $\|\mathbf{v}_1 - \mathbf{v}_1^{h,N}\|_{L^2(\mu_1^N)}$ for different values of h, N in various other colors. We note that for each value of N, there is an approximately optimal value of h that minimizes the error and that this optimal value of h decreases as N increases. We also see that for very small values of h, the error is very large (as predicted above) and that for larger values of h, the discrete approximation is about as good as the true finite difference approximation. This indicates that, indeed, as

the step size increases, the optimal transport map better captures the distribution's bulk behavior, and the limiting factor on choosing h too big is the fact that we are approximating a derivative using a finite difference. While these experiments do not precisely formalize this behavior, they corroborate the assertion that there should be a "happy-medium" value of h that decreases as the number of samples increases.

We expect this phenomenon to appear in most random particle systems one would be interested in studying using optimal transport. Indeed, there is evidence that this phenomenon is due to a gap in the eigenvalue spectrum of the macroscopic, coarse equation that governs the dynamics "in the background" [GK03a, RMGK03]. However, as always, we are considering the setting in which we do not know these "underlying" dynamics, and so we cannot *analytically* determine an appropriate value for h. As such, one might hope to *empirically* determine the optimal choice for h to use in the prediction algorithm. Unfortunately, the previous paragraph's analysis was only possible because we had explicit a priori knowledge of $\mathbf{v}$, which defeats the purpose of approximating it. A different approach is therefore needed to determine the optimal step size h. To this end, we perform another experiment to assess the viability of a simple intrinsic criterion: how much the approximation of the vector field changes after another step of size h, i.e., $\|\mathbf{v}_1^{h,N} - \mathbf{v}_1^{2h,N}\|_{L^2(\mu_1^N)}$. In Figure 2.3b, we compare $\|\mathbf{v}_1^{h,N} - \mathbf{v}_1^{2h,N}\|_{L^2(\mu_1^N)}$ with $\|\mathbf{v}_1^{h} - \mathbf{v}_1^{2h}\|_{L^2(\mu_1^N)}$, the "normal" difference between two consecutive finite-difference approximations. While the correspondence is not perfect, the plots are remarkably similar, which suggests the method's potential use.

Remark 2.3.1.1. *On a practical note, if the step size of the micro-scale simulator is fixed (e.g., as it is in the model used in Section 2.4, see Remark 2.4.2.1), then one can effectively increase h by taking E_h to be the result of applying* multiple *micro-scale steps in succession. In that case, we think of h as a "burst period" because we do a short burst of successive micro-scale steps. This means we always have the flexibility to increase the effective value of h. Thus, it still makes sense to discuss how to tune h appropriately, even if the micro-scale simulator itself has a fixed time step.*

In our experiments in Sections 2.4, 2.5, and 2.6, we fix this "burst period" as a parameter based on data from previous experiments; however, one could easily imagine adaptations to the algorithm in Section 2.3.3 that dynamically adjust h using this heuristic.

2.3.2 Additional Considerations

Statistical Averaging When working with random particle systems, it is often advantageous to replace the single finite difference $\mathbf{v}_t^h$ with an average of many finite differences. There are many ways to do this, but perhaps the simplest is to average finite differences computed with a range of different time steps over some interval. Also, if we take the middle of this interval to be the time of interest, then taking an average of both forward and backward differences becomes equivalent to taking an average of centered differences, which are typically more accurate numerically. Explicitly, for a fixed time step h and discrete times $t_0 - kh =: t_{-k}, \ldots, t_0, \ldots, t_k := t_0 + kh$, we set

$$\mathbf{v}_{t_0}^{h,N,k}(x_i(t_0)) := \frac{1}{2k} \sum_{\substack{j=-k \\ j\neq 0}}^{k} \mathbf{v}_t^{jh}(x_i(t_0)) = \frac{1}{k} \sum_{j=1}^{k} \frac{T_{\mu_{t_0}^N}^{\mu_{t_j}^N}(x_i(t_0)) - T_{\mu_{t_0}^N}^{\mu_{t_{-j}}^N}(x_i(t_0))}{2jh}.$$

This vector field is the one we use in practice in the experiments in Sections 2.4, 2.5, and 2.6.

Burn-in Period To approximate the particle distribution μ_t^N over a long time, we need to perform many steps of our Euler-type method. Of course, this involves re-initializing the micro-scale simulation many times, once after each step. One potential obstacle of this approach is that the approximate distribution after an Euler step will inevitably be slightly different from the true distribution, and running the micro-scale simulation on the approximate distribution might result in behavior that is initially dominated by the approximate distribution's return to a relative equilibrium. To account for this and ensure the approximated velocity fields capture mostly the long-term dynamics, we allow the

approximate distribution to evolve according to the micro-scale simulation for a prescribed "burn-in" time H_R before starting to take samples to approximate the velocity field at time $t + H + H_R$. This ensures that any fringe effects from the Euler step are washed away before the next Euler step. For a more theoretical discussion of this intuitive motivation for the burn-in period, we defer to [GK03a, GKKZ05, ZVG+12].

We also note that the burn-in period also helps to avoid the pathologies that can arise from using Euler's method on, e.g., Wasserstein gradient flows [XL24]. For example, there may be locations with a high density of approximated particles (or even particles with exactly the same approximated location), which can result in irregularities in the subsequent vector field approximation. By allowing the particle system to "burn in" after each approximating step, we ensure that the particles' density is, loosely speaking, smoothed out. For an illustration of how the burn-in time allows the density to smooth, see Figures 2.9 and 2.10. Empirical tests indicate that this burn-in is essential for our method to work well. Understanding these benefits theoretically is a subject of ongoing work.

2.3.3 Detailed Description of Our Algorithm

Table 2.1: Parameter definitions.

Parameter	Description
h	the time of one step of the micro-scale simulation
k	the number of centered differences to average
$H_S := kh$	*half* the length of the derivative-sample time window
N_T	the number of Euler-type prediction steps
H	the length of each Euler step
H_R	the burn-in time after each Euler step
$\Delta := H_S + H + H_R$	the time difference between Euler steps
S	the length of start-up time
R	the length of final recovery time after all Euler steps

We now present our complete algorithm for approximating the evolution of a particle

system given by a particular micro-scale simulation E_h (e.g., the micro-scale simulation could be an Euler-Maruyama numerical simulation of an SDE as in (2.3.0.3), a biological system as in Section 2.4, or any other method for evolving particles for a short time). For a particular initial distribution of particles μ_0, our algorithm approximates the distribution μ_T^N for large T with all the additional considerations discussed in this section. We fix values for the parameters as defined in Table 2.1 and then:

1. Run the micro-scale simulation for the start-up time S (by taking S/h micro-scale steps) to allow the initial distribution μ_0 to reach a relative equilibrium (if necessary). Call the resulting distribution $\mu_{(-1)+H_R}$.

2. For each approximating step $n = 0, \ldots, N_T$:

   (a) Set $\mu_{(n)-kh} := \mu_{(n-1)+H_R}$, and take $2k$ micro-scale steps to get distributions

   $$\mu^N_{(n)-kh}, \ldots, \mu^N_{(n)}, \ldots, \mu^N_{(n)+kh}.$$

   (b) For each x_i in the support of $\mu^N_{(n)}$, compute

   $$\mathbf{v}^{h,N,k}_{(n)}(x_i) := \frac{1}{k} \sum_{j=1}^{k} \frac{T^{\mu^N_{(n)+jh}}_{\mu^N_{(n)}}(x_i) - T^{\mu^N_{(n)-jh}}_{\mu^N_{(n)}}(x_i)}{2jh}.$$

   (c) Set

   $$\mu^N_{(n)+H} := \left(\mathrm{id} + \mathbf{v}^{h,N,k}_{(n)}\right)_{\#} \mu^N_{(n)}.$$

   (d) Take H_R/h micro-scale steps to allow the distribution to burn in. Call the resulting distribution $\mu^N_{(n)+H_R}$.

3. Run the micro-scale simulation for some final recover time R (by taking R/h micro-scale steps) to ensure any fringe errors in the approximations are minimized (e.g., that the distribution has indeed reached a true steady state).

While the notation used above is convenient for highlighting the recursive nature of the algorithm, it becomes unwieldy when discussing how to interpret the approximations. As such, we set the following notation convention:

Notation 2.3.3.1. *Let $\widetilde{\mu}_t^N$ denote the approximation produced by the algorithm that corresponds to time t.*

Explicitly, if $t_n := S + n(H_S + H + H_R) + H_S$, then $\widetilde{\mu}_{t_n}^N = \mu_{(n)}^N$ is the approximation we use to compute the Euler step, $\widetilde{\mu}_{t_n+H}^N = \mu_{(n)+H}^N$ is the approximation immediately after the Euler step, and $\widetilde{\mu}_{t_n+H+H_R}^N = \mu_{(n)+H_R}^N$ is the approximation after the burn-in period. This notation is convenient because $\widetilde{\mu}_t^N$ is an approximation of μ_t^N for all t for which the former is defined.

Remark 2.3.3.2. *One limitation of this scheme that will become relevant in Section 2.4 is that it only gives a way to approximate future* locations *of particles and not any underlying data. If the micro-scale model has any parameters besides the position of the particles (e.g., the current "underlying state" of a particle), then these data must also be predicted across the Euler-type predicting step taken in step 2(c) above. In general, there is no single method for doing this, but we will discuss ad hoc approaches as they come up in Section 2.4 where we must also predict, for example, the state of bacteria flagella (see Sections 2.4.2 and 2.4.5).*

2.4 Simulation Study I: Biological Agent-Based Model

We now demonstrate the efficacy of our algorithm on a few illustrative example micro-scale models. The first such model we consider is one of bacterial chemotaxis.

2.4.1 Micro-scale Model Description

The specific model we use is described in detail in [SGOK05, LPSK23]. For brevity, we omit the precise details of the (see [SGOK05, LPSK23] for more details), but we provide a

Table 2.2: Parameters for chemotaxis micro-scale simulation.

Parameter	Description
$N = 10{,}000$	the number of bacteria (the number of simulations)
$h = 1/16$	micro-scale time step
$v_{\text{cell}} = 0.003$	swimming speed of each bacteria
$S(x) = \mathcal{N}(\mu = 6.5, \sigma = 1.35)$ $= (1.35\sqrt{2\pi})^{-1} \exp\left(-\frac{(x-6.5)^2}{2(1.35)^2}\right)$	chemoattractant profile

high-level overview here to illustrate its computational complexity (and hence the desire for an algorithm to predict its evolution). Each cell's mobility is affected by a chemoattractant profile (specifically, its spatial derivative) which influences the "rotations" of the cell's six "flagella." The model computes the probability that each flagellum changes its rotational direction based on changes of the concentration of the protein CheY-P. The concentration of CheY-P has a time-derivative which is a function variables which evolve according to a joint ODE that itself is a function of the chemoattractant profile. The rotational directions of the flagella determine the motion of the cell at each step — if fewer than three are rotating counter-clockwise, the cell "tumbles," meaning it does not move at that time step. Otherwise, the cell "swims," meaning it moves with constant speed in the direction that it moved in the last time step (or chooses a new random direction if it was tumbling in the last time step). Then to actually execute one time step, the model updates the locations of the cells that swim for that step, and then it updates all underlying variables according to the prescribed derivatives and the value of the chemoattractant profile at the new location.

Due to its intricate nature, many more operations are needed to execute one step of this model than, e.g., a standard SDE Euler-Maruyama model. Thus, our approximation algorithm provides an appreciable computational improvement by allowing us to skip many of these expensive micro-scale steps.

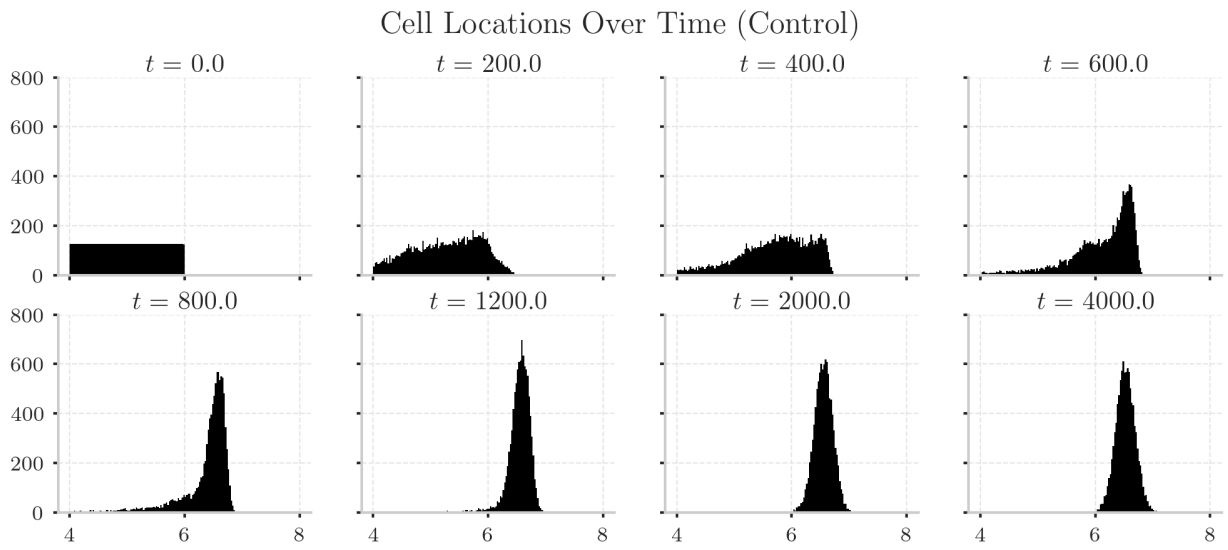


Figure 2.4: Histograms of bacteria locations at various times t from $t = 0$ to $t = 4000$ for the control simulation of chemotaxis.

2.4.2 Experimental Results

For reference, we run the simulation without prediction to establish a "control" as the ground truth behavior of the system. We initialize the cell locations $x_i(0)$ to be evenly spaced in the interval $[4, 6]$, and we use the same parameters as in [SGOK05, LPSK23] with only the changes specified in Table 2.2. Histograms of bacteria positions at different times in the control simulation are shown in Figure 2.4. Then, to compare with the control data, we perform a simulation using the approximation algorithm with the additional parameters in Table 2.3. Note that we choose these parameters (in particular, the number of centered differences k — which effectively gives the length of the "burst" period discussed in Remark 2.3.1.1 — and the length of the Euler-type prediction step H) according to the heuristics discussed in Section 2.3.1. For simplicity, we choose to fix these parameters at the beginning based on previously-observed data about the micro-scale model, but one could design algorithms that make these choices adaptively based on, e.g., how quickly the computed velocity field is changing — the criterion discussed at the end of Section 2.3.1.

To continue simulating after each Euler step, the micro-scale simulator requires not

Table 2.3: Parameters for approximation algorithm with chemotaxis micro-scale simulation.

Parameter	Description
$S = 200$	length of start-up time
$k = 80$	number of centered differences to average
$H = 95$	length of each Euler-type prediction step
$H_R = 100$	burn-in time after each Euler step
$N_T = 19$	number of Euler steps
$R = 0$	length of final recovery time after last Euler step

only current bacteria locations but also all microscopic quantities. However, as discussed in Remark 2.3.3.2, the algorithm described in Section 2.3.3 predicts only the *locations* of bacteria after each Euler step, resulting in a lack (or missing) of microscopic quantities. Hence, after the predicting time step, the suggested algorithm requires the re-initialization of microscopic quantities, such as u_1, u_2, and $s^{(i)}$. Even though some (quantitatively bad but) qualitatively good re-initializations quickly reach slow dynamics, we suggest a better re-initialization scheme based on the history of microscopic quantities and demonstrate a more accurate prediction of bacteria density distribution over multiple Euler steps.

After each Euler step, we re-initialize the underlying data according to the history-dependent scheme in Table 2.4. For example, given data at time t_0, we use the optimal transport approximation scheme to predict the bacteria locations at $t_0 + H$ and re-initialize the microscopic properties at $t_0 + H$ as follows:

- Set $u_1 = 0$ for all bacteria,
- compute $u_2'(t_0) = \frac{du_2}{dt}\big|_{t=t_0}$ from the prescribed ODE in [SGOK05] to take a forward Euler step $\widetilde{u}_2(t_0 + H) := u_2(t_0) + H \cdot u_2'(t_0)$, and
- keep the flagella rotating direction and bacteria moving direction exactly the same as they were at time t_0.

To qualitatively inspect the accuracy of the approximations, we show histograms of

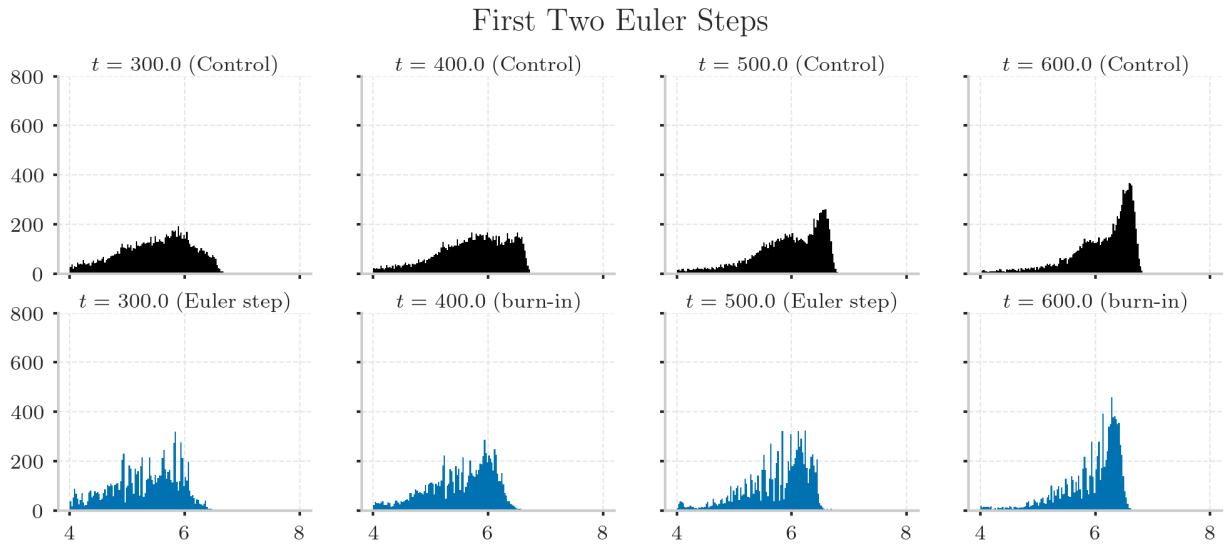


Figure 2.5: Histograms of bacteria locations after first two Euler steps (bottom) compared with corresponding times from control simulation (top). We run the micro-scale model until $t = 200$ (not pictured), take an Euler step to $t = 300$ (leftmost), burn in until $t = 400$ (second from left), take an Euler step to $t = 500$ (second from right), and burn in until $t = 600$ (rightmost).

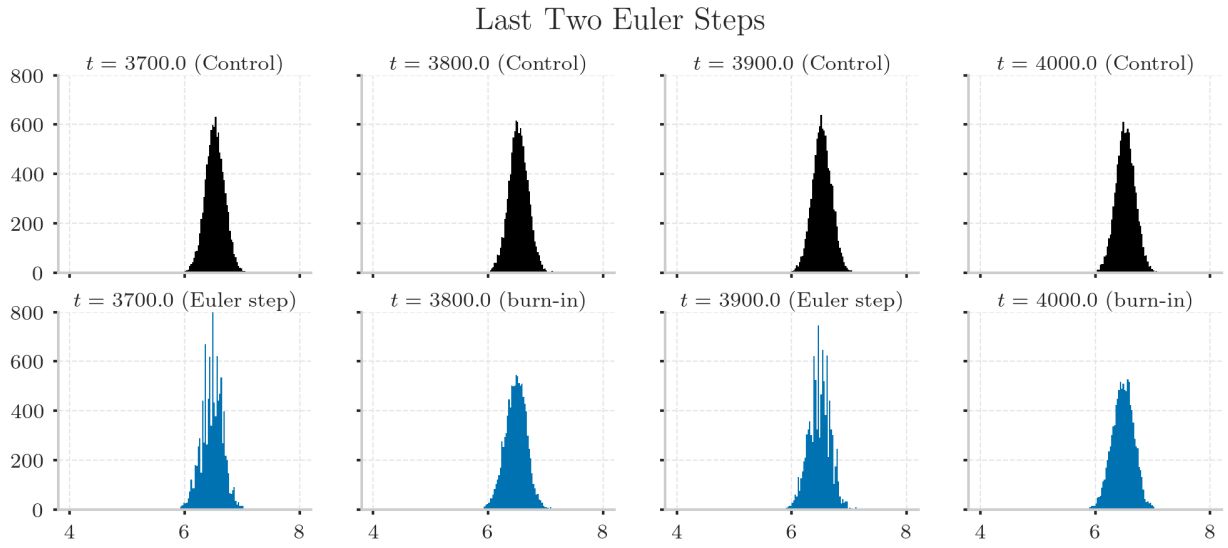


Figure 2.6: Histograms of bacteria locations after final two Euler steps (bottom) compared with corresponding times from control simulation (top). We take an Euler step from $t = 3600$ (not pictured) to $t = 3700$ (leftmost), burn in until $t = 3800$ (second from left), take an Euler step to $t = 3900$ (second from right), and finally burn in until $t = 4000$ (right-most), the end time.

the control and approximated distributions of the bacteria locations after the first two and last two Euler steps in Figures 2.5 and 2.6, respectively. We see that the approximations match reasonably closely with the control distributions at the corresponding times. In particular, we see that our algorithm approximates the correct steady-state distribution, as evidenced by Figure 2.6 — the last two Euler steps do not change the approximated distribution, meaning the velocity field is approximately zero, which is what it should be for a distribution in a steady state.

To quantitatively assess the accuracy of our approximations, we create two plots: one which shows the approximation error over time, and one which depicts (a dimension-reduced version of) the control distribution's trajectory and the results of each Euler step. The resulting plots for the bacterial chemotaxis experiment are Figures 2.7a and 2.7b, but since these will be used a few times throughout this chapter, we describe their construction in general here.

Approximation Error Plot To assess the numerical accuracy of a particular approximation method, we compute the Wasserstein distance between the control distribution μ_t^N and the approximated distribution $\widetilde{\mu}_t^N$ for all times t at which we have an approximated distribution (i.e., for the times during the burn-in and velocity field estimation periods). We do not compute this error during the period accounted for by the Euler-step itself, as the point of our algorithm is to take a step forward in time without explicitly generating an approximated distribution for those skipped time steps. We then plot these distances $W_2(\mu_t^N, \widetilde{\mu}_t^N)$ versus time t for the appropriate times.

Trajectory Plot We can also visualize how well the approximations track the true curve $t \mapsto \mu_t^N$ by embedding into a common linear space and performing a dimension reduction. Specifically, we (1) fix a common (discrete) reference measure σ (in our case, by choosing $N = 10{,}000$ independent samples of a standard Gaussian); (2) compute optimal transport maps $T_\sigma^{\mu_t^N}$ and $T_\sigma^{\widetilde{\mu}_t^N}$; (3) vectorize $T_\sigma^{\mu_t^N}$ and $T_\sigma^{\widetilde{\mu}_t^N}$ by interpreting them as $N \times d$ matrices,

where N is the number of particles in σ and d is the ambient dimension (two in our case); and (4) perform PCA to project onto a two-dimensional subspace. By doing this, we get a (dimension-reduced) depiction of the distribution's trajectory over time, which allows us to visually inspect how well the approximation method's Euler steps track the control trajectory. We plot the control curve in a rainbow color, where the color represents the corresponding time. To avoid cluttering the plot, we only plot each approximation after the burn-in period. We can interpret this as the result of one complete Euler step: we use the data near one dot to estimate a velocity field (tangent vector to the curve), take an Euler-step forward, and then burn-in. Thus, the points that correspond to the approximations should be seen as a visual representation of Euler steps, and an approximation method is "better" if it more closely follows the control distribution as it evolves over time and reaches the same steady state.

Remark 2.4.2.1. *We note that the micro-scale model used in this example actually breaks down for h much larger than our choice of $1/16$. For example, with the rest of the model parameters fixed, the computation of the underlying variables as described in [SGOK05] fails badly for $h = 1/4$. In particular, there is an assumption in [SGOK05] that $h \cdot k_{\pm} \ll 1$, and by choosing $h = 1/4$, the values of $k_{\pm}$ over time become many orders of magnitude larger than 1, which means this assumption is grossly false. Through trial and error, we determine that $h = 1/16$ is about the largest time step that preserves the above inequality for the entire simulation. This means that we really cannot improve the efficiency of the micro-scale model by making the time step larger — the time step must be kept sufficiently small to preserve the model of the micro-scale dynamics, but we still need to simulate the model for a long time to observe the steady-state behavior, which is exactly the motivating archetype of this work. For more examples of models with a similar property, and a more detailed discussion of the resulting tradeoff between accuracy, stability, and computational cost, we defer the reader to [GK03a, RMGK03].*

2.4.3 Computational Improvement

As we briefly discuss in the introduction, we are not interested in the precise implementation of the micro-scale model, and instead we are interested in demonstrating that our approach does indeed reduce the number of micro-scale steps needed to accurately take a "macro" time step. Because of this, we illustrate the computational improvement of our method by computing its reduction in required micro-scale steps. With the parameters defined above, our algorithm requires 11/20 as many micro-scale steps as the equivalent control simulation (10 seconds of velocity-field-approximation and 100 seconds of burn-in time for every 200 seconds of simulated time). Importantly, the length of each micro-scale burst to approximate the velocity field is 1/10 as long as the corresponding Euler-step, and so ignoring burn-in, our "macro-scale timestepper" only requires 1/10 as many micro-scale steps as running the simulation to the equivalent time. Thus, the primary way the overall improvement could be increased is by decreasing the burn-in time. We have experimentally observed that with our choice of re-initialization, the burn-in time required for the underlying variables to return to relative equilibrium scales roughly with the size of the Euler step. This means that achieving a better ratio of required micro-scale steps likely requires a better approach to re-initializing the underlying variables. We have seen in further experiments (omitted for brevity) that the burn-in time can be reduced considerably if the underlying variables are re-initialized perfectly, e.g., by using data from the control simulation. This suggests that the current barrier to more significant improvements is the quality of re-initializations, not the quality of the distribution approximations. We demonstrate the importance of the re-initialization scheme in Section 2.4.5.

2.4.4 Comparison with Alternative Methods

Particle-wise approximation To demonstrate the value of using optimal transport in our algorithm, we compare our method with a similar method that uses a different approach

to learning the underlying dynamics. Our method uses a short burst of micro-scale steps to gather data that is used to approximate a velocity field, so a natural question is whether the same data could be used more effectively; i.e., is the apparent success of our method due to optimal transport, or is it simply the case that there is enough data in the short burst that any reasonable approach would be just as successful?

One extremely naive approach is to attempt to approximate the velocity field without using optimal transport. Specifically, we can perform the finite-difference calculations on the individual cells:

$$x_i^{[1],h,k}(t) \coloneqq \frac{1}{k}\sum_{j=1}^{k}\frac{x_i(t+jh)-x_i(t-jh)}{2jh}, \qquad i=1,\dots,N.$$

Effectively, this attempts to approximate $x_i'(t)$ for each individual cell, which we could then use in an Euler-type approximation

$$\widetilde{x}_i(t+H) \coloneqq x_i(t) + Hx_i^{[1],h,k}(t), \qquad i=1,\dots,N.$$

As explained in Section 2.3.1, this approach should result in a much worse approximation of the distribution's overall evolution. Our experiments agree with this prediction. We run exactly the same approximation algorithm with the same parameters as above, with the only change being that we use the particle-wise finite differences as the approximated velocity field rather than the OT finite differences. In Figure 2.7a, we see that the W_2 distance between the control and approximated distributions is substantially higher, and in Figure 2.7b, we see that the approximations fail to closely track the control distribution's trajectory. This is a naive approach, but it validates the intuition that optimal transport really is doing something nontrivial in the algorithm.

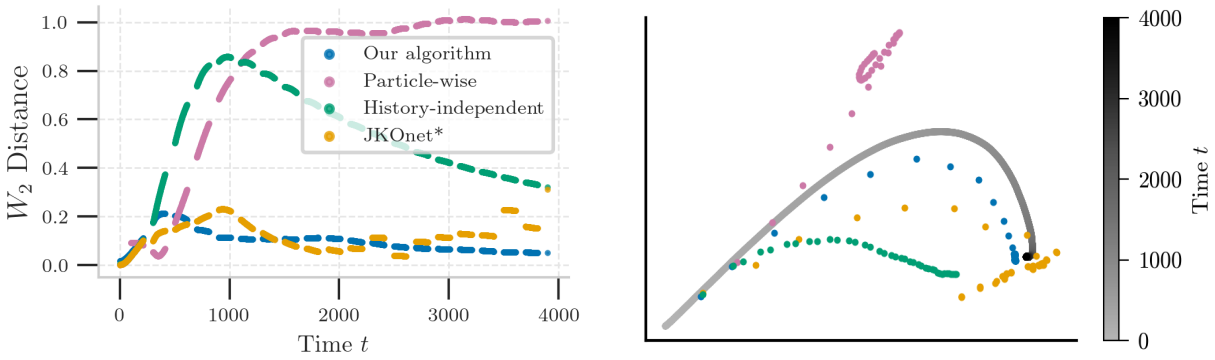
JKOnet* approximation A much more reasonable alternative to our method is to attempt to learn an SDE that approximates the observed behavior. That is, instead of using the short

burst of micro-scale steps to approximate a velocity field, we could use that data to learn an SDE whose behavior closely mimics the observed cell behavior from the micro-scale model. With this SDE in hand, we could hope to approximate the cells' distribution a long time in the future by running the learned SDE for the equivalent time. Since simulating SDEs is computationally cheap, this would also give significant computational improvements by replacing many expensive micro-scale steps with cheap SDE steps.

To evaluate whether this approach is viable, we perform a similar experiment (again with the same parameters as above, including start-up, burn-in, and final recovery), except we use the data from the $2k + 1$ micro-scale steps in the micro-scale burst to learn an SDE rather than approximate the velocity field. To learn the SDE, we use JKOnet* [TLGD24], which is a particularly successful method for learning SDEs given particle data at successive times. We use this data to learn[1] the potential energy of an approximating underlying SDE, and use those learned parameters to simulate an SDE with initial particle positions equal to the locations of the cells in the last cell distribution from the micro-scale burst. We simulate this SDE for the same length of time as our "Euler-type prediction step" ($H = 95$ in the case of the experiment above), and reinitialize our micro-scale chemotaxis model with cells at the locations of the particles after the SDE simulation. We use the same re-initialization scheme for all the underlying data, meaning the only difference between our method and using JKOnet* is how we predict the cell locations at time $t + H$ in each Euler-type step.

In Figure 2.7a, we produce a comparison of the W_2 distance between the control and approximated distributions over time for the two different methods, and in Figure 2.7b, we perform PCA to visualize how well the approximations track the trajectory of the control evolution. We see that the JKOnet* approach performs substantially worse than our method — Figure 2.7a shows that the approximation errors are larger than for our method, especially towards the end of the simulation when the control distribution

[1]To learn the potential, we use code in the `train.py` file from [TLGD24]. We use the `jkonet-star-potential` model with 1000 epochs, and the training data we use is exactly the particle data from the $2k + 1$ micro-scale steps in the micro-scale burst after the burn-in period (exactly the same data we use to approximate the vector field in our method).



(a) 2-Wasserstein distance between control distribution and approximated distributions versus time. Note that we only plot points during the burn-in and velocity-field estimation periods; we skip the intervals that correspond to each Euler-step.

(b) Projections of distribution trajectories onto the two principal components of the control distribution's evolution over time. The solid rainbow curve shows the control distribution, where the color represents time according to the color bar at the right (the black X marks steady state). Each colored dot represents one Euler step for the corresponding approximation method.

Figure 2.7: Comparisons between control distributions and approximations. The four approximation schemes are as follows: the blue is our algorithm, as described in Section 2.4.2; the purple is using the particle-wise finite differences, as described in the first half of Section 2.4.4; the green is using the JKOnet* method for learning and simulating an SDE, as described in the second half of Section 2.4.4; the orange is using our algorithm with the history-independent re-initialization scheme, as described in Section 2.4.5.

has reached steady state, and Figure 2.7b shows that our method more closely follows the trajectory of the control distribution and reaches approximately the same steady state (whereas the JKOnet* method is less stable near steady state). The reason the equivalent JKOnet* method performs worse than our method is because learning an SDE typically requires simulation data from many more times (and in particular, from a wide range of times during the evolution). In our case, we only have access to simulation data from the short micro-scale burst of $2k + 1$ steps, which as a time interval is a small fraction of the total length of time we wish to simulate. It is this restriction to only *local* data that makes the SDE-learning approach so prone to errors in this context. This experiment shows how our method provides a tradeoff between the need for data and the generality of the object being learned — less data is required if one is only interested in understanding how a particular distribution evolves according to the micro-scale model rather than if one hopes to learn a model that captures the global dynamics of the model. If we wanted an SDE that could quickly simulate the evolution of *any* initial distribution, then we could run many more micro-scale steps and then use JKOnet* to learn an appropriate SDE. However, our goal is not to learn a model for how the model *would* act on *any* distribution — we are only interested in developing a method for taking a *particular* initial distribution and predicting its evolution using as few micro-scale steps as possible.

We should note that the JKOnet* algorithm is asymptotically slightly better than our algorithm — JKOnet* is $O(MTN)$, where M is the number of epochs, T is the length of the training trajectory, and N is the number of particles [TLGD24], whereas our algorithm is $O(TN \log N)$ since computing an optimal transport map at each timestep requires sorting the N particle locations. Even still, we point out that one of the major advantages of our method is the general class of systems to which it can be applied. JKOnet* (and other similar methods) rely on the system evolving according to some energy-minimizing trajectory, which is not always the case for arbitrary agent-based models. For systems which are not energy-minimizing in a precise sense, these methods break down. Our method, however,

does not rely on any such assumptions about the underlying system, which is one of its primary advantages.

2.4.5 Comparison with Different Re-initializations

In section 2.4.3, we briefly allude to the importance of the method by which we re-initialize the underlying variables across each Euler-type step. To demonstrate the efficacy of our suggested re-initialization described in Section 2.4.2, we also perform a history-independent re-initialization as described in [SGOK05]. Table 2.4 shows the different parameters of re-initialization between the two schemes. In Figure 2.7a, we provide a comparison of the W_2 distance between the control and approximated distributions over time for the two different approximation schemes, and in Figure 2.7b, we perform PCA to visualize how well the approximations track the trajectory of the control evolution.

Table 2.4: Parameters of different re-initialization schemes. The "history-dependent" scheme (as suggested in this chapter) retains the flagella states from the micro-scale simulation immediately before the Euler step, and it attempts to predict u_2 via an Euler forward step. The "history-independent" scheme (described in [SGOK05]) sets a fixed number of flagella to rotate counterclockwise (here, 0), and u_2 is set solely as a function of the predicted location of bacteria.

Parameters	History-Dependent Scheme	History-Independent Scheme
u_1	$\widetilde{u}_1(t_0 + H) := 0$	$\widetilde{u}_1(t_0 + H) := 0$
u_2	$\widetilde{u}_2(t_0 + H) := u_2(t_0) + H \cdot u_2'(t_0)$	$\widetilde{u}_2(t_0 + H) := f(S(\tilde{x}(t_0 + H)))$
$s^{(i)}$	$\tilde{s}^{(i)}(t_0 + H) := s^{(i)}(t_0)$	$\tilde{s}^{(i)}(t_0 + H) := 0$ (fixed)

The first Euler step shows similar results between the two experiments, but the history-independent scheme becomes (qualitatively good but) "relatively" inaccurate by the end of the first burn-in period. This is because the history-independent method re-initializes the microscopic properties with a "fixed" value regardless of current conditions, resulting in a large burn-in time to reach slow dynamics, while the history-dependent scheme uses the current microscopic properties to re-initialize after the next Euler step. In

other words, the underlying variables (particularly u_2 and the flagella rotating direction) could not respond quickly enough to the fixed re-initialization. Furthermore, even though the distributions look the same after the first Euler step, the two methods still yield different subsequent microscopic behavior, leading to slightly different slow dynamics.

The difference in the two methods' approximation accuracy suggests that the underlying microscopic properties are important components in the micro-scale simulation. Hence, different settings of the properties result in different overall distribution behavior. In the language of Section 2.2, the tangent field that describes the distribution's evolution is somehow a function of these variables, not merely a function of position, so the method we choose to predict them across Euler steps is critical to our capability to approximate the evolution using optimal transport maps. Like all mathematical models, choices about which prediction methods should be used depend entirely on the needs of the particular application. Unfortunately, these choices are mostly ad hoc and context-specific, so there is little to say in general about how they impact the accuracy of the overall approximation algorithm.

2.5 Simulation Study II: Partial Differential Equation

To further demonstrate the ubiquity of our approach, we now perform our algorithm on a particle model which simulates the well known Burgers' equation

$$\partial_t \mu = -\mu \partial_x \mu + \nu \partial_{xx} \mu$$

with kinematic viscosity parameter ν. The micro-scale model we use is described in detail in [AK21], but we provide a brief description here. At each discrete time step of size h (i.e., $t_{j+1} = t_j + h$), we update the location of the ith particle according to

$$x_i(t_{j+1}) = x_i(t_j) + \frac{mh}{Zd_{i,m}} + \sqrt{2\nu h} W_i$$

Table 2.5: Parameters for Burgers' model micro-scale simulation.

Parameter	Description
$N = 20{,}000$	number of independent particles
$h = 1/4096$	micro-scale simulation step size
$x_i(0) \sim \mathcal{N}(\pi, \frac{1}{2})$ i.i.d. $\quad \forall i = 1, \ldots, N$	initial particle locations
$T = 2$	end time
$Z = 1000$	particle coupling strength
$\nu = 2$	kinematic viscosity
$m = 200$	number of particles used in $d_{i,m}$

where Z and ν are parameters of the motion, $d_{i,m}$ represents the distance between the mth particle to the left and right of our ith particle, and $W_i \sim \mathcal{N}(0, 1)$ are i.i.d standard normal random variables. We also enforce periodic boundary conditions on $[0, 2\pi)$; i.e., the particles are understood to move on the circle, and the computations of $d_{i,m}$ take this into account. For more details about the model and its link to Burgers' equation, we refer the reader to [AK21, LKGK03], but we note that do not ever use the Burgers' equation PDE in computing any of our approximations — we only present it to demonstrate that our method works on particle systems that model PDEs.

Table 2.6: Additional parameters for approximation algorithm with Burgers' model micro-scale simulation.

Parameter	Description
$S = 1/8$	length of start-up time
$k = 48$	number of centered differences to average
$H = 37/256$	length of each Euler-type prediction step
$H_R = 3/32$	burn-in time after each Euler step
$N_T = 7$	number of Euler steps
$R = 1/8$	length of final recovery time after last Euler step

As before, we first run the micro-scale model from $t = 0$ to $t = 2$ with the parameters in 2.5. The initial particle locations $x_i(0)$ are chosen randomly by independently sampling

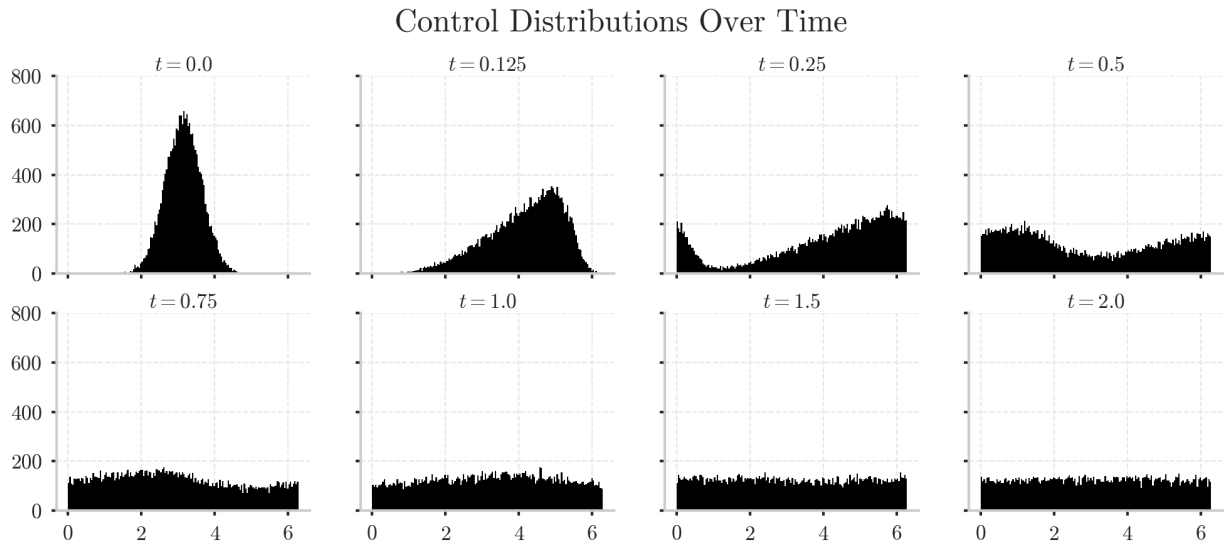


Figure 2.8: Histograms of particle locations at various times t from $t = 0$ to $t = 2$ for the control simulation of the Burgers' model.

from the Gaussian distribution $\mathcal{N}(\pi, \frac{1}{2})$. Histograms of particle locations at different times in the control simulation are shown in Figure 2.8.

We then run the approximation algorithm with the same parameters as in Table 2.5 and the additional parameters (as defined in Section 2.3.3 in Table 2.6). Note that the effective end time of this simulation is $T = S + N_T(H_S + H + H_R) + R = 2$ (since $H_S + H + H_R = 1/4$), which agrees with the control simulation. Figures 2.9 and 2.10 show the distributions obtained after the first two and last two Euler steps (and corresponding burn-in periods) respectively. We observe that, after the burn-in period, the approximated distributions do indeed visually resemble the control distributions at the corresponding times. We also notice that the distributions immediately after the Euler step are much "spikier" than the control distributions. This highlights the importance of the burn-in period — the spiky distribution quickly returns to a relatively smooth distribution that much more closely resembles the control distribution, at which point the velocity field approximation will be much closer to the velocity field describing the control distribution's evolution. If the velocity field were approximated immediately after the Euler step, the effect of the distribution "smoothing out" would dominate the effect of the long-term

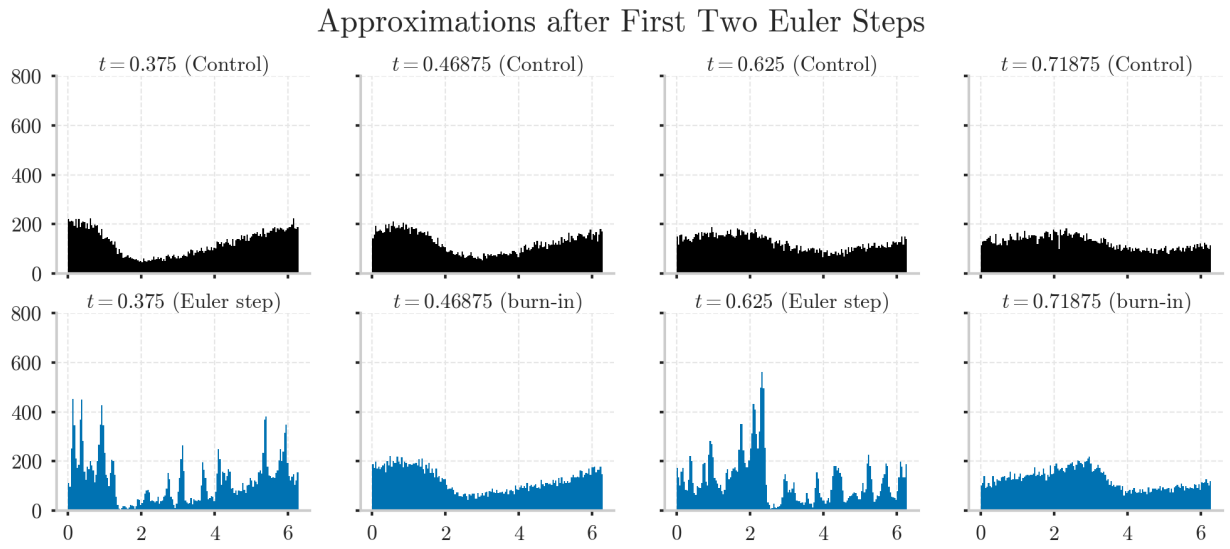


Figure 2.9: Histograms of particle locations after first two Euler steps (bottom) compared with corresponding times from control simulation (top). We run the micro-scale simulation until $t = 11/32$ (not pictured), take an Euler step to $t = 3/8$ (leftmost), burn in until $t = 15/32$ (second from left), take an Euler step to $t = 5/8$ (second from right), and burn in until $t = 23/32$ (rightmost).

Figure 2.10: Histograms of particle locations after final two Euler steps (bottom) compared with corresponding times from control simulation (top). We take an Euler step from $t = 47/32$ (not pictured) to $t = 13/8$ (leftmost), burn in until $t = 55/32$ (second from left), take an Euler step to $t = 15/8$ (second from right), and finally burn in until $t = 2$ (right-most).

evolution, which is the piece we wish to approximate.

Remark 2.5.0.1. *It is worth pointing out that the spikiness observed in the approximated distributions after the Euler step is a result of the approximated velocity field* $\mathbf{v}_t^{k,N,h}$ *not being "smooth enough," as other experimental evidence suggests that smoother velocity fields yield smoother approximated distributions. Thus, this spikiness could likely be reduced by, e.g., first applying a kernel smoothing technique to this velocity field, and doing so could drastically reduce the burn-in time required for the distribution to return to a relative equilibrium.*

To compare the computational cost of our method compared to the control simulation, we once again compute the number of micro-scale steps needed. Each step of our algorithm only requires 15/32 as many micro-scale steps as the control simulation (3/128 seconds of velocity-field estimation and 3/32 seconds of burn-in for every 1/4 second of simulated time), and ignoring burn-in, the ratio drops to 29/128. While not substantial, this still demonstrates that our method provides nontrivial improvement in terms of the number of micro-scale steps needed to accurately take a macro-scale time step.

We finally note that the periodic boundary conditions make it more difficult to directly apply the JKOnet* approach to learn a potential. Thus, we do not provide a comparison to using this method as we do not believe it fairly reflects the setting where JKOnet* performs well. Because of this, we also omit the numeric plots of Wasserstein distance between control and approximated distributions and of the dimension-reduction of the trajectories (which are also more subtle in the periodic-domain setting), and we appeal to the visual similarity of the distributions after burn-in to assess the quality of our method's performance. We do point out, however, that another advantage of our algorithm is its ability to easily generalize to particles evolving in spaces which are not $\mathbb{R}^d$, as optimal transport maps between discrete distributions on manifolds are equally well defined. For another model for which we can fairly compare our algorithm to the equivalent JKOnet* approach and for which we present concrete numerical assessments, see the following section.

Remark 2.5.0.2. *As a final note, we point out that one can learn a PDE that models all three of our simulations (see [LPSK23] for a discussion of PDEs and the chemotactic model from Section 2.4, and note that the Langevin dynamics model in Section 2.6 can be understood using the Fokker-Planck equation). This indicates that our method works for particle models of PDEs, though the practical computational improvement might not be substantial because standard numerical methods for (known) PDEs are relatively cheap. In contrast, our method is better suited for the setting where the PDE is not known, and instead one only has a particle model.*

2.6 Simulation Study III: Langevin Dynamics

One of the key and motivating advantages of framing our algorithm in terms of optimal transport is that it easily generalizes to multi-dimensional models. We perform one final experiment as a proof of concept demonstrating that our algorithm works with a 2D model as well. For simplicity, the micro-scale model we use is an Euler–Maruyama simulation of an SDE: at each micro-scale time step of size h, each particle's position $x_i(t_j)$ is updated according to

$$x_i(t_{j+1}) = x_i(t_j) - \nabla U(x_i(t_j)) \cdot h + \sqrt{2h} \cdot Z_i(t_j)$$

where $Z_i(t_j) \sim \mathcal{N}(0, I_2)$. This is the standard Euler–Maruyama simulation of the SDE

$$dX_t = -\nabla U(X_t)dt + \sqrt{2}dW_t$$

with potential $U(x)$. For our micro-scale model, we use a half-moon potential given by

$$U(x, y) = A\,(r(x, y) - R)^2 + B \exp\left(-\alpha(y - y_s)\right)$$

Table 2.7: Parameters for 2-D half-moon micro-scale simulation.

Parameter	Description
$N = 10{,}000$	number of independent particles
$h = 1/2048$	micro-scale simulation step size
$x_i(0) \sim \text{Unif}([-4,4] \times [-4,4])$ i.i.d. $\quad \forall i = 1, \ldots, N$	initial particle locations
$T = 2$	end time

with $A = 2, B = 0.5, R = 2, \alpha = 1.5, y_s = -0.5$, and where the function $r(x, y) = \sqrt{x^2 + y^2}$ measures the distance from the origin.

Remark 2.6.0.1. *We reiterate that this micro-scale model is computationally cheap over long time scales, so in practice, there is little computational benefit to using our algorithm in this case. We present it as a demonstration of the algorithms performance on a 2D model, and in particular one whose steady-state distribution is not as simple as a standard Gaussian or uniform distribution.*

Table 2.8: Additional parameters for approximation algorithm with chemotaxis micro-scale simulation.

Parameter	Description
$S = 1/8$	length of start-up time
$k = 32$	number of centered differences to average
$H = 7/32$	length of each Euler-type prediction step
$H_R = 1/64$	burn-in time after each Euler step
$N_T = 7$	number of Euler steps
$R = 1/8$	length of final recovery time after last Euler step

As before, we run the micro-scale simulation from $t = 0$ to $t = T$ with the parameters shown in Table 2.7 (the end time is once again chosen long enough to effectively reach steady state). Figure 2.11 shows scatter plots of the particles over time for this "control" simulation.

We then run the approximation algorithm with the same parameters as in Table 2.7 and the additional parameters (as defined in Section 2.3.3) in Table 2.8. Note that

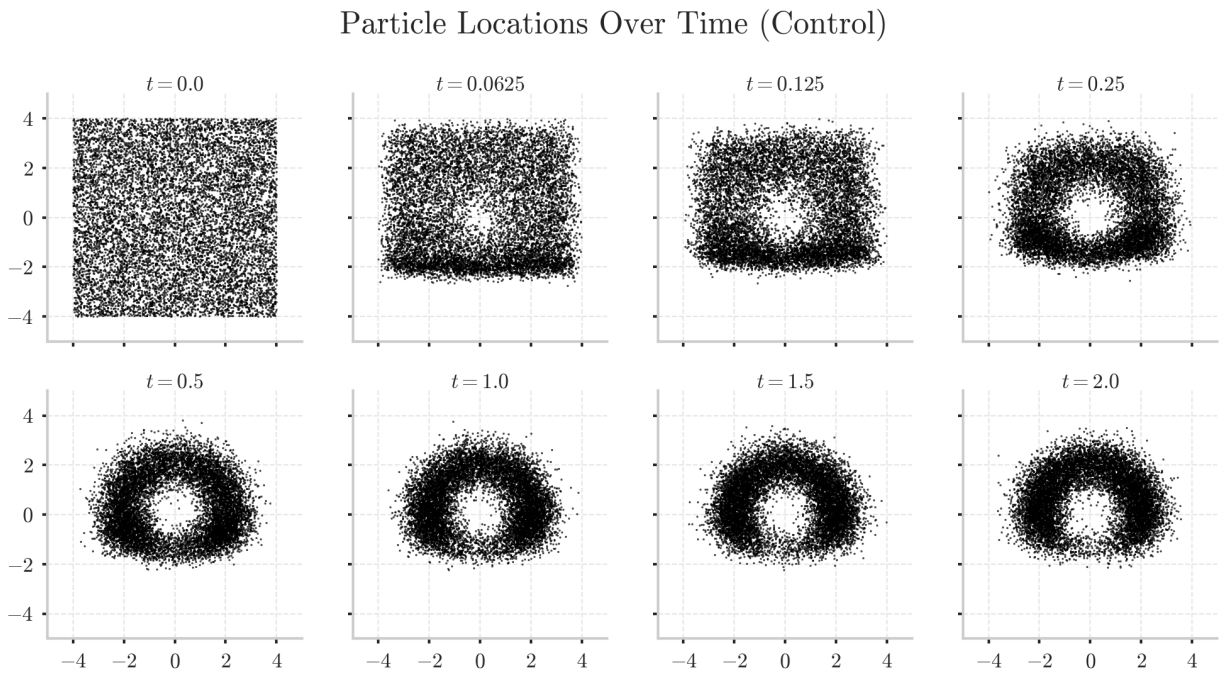


Figure 2.11: Scatter plots of particle locations at various times t from $t = 0$ to $t = 2$ for the control simulation of the half-moon Langevin dynamics model.

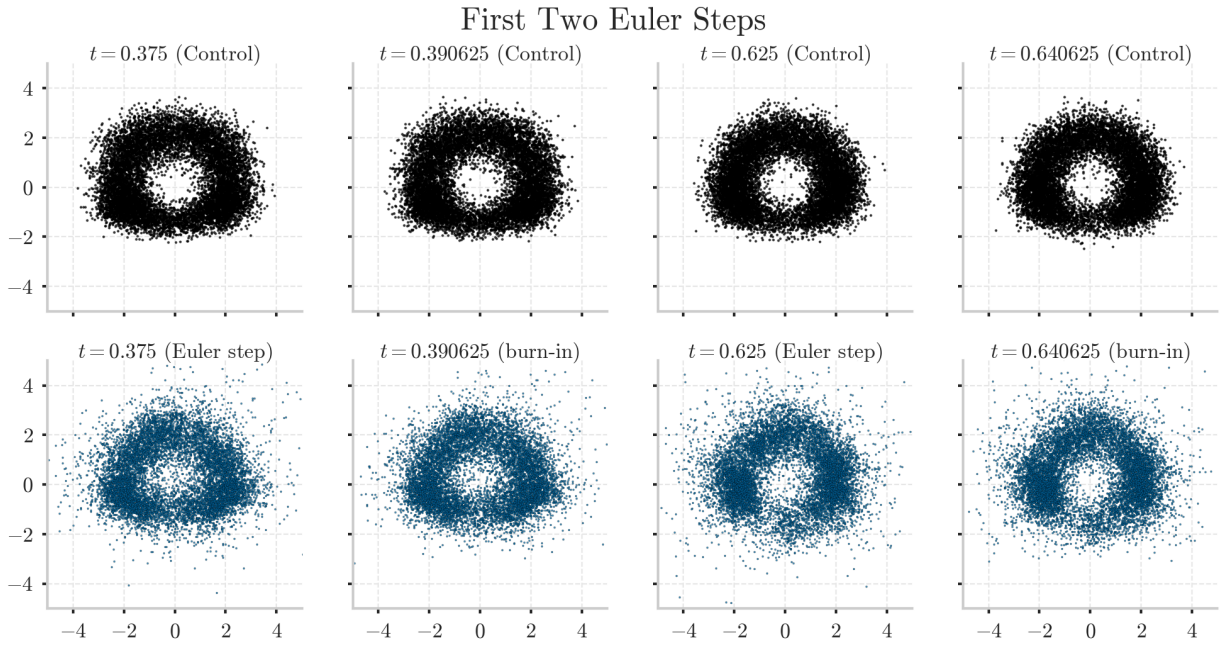
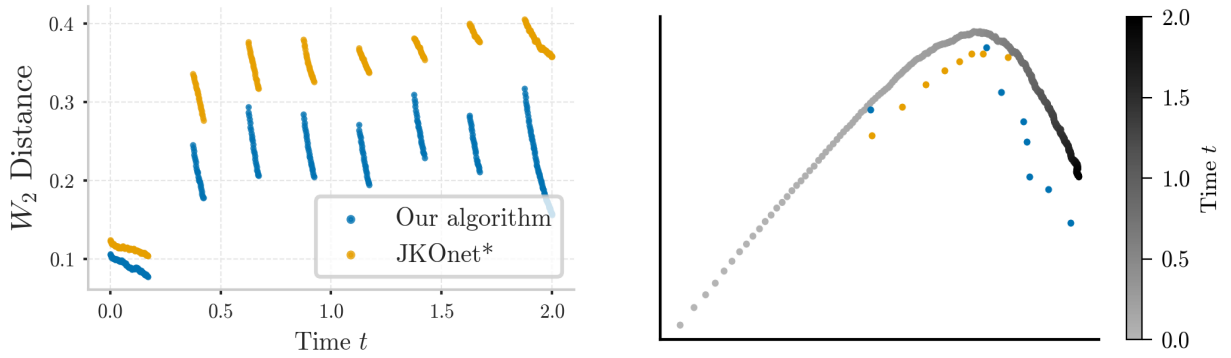


Figure 2.12: Scatter plots of particle locations after first two Euler steps (bottom) compared with corresponding times from control simulation (top). We run the micro-scale simulation until $t = 9/64$ (not pictured), take an Euler step to $t = 3/8$ (leftmost), burn in until $t = 25/64$ (second from left), take an Euler step to $t = 5/8$ (second from right), and burn in until $t = 41/64$ (rightmost).



(a) 2-Wasserstein distance between control distribution and approximated distributions versus time. Note that we only plot points during the burn-in and velocity-field estimation periods; we skip the intervals that correspond to each Euler-step.

(b) Projections of distribution trajectories onto the two principal components of the control distribution's evolution over time. The solid rainbow curve shows the control distribution, where the color represents time according to the color bar at the right. Each colored dot represents one Euler step for the corresponding approximation method.

Figure 2.13: Comparisons between control distributions and approximations. The blue is our algorithm, and the orange is using the JKOnet* method for learning and simulating an SDE.

the effective end time of this simulation is $T = S + N_T(H_S + H + H_R) + R = 2$ (since $H_S + H + H_R = 1/4$), which agrees with the control simulation. Figure 2.12 shows the distributions obtained after the first two Euler steps. We notice that, despite starting with the same uniform distribution as in the control, the approximated distribution after just the first Euler step already well-approximates the steady state distribution. This is consistent with the control distribution at $t = 3/8$ already being near steady state. In particular, Figure 2.12 demonstrates that just one Euler step from the uniform distribution is enough to approximately find the steady state, and the second step shows that our algorithm is stable when the distribution is near that steady state. Aside from a few particles straying far from the bulk distribution, we see that the approximated distributions closely match the shape and density of the control distributions, and the approximations only require 3/16 as many micro-scale steps as the control simulation (1/32 seconds of velocity-field estimation and 1/64 seconds of burn-in for every 1/4 of simulated time).

We also perform the equivalent algorithm using JKOnet* [TLGD24] to use the $2k+1 = 65$ micro-scale steps to learn a potential and then simulate an SDE with appropriate potential for the equivalent duration of the Euler step. We compare the W_2 distance between the control distribution and approximated distributions for both approximation methods in Figure 2.13a, and in Figure 2.13b, we perform PCA (see Section 2.4.2) to visualize how well the approximated distributions track the trajectory of the control distribution's evolution. In both figures, we see that our method outperforms the analogous JKOnet* approach, which suggests that our approach is better able to learn the local evolution. In particular, it appears that the JKOnet* approximated distributions do not update nearly enough over each step, which indicates that the learned potential is too mild. This is consistent with the understanding that the "local" data over short micro-scale bursts is not sufficient to learn a global potential, and it suggests that our approach of approximating a velocity field with optimal transport more efficiently learns the local evolution in order to predict the distribution some time in the future.

Acknowledgements

Chapter 2, in full, is a reprint of material in [KNK+25]: Nicholas Karris, Evangelos A. Nikitopoulos, Ioannis G. Kevrekidis, Seungjoon Lee, and Alexander Cloninger. Using linearized optimal transport to predict the evolution of stochastic particle systems. Preprint, arXiv:2408.01857, 2025. The dissertation author is the primary author and investigator of this paper, which is under revision for publication in the SIAM Journal on Applied Dynamical Systems.

Chapter 3

Choosing an Appropriate Time Step

3.1 Introduction

In Chapter 2, we attempt to approximate the velocity flow vector field v_t with a discrete finite-difference approximation

$$\mathbf{v}_t^{h,N}(x_i) := \frac{T_{\mu_t^N}^{\mu_{t+h}^N}(x_i) - x_i}{h}.$$

The hope is that, for large N, we have $\mathbf{v}_t \approx \mathbf{v}_t^{h,N}$ in the sense that

$$E_t^{h,N} := \|\mathbf{v}_t^{h,N} - \mathbf{v}_t\|_{L^2(\mu_t^N)}^2 \approx 0.$$

Unfortunately, for a fixed N, there is a tension in the choice of h: a small h results in large statistical error, and a large h results in a large model error. Explicitly, if there is an underlying true curve μ_t (i.e., μ_t^N is an empirical estimate of μ_t for all times t), then we can define the "true finite difference velocity field" to be

$$\mathbf{v}_t^h(x) := \frac{T_{\mu_t}^{\mu_{t+h}}(x) - x}{h}.$$

Note that with this definition, $\mathbf{v}_t := \lim_{h\to 0} \mathbf{v}_t^h$ in $L^2(\mu_t)$. The two sources of error in our velocity field approximation are then captured by the following two quantities:

$$M_t^{h,N} := \underbrace{\|\mathbf{v}_t^h - \mathbf{v}_t\|^2_{L^2(\mu_t^N)}}_{\text{“model error”}} \quad \text{and} \quad S_t^{h,N} := \underbrace{\|\mathbf{v}_t^{h,N} - \mathbf{v}_t^h\|^2_{L^2(\mu_t^N)}}_{\text{“statistical error”}}.$$

The model error $M_t^{h,N}$ captures how well the finite difference estimate approximates the true velocity field and is an empirical estimate of the deterministic quantity $M_t^h := \|\mathbf{v}_t^h - \mathbf{v}_t\|^2_{L^2(\mu_t)}$. Note that $M_t^h \to 0$ as $h \to 0$ by definition of $\mathbf{v}_t^h$, and so intuition suggests that the model error grows as h grows. On the other hand, the statistical error $S_t^{h,N}$ captures how well the discrete optimal transport map approximates the continuous one:

$$S_t^{h,N} := \|\mathbf{v}_t^{h,N} - \mathbf{v}_t^h\|^2_{L^2(\mu_t^N)} = \frac{1}{h^2}\|T^{\mu_{t+h}^N}_{\mu_t^N} - T^{\mu_{t+h}}_{\mu_t}\|^2_{L^2(\mu_t^N)}.$$

The observation in Chapter 2 is that for extremely small h, the discrete optimal transport map is actually just the particle-wise assignment, which is in general not a good approximation of the true (continuous) optimal transport map between μ_t and μ_{t+h}. This, combined with a large factor of $\frac{1}{h^2}$, causes the statistical error to be extremely large for small h.

While the triangle inequality gives

$$\sqrt{E_t^{h,N}} \le \sqrt{M_t^h} + \sqrt{S_t^{h,N}},$$

empirical evidence suggests that the stronger relationship

$$E_t^{h,N} \approx M_t^{h,N} + S_t^{h,N}$$

holds. Indeed, the two vector fields $\mathbf{v}_t^h - \mathbf{v}_t$ and $\mathbf{v}_t^{h,N} - \mathbf{v}_t^h$ are roughly “independent” as random variables and hence have no correlation, and so are roughly “orthogonal” with respect to the $L^2(\mu_t^N)$ inner product. This relationship suggests not only that $E_t^{h,N}$ is small

whenever both $M_t^{h,N}$ and $S_t^{h,N}$ are small, as the triangle inequality guarantees, it also suggests that $E_t^{h,N}$ is large whenever *either* $M_t^{h,N}$ or $S_t^{h,N}$ is large. This is the fundamental tension in the choice of h — choosing an h too small will result in a large statistical error $S_t^{h,N}$, and choosing an h too large will result in a large model error $M_t^{h,N}$. Both of these makes the total error $E_t^{h,N}$ large, and so the goal becomes to find an h that appropriately balances these two extremes. In this chapter, we make progress towards better explaining this tension.

To motivate much of our discussion, we will use the concrete example of Gaussian diffusion $\mu_t = \mathcal{N}(0, t)$ in one dimension. Optimal transport in one dimension is particularly nice in that optimal maps can be written in closed-form by composing CDFs, which means $\mathbf{v}_t$ and $\mathbf{v}_t^h$ can be computed explicitly. This allows us to explicitly compute not only the total error, but also the model and statistical errors separately. While this example is overly simplistic and is not *necessarily* representative of general phenomena, we hope that having a concrete example with explicit computations will help provide a useful benchmark for developing appropriate intuition for more general settings.

3.2 Orthogonality of the Model and Statistical Error

To appropriately motivate an in-depth discussion of the model and statistical errors, we first need to understand why these are indeed the two relevant quantities to care about. To this end, we define the vector fields

$$\mathbf{e} := \mathbf{v}_t^{h,N} - \mathbf{v}_t, \qquad \mathbf{m} := \mathbf{v}_t^h - \mathbf{v}_t, \qquad \mathbf{s} := \mathbf{v}_t^{h,N} - \mathbf{v}_t^h$$

so that $\mathbf{e} = \mathbf{m} + \mathbf{s}$. Then the total error $E_t^{h,N}$ can be decomposed as

$$\begin{aligned} E_t^{h,N} = \|\mathbf{e}\|^2_{L_2(\mu_t^N)} &= \|\mathbf{m}\|^2_{L^2(\mu_t^N)} + 2\langle \mathbf{m}, \mathbf{s}\rangle_{L^2(\mu_t^N)} + \|\mathbf{s}\|^2_{L^2(\mu_t^N)} \\ &= M_t^{h,N} + 2\langle \mathbf{m}, \mathbf{s}\rangle_{L^2(\mu_t^N)} + S_t^{h,N}. \end{aligned}$$

While it is not true that the inner product $\langle \mathbf{m}, \mathbf{s}\rangle_{L^2(\mu_t^N)} = 0$ exactly, there are heuristics for why we may expect it to have small contribution in general. At a high level, $\mathbf{m}$ is deterministic and smooth while $\mathbf{s}$ captures how the sampling noise affects the discrete estimate of the optimal transport map, and so we would not expect $\mathbf{s}$ to have a preferred direction relative to $\mathbf{m}$. Thus, we can expect the inner product across many sample points to essentially cancel in the empirical average, making the overall error contribution small compared to $M_t^{h,N}$ and $S_t^{h,N}$.

More precisely, set $\mu_t^N = \sum_{i=1}^N \delta_{x_i}$ so that

$$\langle \mathbf{m}, \mathbf{s}\rangle_{L^2(\mu_t^N)} = \frac{1}{N}\sum_{i=1}^N \langle \mathbf{m}(x_i), r(x_i)\rangle.$$

Note that $\mathbf{m}$ is deterministic and

$$\mathbf{s}(x_i) = \mathbf{v}_t^{h,N}(x_i) - \mathbf{v}_t^h(x_i) = \frac{1}{h}\left[T_{\mu_t^N}^{\mu_{t+h}^N}(x_i) - T_{\mu_t}^{\mu_{t+h}}(x_i)\right]$$

captures how well the discrete OT map approximates the continuous OT map at the sample points x_i. A useful heuristic is that $\mathbf{s}$ is approximately unbiased (i.e., the discrete optimal transport map is an approximately unbiased estimator of the true optimal transport map) so that

$$\mathbb{E}[\langle \mathbf{m}(x_i), \mathbf{s}(x_i)\rangle] = \mathbb{E}[\langle \mathbf{m}(x_i), \mathbb{E}[\mathbf{s}(x_i) \mid x_i]\rangle] \approx 0,$$

where the approximation reflects the heuristic that the empirical optimal transport map is approximately unbiased at fixed locations. Another useful heuristic is that the fluctuations

in $\langle \mathbf{m}(x_i), \mathbf{s}(x_i) \rangle$ are mostly due to sampling noise and so they are only weakly dependent as random variables. In particular, we do not expect there to be substantial accumulation in the empirical average in the same way we do for $M_t^{h,N}$ and $S_t^{h,N}$. Thus, for large N, the contribution from this cross term is typically small compared to the contribution from the two pure terms, which serves as loose justification for the approximation

$$E_t^{h,N} \approx M_t^{h,N} + S_t^{h,N}.$$

We emphasize that much of this discussion is heuristic and that many of the approximations indeed do not hold precisely. For example, it is not true that the random variables $\langle \mathbf{m}(x_i), \mathbf{s}(x_i) \rangle$ are exactly mean-zero, nor are they exactly independent. One can make more precise mathematical statements about statistical rates of empirical estimates of optimal transport maps, and while the details are outside the scope of this chapter, it is true that the given approximation holds well enough to be useful in many settings [MBNWW23, MBNWW24, MNW24].

To empirically verify this claim, consider the standard example for this chapter where $\mu_t = \mathcal{N}(0, t)$. Recall that for two absolutely continuous distributions $\mu, \nu \in \mathcal{P}_2(\mathbb{R})$, optimal tranpsort maps in one dimension can be computed as a composition of (inverse) CDFs: if F_μ, F_ν are the CDFs of μ, ν and are invertible, then the optimal transport map between μ and ν is given by

$$T_\mu^\nu = F_\nu^{-1}(F_\mu(x)).$$

Using this, we can compute

$$T_{\mu_t}^{\mu_{t+h}}(x) = \sqrt{\frac{t+h}{t}}x$$

and so

$$\mathbf{v}_t^h(x) = \frac{\sqrt{\frac{t+h}{t}} - 1}{h}x.$$

Taking the limit as $h \to 0$ gives that the true "underlying" velocity field is

$$\mathbf{v}_t(x) = \frac{x}{2t}.$$

As we will often do in this chapter, for the sake of the following experiment, we focus on $t = 1$, at which time we have

$$\mathbf{v}_1^h(x) = \frac{\sqrt{1+h} - 1}{h} x$$

and

$$\mathbf{v}_1(x) = \frac{x}{2}.$$

To assess the orthogonality of the vector fields $\mathbf{m}$ and $\mathbf{s}$ at $t = 1$ for different values of h and N, we run the following experiment. We first randomly sample N particles independently from $\mathcal{N}(0, 1)$ and call the resulting distribution μ_1^N. Then we add independently sampled steps from $\mathcal{N}(0, h)$ to each particle (in accordance with the Euler-Maruyama timestepper for the SDE $dX_t = dW_t$), and call the resulting distribution μ_{1+h}^N. Finally we sort both the initial and final distributions to effectively compute the optimal transport map between them. From the optimal transport map, we compute the vector field $\mathbf{v}_1^{h,N}$ in the usual way, and since we have closed-form expressions for $\mathbf{v}_1$ and $\mathbf{v}_1^h$, we can compute the total error $E_1^{h,N}$, the model error $M_1^{h,N}$, and the statistical error $S_1^{h,N}$ for each experiment. We perform this experiment for different values of h and N, and we plot the average values (across 32 independent trials per pair) for $E_1^{h,N}$ and $M_1^{h,N} + S_1^{h,N}$ in Figures 3.1 and 3.2. We see that the average values for $E_1^{h,N}$ and $M_1^{h,N} + S_1^{h,N}$ are nearly identical across many trials, which serves as empirical support for the validity of the approximation $E_t^{h,N} \approx M_t^{h,N} + S_t^{h,N}$ given above. We also see that for small values of h, the total error is closely approximated by the statistical error $S_1^{h,N}$, and for large values of h, the total error is closely approximated by the model error $M_1^{h,N}$, as we expect. The "optimal" value of h then comes when these

two quantities are roughly equal. These empirical observations indeed suggest that in order to understand the total error, it suffices to understand the behavior of the model and statistical errors seperately.

Finally, for completeness, we also show in Figure 3.3 the average value of the inner product

$$\left|\langle \mathbf{m}, \mathbf{s} \rangle_{L^2(\mu_1^N)}\right| = \frac{1}{2}\left|E_1^{h,N} - (M_1^{h,N} + S_1^{h,N})\right|$$

for various values of h, N averaged over 32 independent trials. This plot shows that, while it is not true that $\mathbf{m}$ and $\mathbf{s}$ are exactly orthogonal for any individual experiment, it is true that the size of their inner product is small compared to the values of $\|\mathbf{m}\|^2$ and $\|\mathbf{s}\|^2$, as the heuristics above suggest. Again, we admit that this particular example is not guaranteed to be representative of more intricate settings, but it does demonstrate that we are justified in focusing exclusively on the model and statistical errors for the rest of this chapter.

3.3 Model Error

We first focus on the model error $M_t^{h,N}$. To understand how $M_t^{h,N}$ can grow with h, we again consider the standard example of Gaussian diffusion $\mu_t = \mathcal{N}(0, t)$. As before, we compute the finite difference and true velocity fields

$$\mathbf{v}_t^h(x) = \frac{\sqrt{\frac{t+h}{t}} - 1}{h} x, \qquad \mathbf{v}_t(x) = \frac{x}{2t}.$$

Since $\mathbf{m} = \mathbf{v}_t^h - \mathbf{v}_t$ is deterministic, if $\mu_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{x_i}$ is an independent random sample from μ_t, then we know that

$$M_t^{h,N} := \|\mathbf{m}\|^2_{L^2(\mu_t^N)} = \frac{1}{N}\sum_{i=1}^N \|\mathbf{m}(x_i)\|^2$$

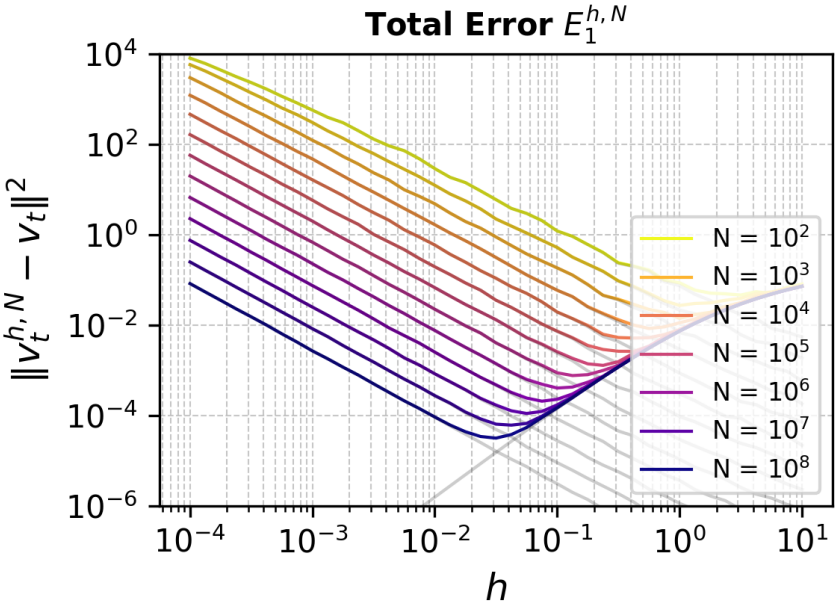


Figure 3.1: Average total error $E_1^{h,N}$ for different values of h, N, averaged over 32 independent simulations. Actual values of the model error $M_1^{h,N}$ and the statistical error $S_1^{h,N}$ are shown in gray, for comparison.

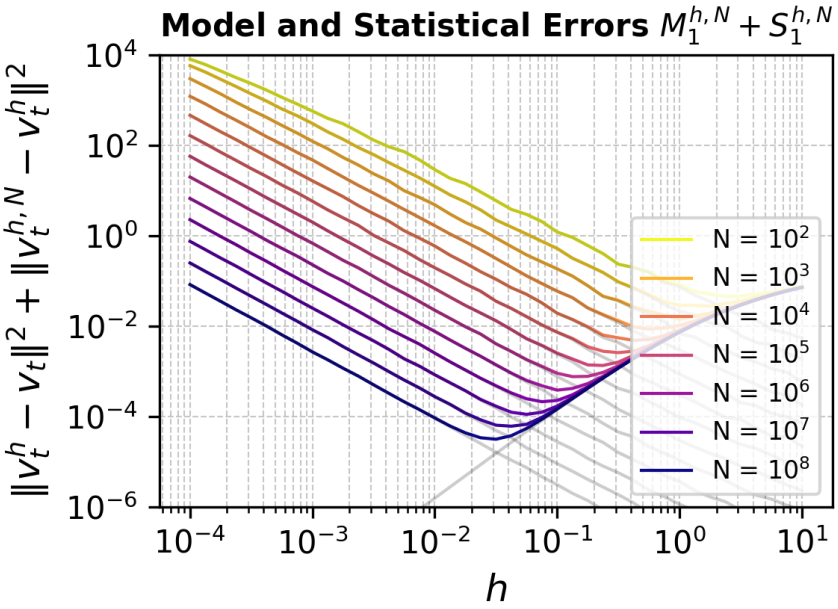


Figure 3.2: Average sum of model and statistical error $M_1^{h,N} + S_1^{h,N}$ for different values of h, N, averaged over 32 independent simulations. Actual values of the model error $M_1^{h,N}$ and the statistical error $S_1^{h,N}$ are shown in gray, for comparison.

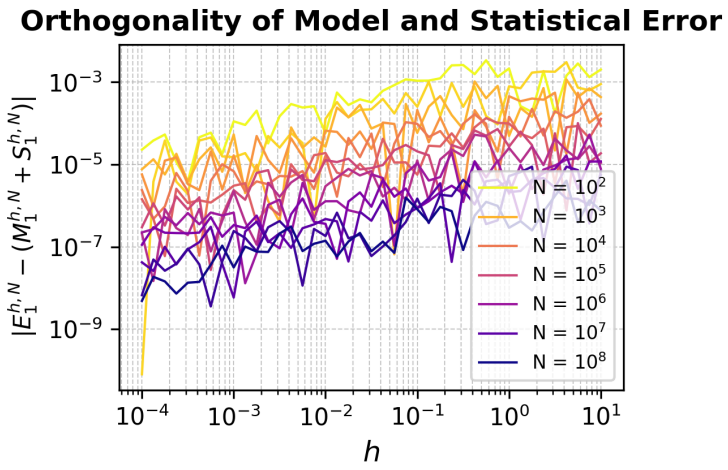


Figure 3.3: Average absolute value of the inner product $|\langle \mathbf{m}, \mathbf{s} \rangle|_{L^2(\mu_1^N)}$ for different values of h, N, averaged over 32 independent simulations. These values are consistently substantially smaller than the values of the model and statistical errors for corresponding h, N as shown in Figure 3.2. This indicates that indeed the inner product gives small contribution to the overall error, as suggested above.

is an unbiased estimator of $\|\mathbf{m}\|^2_{L^2(\mu_t)}$, and hence for large N we have

$$\begin{aligned} M_t^{h,N} = \|\mathbf{m}\|_{L^2(\mu_t^N)} &\approx \|\mathbf{m}\|_{L^2(\mu_t)} \\ &= \|\mathbf{v}_t - \mathbf{v}_t^h\|^2_{L^2(\mu_t)} \\ &= \int_{\mathbb{R}} \left| \frac{x}{2t} - \frac{\sqrt{\frac{t+h}{t}} - 1}{h} x \right|^2 d\mu_t(x) \\ &= \left| \frac{1}{2t} - \frac{\sqrt{\frac{t+h}{t}} - 1}{h} \right|^2 \int_{\mathbb{R}} |x|^2 d\mu_t(x) \\ &= \left| \frac{1}{2t} - \frac{\sqrt{\frac{t+h}{t}} - 1}{h} \right|^2 t. \end{aligned}$$

Plugging in $t = 1$ (as we will for our numerical experiments) gives

$$M_1^{h,N} = \left| \frac{1}{2} - \frac{\sqrt{1+h} - 1}{h} \right|^2.$$

Taylor expanding gives

$$\sqrt{1+h} \approx 1 + \frac{h}{2} - \frac{h^2}{8} + o(h^3)$$

so

$$M_t^{h,N} \approx \left(\frac{h}{8} + o(h^2)\right)^2 \approx \frac{h^2}{64}.$$

By taking logs, we have

$$\log_{10}(M_1^{h,N}) \approx 2\log_{10}(h) - \log_{10}(64) \approx 2\log_{10}(h) - 1.8$$

for small h. To empirically verify this relationship, we perform the same experiment described at the end of Section 3.2 and plot $M_1^{h,N}$ for various values of h, N in Figure 3.4. Since $\mathbf{m}$ is deterministic and does not depend on μ_{1+h}^N, this experiment is actually equivalent to empirically estimating $\|\mathbf{m}\|^2_{L^2(\mu_1)}$ using independent random samples from $\mu_1 = \mathcal{N}(0,1)$, so it is not surprising that the resulting plot agrees exactly with our calculations above. We show it for completeness nonetheless.

3.4 Statistical Error

Now we turn our attention to the more interesting statistical error $S_t^{h,N}$. We recall that we can compute $S_t^{h,N}$ as

$$S_t^{h,N} = \|\mathbf{s}\|^2_{L^2(\mu_t^N)} = \|\mathbf{v}_t^{h,N} - \mathbf{v}_t^h\|^2_{L^2(\mu_t^N)} = \frac{1}{h^2}\left\| T_{\mu_t^N}^{\mu_{t+h}^N} - T_{\mu_t}^{\mu_{t+h}} \right\|^2_{L^2(\mu_t^N)},$$

and so $S_t^{h,N}$ essentially measures how well the discrete optimal transport map approximates the true optimal transport map between the underlying continuous distributions. While there exists work analyzing the rates of convergence of discrete optimal transport maps, nearly all of it is concerned with the case of independent samples from the source and target distributions. In our case with a microscale timestepper, the two distributions μ_t^N

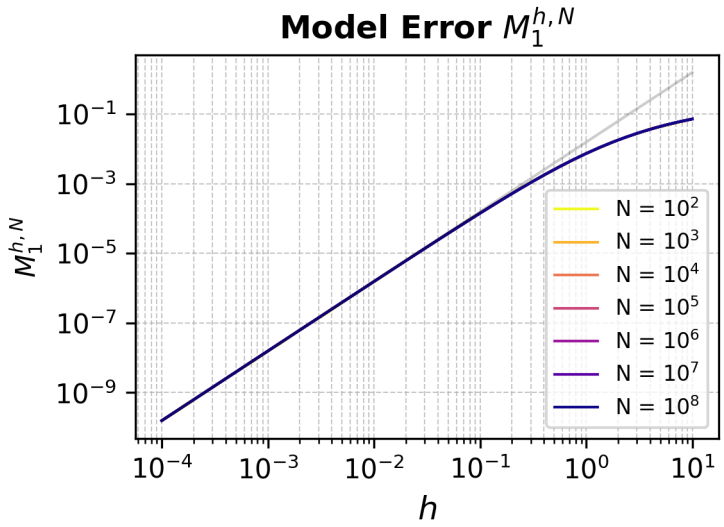


Figure 3.4: Average value of the model error $M_1^{h,N}$ for different values of h, N, averaged over 32 independent samples. Note that, for small values of h, this plot agrees exactly with the relationship $\log_{10}(M_1^{h,N}) \approx 2\log_{10}(h) - 1.8$ derived above (this theoretical line is plotted lightly in gray for comparison). In particular, the model error does not depend on N (all values of N give almost identical curves in the plot).

and μ_{t+h}^N are definitely *not* independent — μ_{t+h}^N is thought of as the result of using the timestepper to advance the particle system h seconds forward in time, so the location of the particles at time $t + h$ is therefore a (typically random) function of the location of the particles at time t. Therefore, none of the statistical rates for discrete optimal transport maps apply directly.

At first glance, our setting may sound worse because the two distributions are dependent, and so we may expect our empirical estimators to be biased. However, the situation is actually intuitively better than the case of independent samples from each distribution. With independent samples, there is no guarantee of locality — samples at time $t + h$ may not be anywhere near the samples at time t, and so the "curves" that the "particles" trace out need not even be continuous. This makes using a finite difference to estimate a velocity field fraught. In the case of using a timestepper to step each particle forward in time, at least the particle curves are "continuous" in the sense that as we take $h \to 0$, the distribution at time $t + h$ given by the timestepper gets close (with high

probability) to the distribution at time t. Thus, we expect that using finite differences to estimate velocity fields is in fact *better* in the case of a timestepper rather than taking independent samples. However, it is true that the behavior of the statistical error is easier to understand in the case of independent samples from each distribution, so we will first begin our analysis there.

3.4.1 Statistical Error with Independent Samples in 1D

Let μ, ν be continuous measures on $\mathbb{R}$. Let f, g be the respective PDFs and F, G the respective CDFs. If μ^N, ν^N are independent empirical estimates, then they can be written

$$\mu^N = \sum_{i=1}^{N} \delta_{F^{-1}(U_i)}, \quad \nu^N = \sum_{i=1}^{N} \delta_{G^{-1}(V_i)}$$

where U_i, V_i are independent uniform order statistics.

Recall that for one-dimensional distributions, the true optimal transport map has a closed-form expression

$$T_\mu^\nu(x) = G^{-1}(F(x)),$$

and the empirical optimal transport map

$$T_{\mu^N}^{\nu^N}(F^{-1}(U_i)) = G^{-1}(V_i)$$

is obtained by sorting the particles (i.e., the ith order statistic (the ith sorted particle) from μ^N gets mapped to the ith order statistic from ν^N). Thus, the error in the empirical optimal

transport map can be computed as

$$\begin{aligned}\|T_\mu^\nu - T_{\mu^N}^{\nu^N}\|_{L^2(\mu^N)}^2 &= \frac{1}{N}\sum_{i=1}^N (G^{-1}(F(F^{-1}(U_i))) - G^{-1}(V_i))^2 \\ &= \frac{1}{N}\sum_{i=1}^N (G^{-1}(U_i) - G^{-1}(V_i))^2.\end{aligned}$$

Note that this does not depend on μ at all.

We can attempt to approximate this quantity by Taylor expanding G^{-1} around the expected values for U_i, V_i. Recall that U_i, V_i are independent order statistics and so are independent samples from the Beta distribution $\text{Beta}(i, n+1-i)$. Thus we have

$$\mathbb{E}[U_i] = \frac{i}{N+1} =: p_i, \quad \text{Var}[U_i] = \frac{i}{N+1}\cdot\frac{N+1-i}{N+1}\cdot\frac{1}{N+2} =: \frac{p_i(1-p_i)}{N+2}.$$

Now write $U_i = p_i + \delta_i^U$ and $V_i = p_i + \delta_i^V$. Then Taylor expanding G^{-1} around p_i gives

$$\begin{aligned}G^{-1}(U_i) - G^{-1}(V_i) &\approx (G^{-1})'(p_i)(U_i - V_i) \\ &= (G^{-1})'(p_i)(\delta_i^U - \delta_i^V) \\ &= \frac{1}{G'(G^{-1}(p_i))}\cdot(\delta_i^U - \delta_i^V) \\ &= \frac{1}{g(G^{-1}(p_i))}(\delta_i^U - \delta_i^V)\end{aligned}$$

so the error estimate is

$$\|T_\mu^\nu - T_{\mu^N}^{\nu^N}\|_{L^2(\mu^N)}^2 = \frac{1}{N}\sum_{i=1}^N (G^{-1}(U_i) - G^{-1}(V_i))^2 \approx \frac{1}{N}\sum_{i=1}^N \frac{(\delta_i^U - \delta_i^V)^2}{g(G^{-1}(p_i))^2}.$$

The only random variables in the RHS are δ_i^U, δ_i^V which are independent with mean zero and variance $\frac{p_i(1-p_i)}{N+2}$, and so

$$\mathbb{E}[(\delta_i^U - \delta_i^V)^2] = \text{Var}[U_i] + \text{Var}[V_i] = \frac{2p_i(1-p_i)}{N+2}$$

and hence

$$\mathbb{E}\left[\|T_\mu^\nu - T_{\mu^N}^{\nu^N}\|_{L^2(\mu^N)}^2\right] \approx \frac{2}{N(N+2)} \sum_{i=1}^{N} \frac{p_i(1-p_i)}{g(G^{-1}(p_i))^2}.$$

Note that for large N, the sum is approximating the integral

$$J_2(\nu) := \int_0^1 \frac{p(1-p)}{g(G^{-1}(p))^2} dp = \int_{-\infty}^{\infty} \frac{G(y)(1-G(y))}{g(y)} dy$$

(with J_2 defined by (5.1) in [BL19]) and so we have

$$\mathbb{E}\left[\|T_\mu^\nu - T_{\mu^N}^{\nu^N}\|_{L^2(\mu^N)}^2\right] \approx \frac{2}{N} J_2(\nu).$$

We note that this is consistent with standard results for statistical rates of discrete optimal transport maps, which typically give rates of convergence of $N^{-1/d}$ (recall $d = 1$ for this analysis).

We now return to the standard example $\mu_t = \mathcal{N}(0, t)$ to empirically verify these estimates. We perform a similar experiment as above in which we independently sample from $\mathcal{N}(0, 1)$ and $\mathcal{N}(0, 1+h)$, compute the optimal transport map by sorting, and compute $S_1^{h,N}$ in the usual way. We plot the average value of

$$h^2 S_1^{h,N} = \left\| T_{\mu_t^N}^{\mu_{t+h}^N} - T_{\mu_t}^{\mu_{t+h}} \right\|_{L^2(\mu_t^N)}^2$$

(across 32 independent simulations) for various values of h, N in Figure 3.5.

Now note that for $\nu = \mathcal{N}(0, 1+h)$, we have

$$J_2(\nu) = (1+h) J_2(\mathcal{N}(0,1)) = (1+h) \cdot \pi$$

and so

$$\mathbb{E}\left[\|T_\mu^\nu - T_{\mu^N}^{\nu^N}\|_{L^2(\mu^N)}^2\right] \approx \frac{2\pi(1+h)}{N} \tag{3.4.1.1}$$

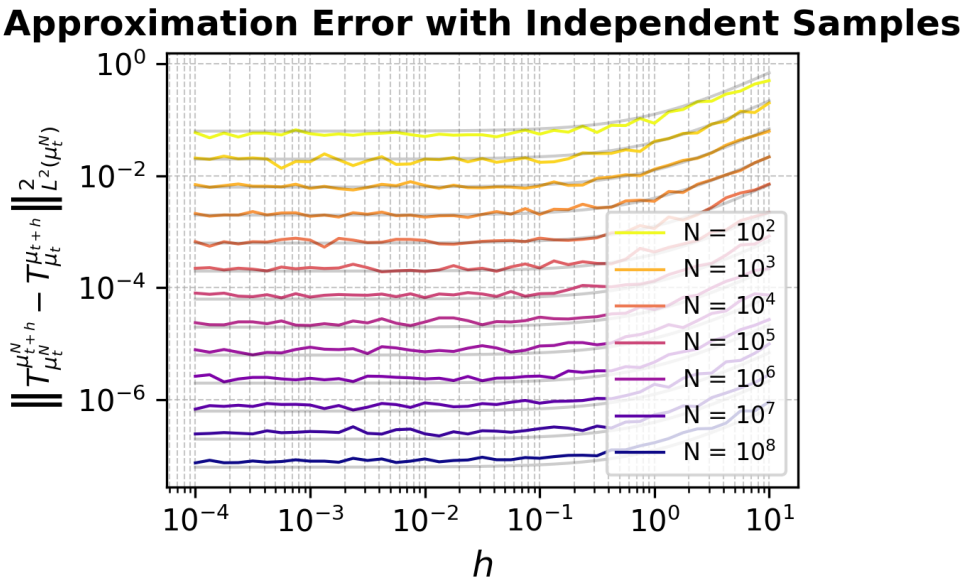


Figure 3.5: Average value of the optimal transport estimation error $h^2 S_1^{h,N}$ for different values of h, N, averaged over 32 independent trials per pair, in the case of independent samples from $\mu_1 = \mathcal{N}(0,1)$ and $\mu_{1+h} = \mathcal{N}(0, 1+h)$. The theoretical values suggested by (3.4.1.1) are overlayed in gray for comparison. We see that the empirical estimates closely match the analytically predicted values, as expected.

which closely matches the experimental data, as shown in Figure 3.5.

We finally plot the actual statistical error $S_1^{h,N}$ in the regime of independent samples in Figure 3.6. Because of the flatness of the optimal transport approximation error curves seen in Figure 3.5, the large statistical error for small h is primarily caused by the large factor of $\frac{1}{h^2}$. This hints at the true cause of the large statistical error in the setting with a microscale simulator.

3.4.2 Statistical Error with Particle Timesteppers

While the analysis above is quite nice and gives a satisfying explanation for why small h leads to a large statistical error in the case of independent samples, it is difficult to extend the analysis to the case where the particles are pushed forward by a timestepper. To see why, suppose μ is a continuous measure with CDF F and U_i are independent order

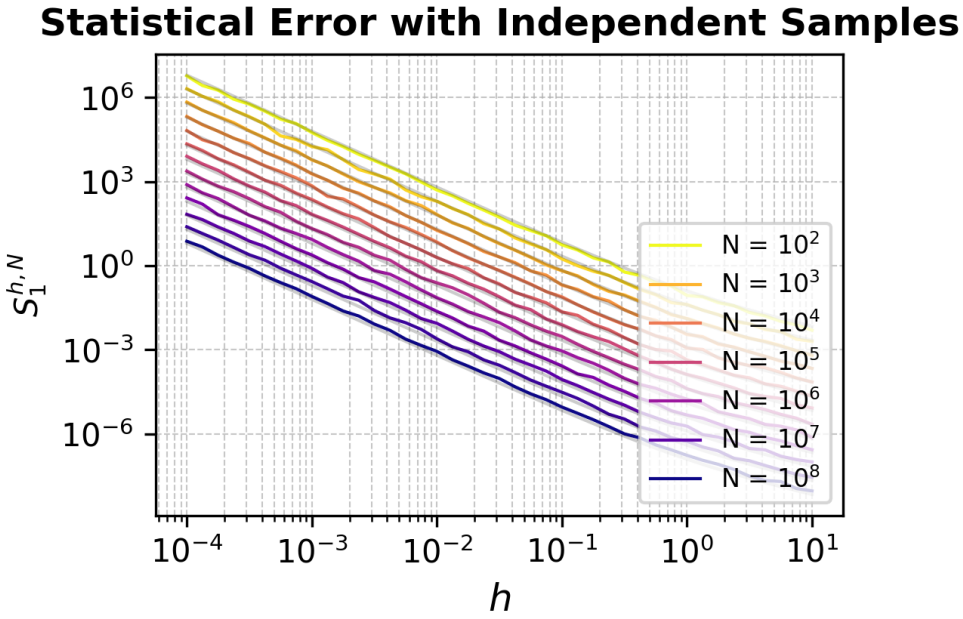


Figure 3.6: Average value of the statistical error $S_1^{h,N}$ for different values of h, N, averaged over 32 independent trials per pair, in the case of independent samples from $\mu_1 = \mathcal{N}(0,1)$ and $\mu_{1+h} = \mathcal{N}(0, 1+h)$. The theoretical values suggested by (3.4.1.1) (appropriately scaled by $\frac{1}{h^2}$) are overlayed in gray for comparison. We see that the factor of $\frac{1}{h^2}$ causes a large statistical error for small h, as expected.

statistics so that

$$\mu^N := \sum_{i=1}^{N} \delta_{F^{-1}(U_i)}$$

is an empirical estimate of μ. Now suppose the timestepper moves particle i by ε_i and that

$$\nu^N := \sum_{i=1}^{N} \delta_{F^{-1}(U_i)+\varepsilon_i}$$

is an empirical estimate of some continuous measure ν with CDF G. Then if we set $V_i = G(\{F^{-1}(U_j) + \varepsilon_j\}_{(i)})$ (i.e., V_i is the percentile of the ith sorted particle in ν^N), it is not true that V_i is an independent order statistic distributed according to a beta distribution. In fact, it is not at all clear how V_i depends on U_i in general, as V_i generically does not corresponding to the same particle as U_i — that's the whole point of resorting the resulting distribution. We can still perform the same analysis as in Section 3.4.1 to get

$$\|T_\mu^\nu - T_{\mu^N}^{\nu^N}\|_{L^2(\mu^N)}^2 \approx \frac{1}{N} \sum_{i=1}^{N} \frac{(\delta_i^U - \delta_i^V)^2}{g(G^{-1}(p_i))^2},$$

but in this case, δ_i^U and δ_i^V are not independent, nor frankly is δ_i^V a random variable that we know much about. Thus, the best we can say is that

$$\mathbb{E}[(\delta_i^U - \delta_i^V)^2] = \mathbb{E}\left[(\delta_i^U - [G(\{F^{-1}(U_j) + \varepsilon_j\}_{(i)}) - p_i])^2\right],$$

which is generally unhelpful. The sorting really does wreak havoc on our ability to calculate this explicitly.

However, we can still perform experiments to get a sense for how the statistical error behaves in the case of particle timesteppers. We perform the same experiment as in the case of independent samples, except now to generate μ_{1+h}^N, we add noise sampled from $\mathcal{N}(0, h)$ to each particle in the sample μ_1^N. We recall that this is precisely the timestepper given by an Euler-Maruyama simulation of the Gaussian diffusion SDE ($dX_t = dW_t$) with

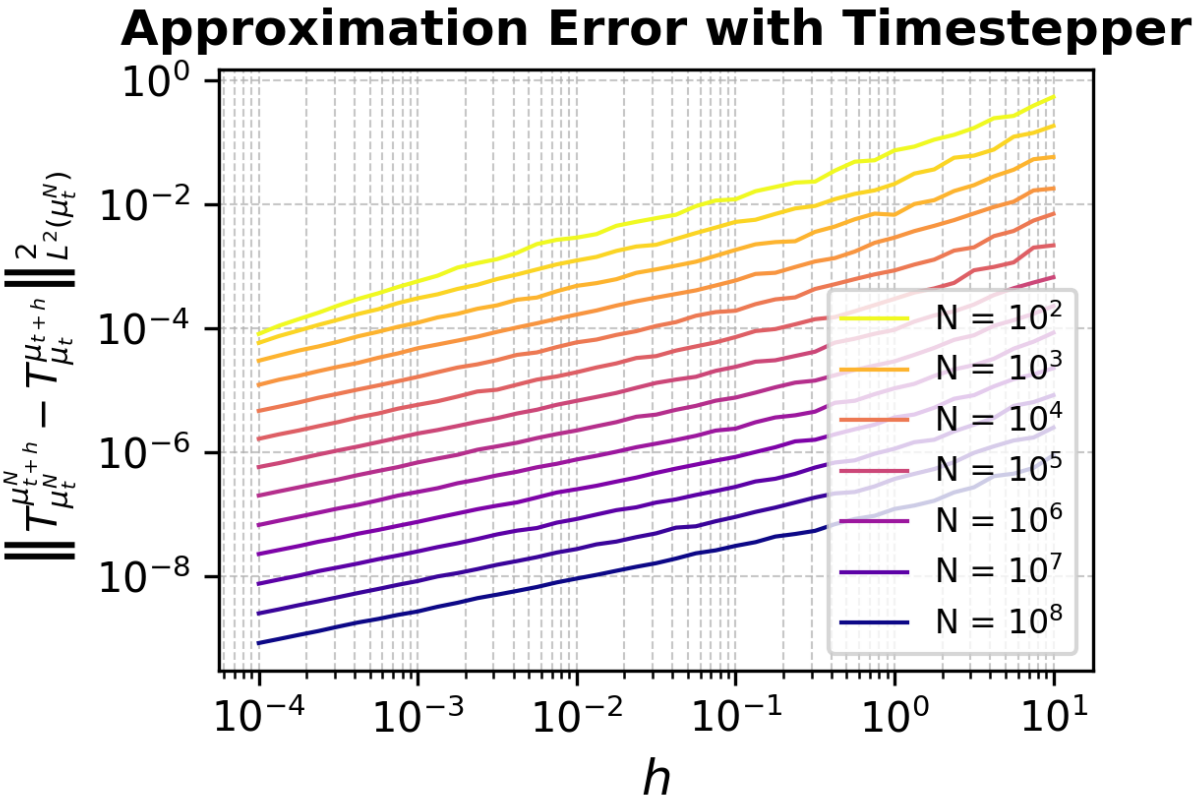


Figure 3.7: Average value of the optimal transport estimation error $h^2 S_1^{h,N}$ for different values of h, N, averaged over 32 independent trials per pair, in the case of an Euler-Maruyama particle timestepper.

initial condition $X_1 \sim \mathcal{N}(0,1)$, and that μ_{1+h}^N is an empirical estimate of the underlying distribution $\mu_t = \mathcal{N}(0, 1+h)$. We use this simulation to compute both the optimal transport approximation error $\|T_{\mu_1^N}^{\mu_{1+h}^N} - T_{\mu_1}^{\mu_{1+h}}\|^2_{L^2(\mu_1^N)}$ and the statistical erorr $S_1^{h,N}$ (averaged over 32 independent simulations) for different values of h, N, which we plot in Figures 3.8 and 3.7, respectively. We recall that

$$S_1^{h,N} = \frac{1}{h^2} \|T_{\mu_1^N}^{\mu_{1+h}^N} - T_{\mu_1}^{\mu_{1+h}}\|^2_{L^2(\mu_1^N)},$$

so these plots are showing the same phenomenon up to a factor of h^2. We show them both because they highlight different patterns.

We notice that the optimal transport approximation error (Figure 3.7) is generally

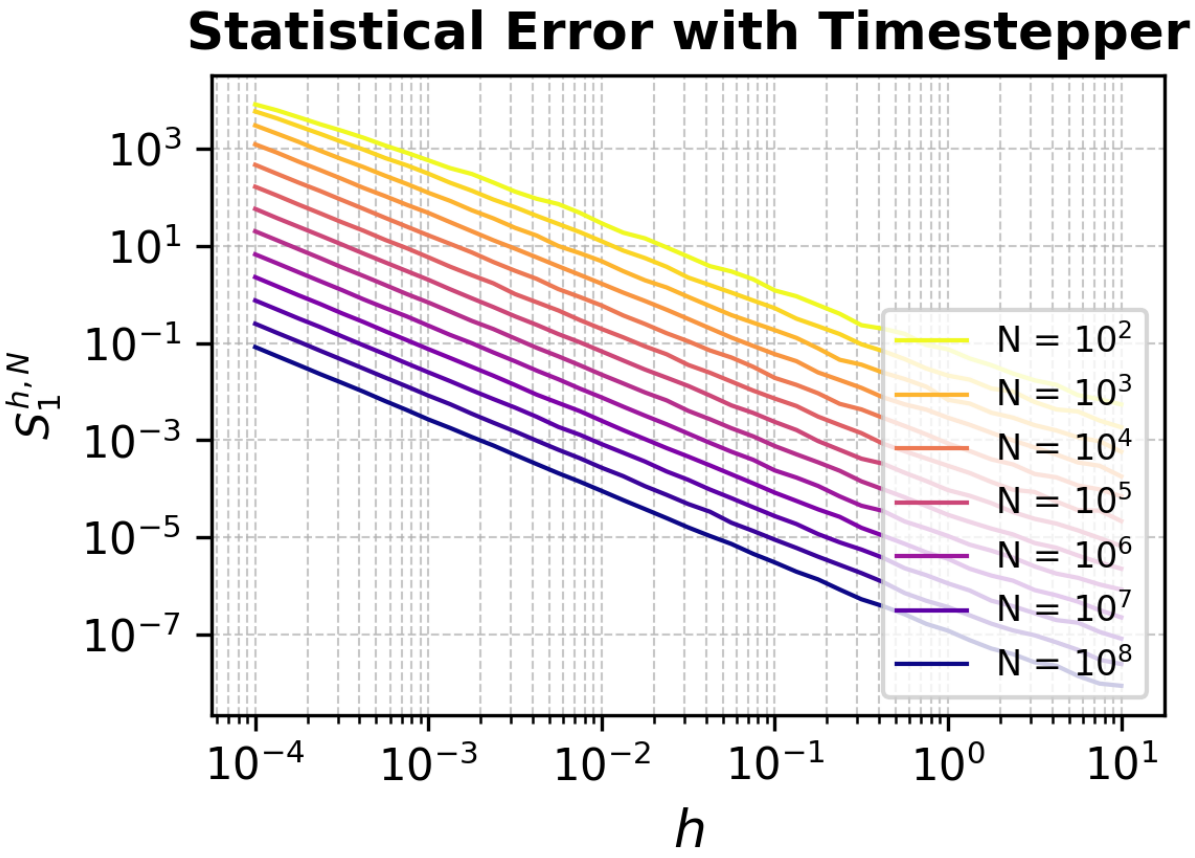


Figure 3.8: Average value of the statistical error $S_1^{h,N}$ for different values of h, N, averaged over 32 independent trials per pair, in the case of an Euler-Maruyama particle timestepper.

smaller than in the case of independent samples (compare with Figure 3.5), especially for small h. This confirms the intuition discussed at the beginning of this section that the case of independent samples is actually worse for approximating the correct optimal transport map. For large h, the approximation errors are closer to those in the case of independent samples, which is also intuitive: the larger the timestep h, the more "independent" the two distributions will seem. The plots also suggest that the true error seems to roughly follow a power law. Using linear regression gives an approximate power law of

$$S_1^{h,N} \approx 3.30 \cdot N^{-0.92} \cdot h^{-1.44}. \tag{3.4.2.1}$$

Understanding the theoretical underpinnings that give rise to these specific powers is an area of ongoing investigation. However, the dependence on the number of particles N does raise an important concern: if we were not sorting the particles (i.e., we were approximating the optimal transport map using the particle-wise assignment), then we would expect the approximation error to be an empirical estimate of some deterministic quantity, which should not depend on N. Indeed, if the particles are not being sorted, then the number of particles used in the simulation cannot affect the approximation error — the approximate "optimal transport map" at a particular location is independent of the other particles.

Explicitly, if we set $\mu_1^N = \delta_{x_i}$ and apply the timestepper, then $\mu_{1+h}^N = \delta_{x_i+\varepsilon_i}$, where $\varepsilon_i \sim \mathcal{N}(0, h)$ are independent. If we also then set

$$T_p(x_i) = x_i + \varepsilon_i$$

to be the particle-wise transport map, we can compute the approximation error as

$$\begin{aligned}\|T_p - T_{\mu_1}^{\mu_{1+h}}\|_{L^2(\mu_1^N)}^2 &= \frac{1}{N}\sum_{i=1}^{N}\left\|x_i + \varepsilon_i - \sqrt{1+h}x_i\right\|^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\left\|\left(1-\sqrt{1+h}\right)x_i + \varepsilon_i\right\|^2.\end{aligned}$$

Notice that $x_i \sim \mathcal{N}(0,1)$ and $\varepsilon_i \sim \mathcal{N}(0,h)$ are independent Gaussians, and so

$$\begin{aligned}\mathbb{E}\left[\left\|\left(1-\sqrt{1+h}\right)x_i + \varepsilon_i\right\|^2\right] &= (1-\sqrt{1+h})^2\mathrm{Var}[x_i] + \mathrm{Var}[\varepsilon_i] \\ &= (1-\sqrt{1+h})^2 + h \\ &= 2 + 2h - 2\sqrt{1+h}.\end{aligned}$$

Thus, the approximation error $\|T_p - T_{\mu_1}^{\mu_{1+h}}\|_{L^2(\mu_1^N)}^2$ is an empirical estimate of $2+2h-2\sqrt{1+h}$, which does not depend on N. This suggests that it is actually *not* a lack of sorting that is leading to the large statistical error for small h. It is still true that there are values of h small enough that would make the optimal map the particle-wise one, but it turns out that these values are even smaller than the ones shown in the plots. In fact, we can begin to see this behavior with the $N = 10^2$ curve beginning to approach the $N = 10^3$ for $h = 10^{-4}$. For even smaller values of h, these curves would all approach the theoretical curve of $2 + 2h - 2\sqrt{1+h}$ computed above.

We conclude this section with a few takeaways. For the Euler-Maruyama timestepper for Gaussian diffusion $\mu_t = \mathcal{N}(0,t)$, the true statistical error $S_1^{h,N}$ seems to follow a power law with powers that are not theoretically understood yet but are vaguely reminiscent of the case of independent samples. This relationship suggests that the large statistical error for small values of h is primarily due to the large factor of $\frac{1}{h^2}$, as opposed to the simpler description of the optimal transport map estimation being less accurate. In fact, the empirical optimal transport map actually is *more* accurate for small h, but its convergence

is too slow to overcome the scale factor of $\frac{1}{h^2}$. All of this points to a fruitful direction of further research attempting to give theoretical backing to these empirical observations.

3.5 Consecutive Finite Difference Error

We conclude this chapter by returning to the big-picture purpose of this discussion, which is to better understand how to choose an appropriate time step h in our Euler-type algorithm from Chapter 2. Essentially, we want a way to choose the value of h that minimizes the total error $E_t^{h,N}$ for the given values of t and N. We have seen that the total error is approximately the sum of the model and statistical errors, and so it suffices to attempt to approximate each separately.

Recall that the fundamental issue with attempting to compute $E_t^{h,N}$ directly is that it requires knowledge of the underlying "true" velocity field $\mathbf{v}_t$, which we assume is unknown to us. If $\mathbf{v}_t$ was known, then we could simply use it in a standard application of Euler's method, and there would be no need to use the timestepper and optimal transport maps to approximate it. Therefore, we need a way to estimate $E_t^{h,N}$ without needing to know $\mathbf{v}_t$. To this end, we return to a heuristic mentioned briefly in Chapter 2, which is to use the difference of consecutive velocity field estimates:

$$\hat{E}_t^{h,N} := \|\mathbf{v}_t^{h,N} - \mathbf{v}_t^{2h,N}\|^2_{L^2(\mu_t^N)}.$$

Empirically, this "consecutive difference" well-approximates the true error, as we see in Figure 3.9.

Indeed, this resemblance is not a coincidence. Similar to before, set

$$\mathbf{m}^h := \mathbf{v}_t^h - \mathbf{v}_t, \qquad \mathbf{m}^{2h} := \mathbf{v}_t^{2h} - \mathbf{v}_t, \qquad \mathbf{s}^h := \mathbf{v}_t^{h,N} - \mathbf{v}_t^h, \qquad \mathbf{s}^{2h} := \mathbf{v}_t^{2h,N} - \mathbf{v}_t^{2h}.$$

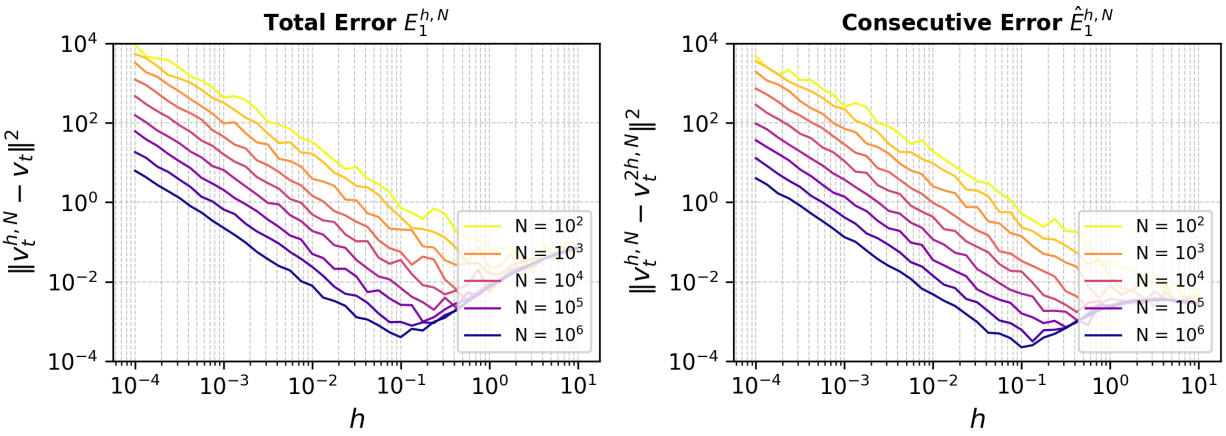


Figure 3.9: Average value of the consecutive error $\hat{E}_1^{h,N}$ for different values of h, N, averaged over 32 independent trials per pair, in the case of an Euler-Maruyama particle timestepper.

Then

$$\hat{E}_t^{h,N} = \|\mathbf{m}^h - \mathbf{m}^{2h}\|^2 + \|\mathbf{s}^h - \mathbf{s}^{2h}\|^2 + 2\langle \mathbf{m}^h - \mathbf{m}^{2h}, \mathbf{s}^h - \mathbf{s}^{2h} \rangle.$$

As before, a useful heuristic is that the cross term is negligible because $\mathbf{m}^h - \mathbf{m}^{2h}$ is deterministic and $\mathbf{s}^h - \mathbf{s}^{2h}$ is approximately unbiased, so

$$\hat{E}_t^{h,N} \approx \|\mathbf{m}^h - \mathbf{m}^{2h}\|^2 + \|\mathbf{s}^h - \mathbf{s}^{2h}\|^2 =: \hat{M}_t^{h,N} + \hat{S}_t^{h,N}.$$

Now consider our standard example of Gaussian diffusion. We can explicitly calculate

$\hat{M}_t^{h,N}$ at $t = 1$ as

$$\begin{aligned}
\hat{M}_1^{h,N} &= \|\mathbf{v}_1^h - \mathbf{v}_1^{2h}\|^2_{L^2(\mu_1^N)} \\
&\approx \|\mathbf{v}_1^h - \mathbf{v}_1^{2h}\|^2_{L^2(\mu_1)} \\
&= \int_{\mathbb{R}} \left| \frac{\sqrt{1+h}-1}{h} x - \frac{\sqrt{1+2h}-1}{2h} x \right|^2 d\mu_1(x) \\
&= \left| \frac{\sqrt{1+h}-1}{h} - \frac{\sqrt{1+2h}-1}{2h} \right|^2 \mathrm{Var}[\mu_1] \\
&= \left| \frac{\sqrt{1+h}-1}{h} - \frac{\sqrt{1+2h}-1}{2h} \right|^2 \\
&\approx \left(\frac{h}{8} + o(h^2) \right)^2
\end{aligned}$$

where the last line comes from a Taylor expansion of $\sqrt{1+h}$ and $\sqrt{1+2h}$. The upshot is that

$$\hat{M}_1^{h,N} \approx \frac{h^2}{64} = M_1^{h,N},$$

and so $\hat{M}_1^{h,N}$ is indeed a good estimate for the true model error.

The statistical error estimate $\hat{S}_1^{h,N}$ gets more tricky because $\mathbf{s}^h$ and $\mathbf{s}^{2h}$ are dependent: not only are both μ_{t+h}^N and μ_{t+2h}^N both dependent on μ_t^N, but μ_{t+2h}^N is also dependent on μ_{t+h}^N because it is the result of another timestepper step applied to μ_{t+h}^N. Hence, theoretical analysis of the statistical error estimate is similarly hard to empirical analysis of the true statistical error. However, in Figure 3.10, we empirically observe that

$$\hat{S}_t^{h,N} \approx \frac{1}{2} \left(S_t^{h,N} + S_t^{2h,N} \right).$$

This is an intuitively reasonable approximation, but giving more theoretical justification is an area of ongoing work. It does, however, allow us to make the (albeit empirical) claim

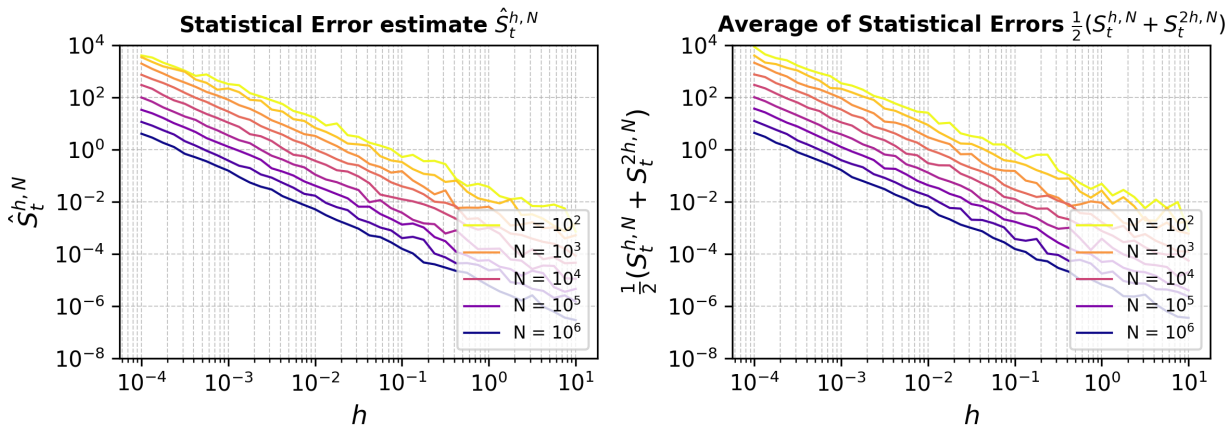


Figure 3.10: Average value of the consecutive statistical error estimate $\hat{S}_1^{h,N}$ compared with the average of two statistical errors $\frac{1}{2}(S_1^{h,N} + S_1^{2h,N})$ for different values of h, N, averaged over 32 independent trials per pair, in the case of an Euler-Maruyama particle timestepper.

that

$$\hat{E}_t^{h,N} = \hat{M}_t^{h,N} + \hat{S}_t^{h,N} \approx M_t^{h,N} + S_t^{h,N} = E_t^{h,N},$$

and so indeed the consecutive difference approximation error $\hat{E}_t^{h,N}$ is a reasonable surrogate for the true error $E_t^{h,N}$. We conclude by noting that choosing an h that minimizes $\hat{E}_t^{h,N}$ does not require any knowledge of the underlying dynamics of the system or timestepper, and so while it may require a parameter search, this analysis suggests that it may be possible to choose an (approximately) optimal h only using the timestepper. We again acknowledge that the example used in the empirical experiments is extremely simplistic, and some of the observed phenomena may not entirely generalize to more complex settings. However, we believe these observations are valuable in better understanding why the question of choosing an appropriate h is subtle and in illuminating the interaction of various effects that contribute to the different sources of error in the vector field estimation.

Chapter 4

Newton's Method for Approximating Steady States

Timesteppers constitute a powerful tool in modern computational science and engineering. Although they are typically used to advance the system forward in time, they can also be viewed as nonlinear mappings that implicitly encode steady states and stability information. In this chapter, we present an extension of the matrix-free framework for calculating, via timesteppers, steady states of deterministic systems to stochastic particle simulations, where intrinsic randomness prevents direct steady state extraction. By formulating stochastic timesteppers in the language of optimal transport, we reinterpret them as operators acting on probability measures rather than on individual particle trajectories. This perspective enables the construction of smooth cumulative- and inverse-cumulative-distribution-function ((I)CDF) timesteppers that evolve distributions rather than particles. Combined with matrix-free Newton–Krylov solvers, these smooth timesteppers allow efficient computation of steady-state distributions even under high stochastic noise. We perform an error analysis quantifying how noise affects finite-difference Jacobian action approximations, and demonstrate that convergence can be obtained even in high noise regimes. Finally, we introduce higher-dimensional generalizations based

on smooth CDF-related representations of particles and validate their performance on a non-trivial two-dimensional distribution. Together, these developments establish a unified variational framework for computing meaningful steady states of both deterministic and stochastic timesteppers.

4.1 Introduction

Timesteppers are a powerful tool in modern scientific computation. Given the state $u(t)$ of our system at time t, i.e., a function or a collection of particles, and a time horizon h, a timestepper ϕ_h advances the state in time

$$u(t+h) = \phi_h\left(u(t)\right), \tag{4.1.0.1}$$

so that iterating the map generates a trajectory $\{u\left(nh\right)\}_{n=0}^{\infty}$. For stable iteration, the limit of $u\left(nh\right)$ approximates the (stable) steady-state solution of the underlying model. Traditionally, timesteppers are used "naively": we apply ϕ_h repeatedly to obtain a numerical trajectory that is subsequently studied for qualitative and quantitative convergence features.

Timesteppers, however, can also be usefully exploited beyond direct simulation. They embody everything about the problem, namely the underlying differential equation, the model parameters, and the boundary conditions. Instead of treating them only as simulators, one may ask what information about the system can be extracted directly from the timestepper mapping itself. Several important system-level tasks can be formulated in this way

1. Steady States. Fixed points $u^* = \phi_h\left(u^*\right)$ of the timestepper correspond to the steady (or invariant) states of the underlying differential equation.

2. Eigenvalues and Stability. The Jacobian $J = D\phi_h(u^*)$ of the timestepper at u^* contains all spectral information from which eigenvalues and stability properties of

the underlying model can be inferred. There is a direct relationship between the eigenvalues of the timestepper and the eigenvalues of the right-hand side of the differential equation.

3. Control and Identification. By systematically tracking the steady-state solutions across varying parameter values, we can trace the corresponding solution branches and identify critical points where the system's behavior changes drastically, such as folds, bifurcations, or Hopf bifurcations.

A convenient way to unify these three tasks is through the residual mapping

$$\psi(u) = u - \phi_h(u). \tag{4.1.0.2}$$

Here, $u \in \mathbb{R}^d$ represents the solution of some differential equation at a given time, typically on a grid, and $\phi_h, \psi : \mathbb{R}^d \to \mathbb{R}^d$ are vector functions. This formulation for calculating steady states allows us to apply the full machinery of iterative nonlinear solvers through matrix-free methods. Particularly in previous work [FEC+24, QEKK06, KKQ04], we used the Newton–Krylov and Arnoldi methods to calculate steady-state profiles and the leading eigenvalues of the Jacobian, respectively. These methods are matrix-free. We do not need explicit Jacobian matrices to compute the steady state or the leading eigenvalues. This last part is essential because codes that implement a timestepper rarely have their Jacobians coded as well. Even if they do so in the form of hand-coded formulas or automatic differentiation, evaluating the (typically large) Jacobian is too time-consuming and uses a lot of memory. Matrix-free methods avoid these bottlenecks by approximating the action of the Jacobian $D\psi(u)v$ in a particular direction v on the fly using finite differences

$$D\psi(u)v \approx \frac{\psi(u + \varepsilon v) - \psi(u)}{\varepsilon}. \tag{4.1.0.3}$$

Although this approximation induces an error, it can often be ignored in view of the

speedups that can be obtained.

In our previous work [KKQ04, QEKK06], we focused on deterministic timesteppers, developing Newton–Krylov and other matrix-free methods to compute steady states and stability information directly from the residual $\psi(u)$. Most recently we also applied the Newton–Krylov method to neural operators [FVG+25]. This line of research established how timesteppers could serve as fixed-point solvers for macroscopic equations, particularly in gradient systems where convergence is theoretically guaranteed.

In stochastic settings, the naive formulation $X - \phi_h(X) = 0$ for a steady state particle ensemble X is ill-defined. Timesteppers are no longer deterministic mappings, but rather involve random evolutions of particle ensembles. Individual particles do not carry steady-state information directly. Indeed, even in steady state, the particles typically fluctuate because of thermal noise. This does not mean that there is no steady state distribution of stochastic particles, but only that the mathematical equality $X = \phi_h(X)$ does not hold. As a consequence, solving $\psi(X) = 0$ for a particle ensemble using Newton-Krylov is useless. In this chapter, we re-interpret what a steady-state of particles means on the level of this timestepper residual.

The idea of computing steady-state distributions from particle timesteppers is not new. In [SGK12], the authors average many independent particle simulations to obtain smooth, macroscopic quantities. They propose a smooth coarse timestepper in terms of these macroscopic variables and use the Newton–Krylov method to compute (macroscopic) steady states and bifurcation diagrams. A similar idea appears in [QEKK06], where an approximate coarse model is introduced to precondition the Newton–Krylov method. The authors also used this preconditioned matrix-free framework to calculate the leading eigenvalues of the Jacobian and to track steady states along the bifurcation diagram. For self-similar systems, coarse representations based on cumulative distribution functions (CDFs) have also been used; see [CDGK04, ZKG05, ZKG06]. In these works, the coarse CDF description is repeatedly lifted to fine-scale particle ensembles — typically via inverse CDF

(ICDF)–based sampling — in order to evolve the system forward in time and ultimately compute steady or self-similar states. Finally, we also mention the work of [Sie10] where kernel-density estimation was applied to stochastic timesteppers in bacterial chemotaxis to calculate the steady-state distribution.

In this chapter, we go back to the original formulation (4.1.0.2) of steady states to understand what can be recovered when the timestepper is stochastic. As we said earlier, a steady-state ensemble X does not solve $\psi(X) = 0$, but some information must still be contained in this residual. Optimal Transport (OT) provides a principled way to compare ensembles in a Wasserstein space, enabling the construction of well-defined particle-based timesteppers. This extension allows us to treat both deterministic and stochastic dynamics within a unified variational framework. From an optimal transport perspective, we treat the evolving distribution of particles as a time dependent probability measure μ_t on $\mathbb{R}^d$, rather than trying to follow each particle individually. If μ_t is "smooth" in the sense of having a density and evolving in a regular way, then there is a velocity field v_t such that the continuity equation

$$\partial_t \mu_t + \nabla \cdot (v_t \mu_t) = 0 \tag{4.1.0.4}$$

describes how mass flows. The minimal energy (and smoothest) velocity field v_t can be computed via the optimal transport map $T_{\mu_t}^{\mu_{t+h}}$:

$$v_t = \lim_{h \to 0} \frac{T_{\mu_t}^{\mu_{t+h}} - \mathrm{id}}{h}.$$

We have previously shown that under certain assumptions on the smoothness of $(\mu_t)_{t \geq 0}$, forward Euler methods using v_t are first order accurate in the Wasserstein distance W_2 (see Chapter 2).

The key advantage over tracking particles individually (which in stochastic or chaotic systems can be very noisy) is that this optimal transport–based method aggregates the motion of *mass* rather than individual sample paths. In many systems, especially those

with fast/slow scales, individual particle trajectories will jitter, be non-differentiable, or otherwise provide a noisy or misleading picture at short time scales. By contrast, the distribution μ_t often changes in a much smoother and more coherent manner. By fitting the velocity field from optimal transport between successive empirical distributions, one effectively filters out the microscopic stochasticity and focuses on how the bulk of the particles move. This produces a cleaner signal, which will allow for a more accurate Newton–Krylov step. In essence, optimal transport allows us to take a "Lagrangian" perspective without needing to actually track each individual particle. The velocity fields we get from OT are smoother and better describe the overall distribution's behavior, but they still fundamentally capture the dynamics from the perspective of individual agents (as opposed to looking at a particular location, as in the "Eulerian" perspective).

These optimal transport algorithms are especially efficient in one dimension. In fact, the optimal transport map of a set of particles can be obtained simply by sorting the particles. This is because the optimal transport map from the uniform distribution on $[0, 1]$ to any probability distribution μ is exactly the inverse of the cumulative density function (ICDF) of μ [San15, Chapter 2]. Moreover, in one dimension, we can compose optimal transport maps to produce new optimal transport maps (i.e., $T_{\mu_t}^{\mu_{t+2h}} = T_{\mu_{t+h}}^{\mu_{t+2h}} \circ T_{\mu_t}^{\mu_{t+h}}$), and so computing optimal transport maps is nothing more than computing CDFs and ICDFs. The advantage of the CDF/ICDF formulation is that inverse CDFs are always smooth (as long as the underlying distribution is absolutely continuous), which allows us to employ advanced optimization techniques for calculating steady states of particle systems. We use this continuous case to motivate an ICDF-to-ICDF timestepper using finite particles and the stochastic timestepper: 1) Sample the ICDF by evaluating it in percentiles; 2) Propagate the particles using the stochastic timestepper; 3) Compute the new ICDF from the particle locations. This scheme gives us a bridge between microscopic particle simulations and macroscopic evolution, and is reminiscent of equation-free methods [KGH$^+$03, KGH04, Gea01].

In multiple dimensions, while the optimal transport velocity fields are still the appropriate objects to consider, computing them is much more challenging. Instead of simply sorting the particles (or computing a CDF), one needs to solve a linear program, which is much more computationally expensive. Thus, in practice, one must construct an approximation in a systematic way. For example, while the inverse cumulative density function is not well-defined in higher dimensions, the regular cumulative density function does carry over to multiple dimensions. This means we can construct a CDF-to-CDF timestepper using the same three steps: 1) Generate samples from the multidimensional CDF; 2) Propagate the particles using the stochastic timestepper; 3) Compute the new CDF from the particles. However, the first step becomes nontrivial in multiple dimensions because simply inverting the CDF at 'percentiles' for sampling is impossible. We need to project the multidimensional CDF down to its one-dimensional representations to generate samples. One such technique is to use marginal and conditional CDFs [ZKG05, ZKG06]. First, we construct the marginal CDF in the first dimension and generate x-samples from this 1D CDF by inverting it. Then, for every 1D sample, we construct the conditional CDF of the second dimension y and invert it. This procedure is analogous to the OT/ICDF method described above, except it comes from using the Knothe-Rosenblatt coupling instead of the true optimal coupling. Knothe-Rosenblatt maps can be expressed as a limit of optimal transport maps with appropriate cost functions, and so we can interpret them as approximations of OT maps. This Knothe–Rosenblatt formulation allows for an efficient extension of an OT-based timestepper in higher dimensions, though at the cost of introducing asymmetry. Other approaches based on sliced optimal transport (i.e., the sliced Wasserstein distance) also offer a flexible alternative, projecting the problem onto many one-dimensional slices before reassembling the results [BRPP15, KNS+19]. Both types of sampling strategies are crucial to making OT-based timesteppers practical for complex and higher-dimensional systems.

Having reinterpreted stochastic timesteppers in optimal transport terms, we can

now deploy advanced numerical solvers to compute steady-state distributions. However, there is one computational roadblock. The (I)CDF-to-(I)CDF timestepper is itself stochastic because it is built around a particle timestepper. Averaging over independent runs or variance reduction might reduce this stochastic variability [QEKK06], but evaluations of ψ will always be non-deterministic. However, we will numerically demonstrate that one can still compute steady-state (I)CDFs of stochastic systems accurately, up to a certain noise level or tolerance.

Contributions of this work This work extends the timestepper-based framework for steady-state computation from deterministic to stochastic particle systems by combining matrix-free Newton–Krylov methods with Optimal Transport formulations. Our main contributions are as follows:

1. **Generalization of timestepper-based fixed-point computation to stochastic systems.** We extend the classical residual formulation $\psi(u) = u - \phi_h(u)$ to stochastic particle timesteppers, where individual particle mappings are ill-defined due to randomness.

2. **Formulation of stochastic timesteppers in the language of Optimal Transport.** We reformulate timesteppers as operations acting on probability measures, and use the Wasserstein distance to define a well-posed residual that is minimized at steady state.

3. **Development of smooth distribution-based timesteppers.** We develop smooth (I)CDF-to-(I)CDF timesteppers that bridge stochastic simulations with distributional evolution, enabling the use of higher-order optimization such as the Newton–Krylov method.

4. **Efficient extensions to higher dimensions.** We propose two practical multidimensional generalizations: a CDF-to-CDF timestepper based on marginal and conditional CDFs, and a sliced Wasserstein formulation relying on (random) one-dimensional projections.

5. **Numerical validation and demonstrations.** We demonstrate the performance of the proposed methods on representative stochastic systems, including a two-dimensional distribution, highlighting accuracy and robustness of Newton–Krylov on smooth representations of particle distributions.

Outline of the Chapter We start this chapter in Section 4.2 by discussing steady-state distributions of particle systems in the language of optimal transport. We particularly focus on the notion of steady state through a reformulation of (4.1.0.2) using the Wasserstein distance. We further relate these as steady-states of the Euler-OT timestepper and discuss its connection to the OT map and the velocity field. From these fundamentals, we build a first-order optimization scheme by minimizing the Wasserstein distance between particles and a time-evolved copy of those particles. We particularly derive an Adam–Wasserstein optimizer and demonstrate, theoretically and numerically on three examples, that this Wasserstein optimization scheme converges quickly and smoothly to the correct steady-state particle distribution. Next, in Section 4.3 we introduce the matrix-free Newton–Krylov method. We then note that second-order optimizers applied to this Wasserstein formulation suffer from large statistical errors. This result motivates us to investigate smooth representations of particle distributions, and in particular the ICDF-to-ICDF timestepper in Section 4.4. We explain the main idea in detail and demonstrate an improved convergence of the Newton–Krylov method using these smooth representations. Finally, in Section 4.5, we discuss two alternative smooth representations for higher-dimensional systems: the CDF-to-CDF timestepper and its equivalent sliced Wasserstein formulation. We then demonstrate their effectiveness on a two-dimensional half-moon probability distribution. We conclude this chapter with a summarizing discussion and outlook for further research in Section 4.6.

4.2 Stochastic Particle Steady States and Optimal Transport

In previous work, we developed a Newton–Krylov method for computing steady states of deterministic timesteppers. In the deterministic setting, we wrote the fixed-point condition in residual form

$$\psi(x) = x - \phi_h(x) = 0. \tag{4.2.0.1}$$

For particle timesteppers, however, a one-step map is inherently random. That is, $y = \phi_h(x;\xi)$ where ξ is the collection of random numbers used in propagating the particles. Even when the system is "in equilibrium", the individual particles keep moving due to the intrinsic noise ξ. An example is the random motion of molecules, even in thermal and statistical equilibrium. More generally, particles move at certain probabilities and 'detailed balance' must be maintained. Hence the point-wise condition $\psi(x) = x - \phi_h(x) = 0$ is not well defined.

Instead, the steady state must be understood at the level of distributions: a distribution $\mu^\star$ is stationary when the distribution obtained after one step from $\mu^\star$ coincides with $\mu^\star$ again. However, distributions are not readily available from particle codes and must be built using histograms or tools like kernel density estimation. These methods for approximating the underlying distribution are sometimes as noisy as the particles themselves, and they are resource-intense.

To address this problem, we cast this distributional steady state as an optimization problem. We work in Wasserstein geometry and measure the mismatch between the current distribution and its one-step image using the 2-Wasserstein distance. Writing $X \sim \mu$ and $Y = \phi_h(X,\xi)$, the objective

$$\mathcal{J}(\mu) = \tfrac{1}{2}\, W_2^2\big(\mu,\ \mathrm{law}(Y)\big) \tag{4.2.0.2}$$

is nonnegative and vanishes if and only if μ is stationary. The Wasserstein distance directly

ties into the heart of Optimal Transport theory, and we present some relevant results here.

4.2.1 Optimal Transport Background

Much of the background we present here is discussed in more detail in our previous work [KNK+25]. We highlight the major results that will be particularly important for us in this chapter, but for a more complete presentation of the theory, we refer the reader to [KNK+25].

Let $\mathcal{P}_2(\mathbb{R}^d)$ be the set of Borel probability measures on $\mathbb{R}^d$ with finite second moment, i.e., the probability measures μ such that $\int_{\mathbb{R}^d} \|x\|_2^2 \, d\mu(x) < \infty$. For $\sigma, \mu \in \mathcal{P}_2(\mathbb{R}^d)$, we define $\Gamma_{\sigma,\mu}$ to be the set of all couplings between σ and μ:

$$\Gamma_{\sigma,\mu} := \{\gamma \in \mathcal{P}_2(\mathbb{R}^d \times \mathbb{R}^d) : \gamma(A \times \mathbb{R}^d) = \sigma(A) \text{ and } \gamma(\mathbb{R}^d \times A) = \mu(A) \text{ for all Borel } A \subseteq \mathbb{R}^d\}. \tag{4.2.1.1}$$

We then define the (2-)Wasserstein distance between σ and μ to be

$$W_2(\sigma, \mu) := \min_{\gamma \in \Gamma_{\sigma,\mu}} \left(\int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 \, d\gamma(x, y) \right)^{\frac{1}{2}}. \tag{4.2.1.2}$$

The coupling γ that satisfies (4.2.1.2) is called an "optimal coupling" or an "optimal plan".

It is well known that W_2 defines a metric on $\mathcal{P}_2(\mathbb{R}^d)$ (see, e.g., [PC19]). It is also well known that the optimal coupling (4.2.1.2) can be satisfied by a proper map $x \mapsto y = T(x)$ under hypotheses:

Theorem 4.2.1.3 (Brenier [Bre91])**.** *Let $\sigma, \mu \in \mathcal{P}_2(\mathbb{R}^d)$. If σ has a density with respect to the Lebesgue measure, then the optimal coupling γ that satisfies* (4.2.1.2) *is unique, and there exists a σ-a.e. unique map $T : \mathbb{R}^d \to \mathbb{R}^d$ such that $\gamma = (\mathrm{id}, T)_{\#}\sigma$. That is,*

$$W_2(\sigma, \mu) = \left(\int_{\mathbb{R}^d} \|x - T(x)\|_2^2 \, d\sigma(x) \right)^{\frac{1}{2}}. \tag{4.2.1.4}$$

Moreover, there exists a σ-a.e. unique (up to additive constant) convex function ϕ such that $T = \nabla\phi$.

We call the map T given in Theorem 4.2.1.3 the "optimal transport map" from σ to μ, and we denote it T_σ^μ.

While optimal transport on discrete measures does not generally admit optimal *maps* as in Brenier's Theorem, there is an important case where we do get optimal maps:

Theorem 4.2.1.5 (Proposition 2.1 in [PC19]). *If $\sigma = \frac{1}{N}\sum_{i=1}^N \delta_{x_i}$ and $\mu = \frac{1}{N}\sum_{i=1}^N \delta_{y_i}$ are uniform measures on the same number of distinct points, then there exists an optimal transport map T_σ^μ in the sense that*

$$W_2(\sigma,\mu) = \left(\frac{1}{N}\sum_{i=1}^N \|x_i - T_\sigma^\mu(x_i)\|^2\right)^{\frac{1}{2}}$$

is minimal. Moreover, if T_σ^μ is such a map, then there exists a permutation $\tau \in S_N$ (the symmetric group on N elements) such that $T_\sigma^\mu(x_i) = y_{\tau(i)}$ for all $i = 1, \ldots, N$.

4.2.2 Evolving Measures as Curves in Wasserstein Space

Now we consider the setting where we have a probability measure evolving over time. Explicitly, consider a curve $\mu : [0,T] \to \mathcal{P}_2(\mathbb{R}^d)$ written as $t \mapsto \mu_t$. If μ_t is absolutely continuous (as a curve) with respect to the Wasserstein distance, then there exists a unique "minimal energy" velocity field $\mathbf{v}_t$ that satisfies the continuity equation

$$\partial_t \mu_t + \nabla \cdot (\mathbf{v}_t \mu_t) = 0,$$

and if μ_t has a density with respect to the Lebesgue measure for all times $t \in [0,T]$, then this velocity field is computable as a derivative of optimal transport maps:

$$\mathbf{v}_t = \lim_{h\to 0} \frac{T_{\mu_t}^{\mu_{t+h}} - \mathrm{id}}{h} \qquad \text{in } L^2(\mu_t). \tag{4.2.2.1}$$

Furthermore, the Benamou-Brenier formulation of the Wasserstein distance [BB00] gives

$$W_2(\mu_t, \mu_{t+h}) = \left(\int_t^{t+h} \|\mathbf{v}_s\|^2_{L^2(\mu_s)} \, ds \right)^{\frac{1}{2}}.$$

Notice that the above formulation says that an evolving distribution reaches steady state exactly when $\mathbf{v}_t = 0$. Moreover, when $\mathbf{v}_s \approx 0$ (meaning $\|\mathbf{v}_s\|_{L^2(\mu_s)} \approx 0$) for all $s \in [t, t+h]$, then $W_2(\mu_t, \mu_{t+h}) \approx 0$, and so the distribution has "approximately" reached steady state.

In our setting, however, we do not have access to μ_t for a continuous time interval. Instead, we have a timestepper ϕ_h for some small step h, and so we only have μ_t for discrete times $t_n = hn$. In this situation, we assume that the timestepper approximates the evolution of some continuous curve, and that we can approximate the velocity field $\mathbf{v}_t$ associated with this continuous curve using the finite-difference approximation suggested by (4.2.2.1):

$$\mathbf{v}_t \approx \mathbf{v}_t^h := \frac{T_{\mu_t}^{\mu_{t+h}} - \mathrm{id}}{h}.$$

Note that by (4.2.1.4), we have $W_2(\mu_t, \mu_{t+h}) = h\|\mathbf{v}_t^h\|_{L^2(\mu_t)}$, and so computing the Wasserstein distance between successive time steps is equivalent to computing the norm of the approximate velocity field. For more details about the link between evolving measures, velocity flow fields, and finite difference approximations, we refer to our previous work in [KNK+25].

4.2.3 Wasserstein Formulation of Particle Steady States

In practice, our timestepper acts on particles, not on continuous distributions. We can still use the ideas discussed above as a heuristic to suggest that the system's evolution is in (approximate) steady state when the Wasserstein distance between successive time steps is minimized.

That is, if we consider the initial locations of the particles $X = (x_1, \ldots, x_N)$ and their

locations $Y = \phi_h(X) = (y_1, \ldots, y_N)$ - after applying the timestepper over a time-horizon of size h - as discrete probability measures $\mu = \frac{1}{N}\sum_{i=1}^{N} \delta_{x_i}$ and $\nu = \frac{1}{N}\sum_{i=1}^{N} \delta_{y_i}$ respectively, then we can consider the objective function

$$F(X;\xi) := \frac{N}{2} W_2^2(\mu,\nu) = \frac{1}{2}\sum_{i=1}^{N} \|x_i - T_\mu^\nu(x_i)\|^2 = \frac{1}{2}\sum_{i=1}^{N} \|x_i - y_{\sigma^\star(i)}\|^2, \tag{4.2.3.1}$$

where $\sigma^\star$ is the optimal coupling - a permutation. Finding an approximate steady state is equivalent to (approximately) finding a minimizer of F over all initial particle locations X. Between changes of the optimal transport coupling, F is a smooth quadratic in the particle locations, and its (sub)gradient with respect to each x_i is the transport displacement from x_i to its match under the current optimal plan. This naturally leads us to first-order optimization methods to minimize (4.2.3.1) such as (stochastic) gradient descent or Adam [KB15].

Let $\Pi^\star$ denote the permutation matrix that corresponds to the optimal coupling, i.e., $\Pi^\star Y = (y_{\sigma^\star(1)}, \ldots, y_{\sigma^\star(N)})$. Now F reads

$$F(X;\xi) = \frac{1}{2}\|X - \Pi^\star Y\|^2$$

and so the gradient of F with respect to X is

$$\nabla F(X) = X - \Pi^\star Y + D\phi_h(X;\xi)^T \left(\phi_h(X,\xi) - (\Pi^\star)^\top X\right). \tag{4.2.3.2}$$

In principle, we can evaluate this gradient and plug it into any first-order optimization algorithm, such as the Adam optimizer.

From a numerical point of view, the second term of (4.2.3.2) poses a problem. Two subsequent evaluations of $F(X)$ for the same particles X will differ due to the influence of ξ. In particular, this means that the Jacobian of the timestepper $D\phi_h(X;\xi)$ will be very noisy, even if we can evaluate it exactly by automatic differentiation. More importantly, this

Jacobian will contain little actual information about the drift of the particles. Fortunately, by the Envelope theorem [AGS08, PC19], we can use only the first term of the gradient

$$\partial_X \tfrac{1}{2} W_2^2(X, Y) = X - \Pi^\star Y \tag{4.2.3.3}$$

for optimization purposes.

Remark 4.2.3.4. *This formulation gives a slightly different perspective on the Euler-type method we develop in detail in our previous work [KNK+25]. Note that, in the language of velocity fields from earlier, we have*

$$\mathbf{v}_t^h(x_i) = \frac{T_\mu^\nu(x_i) - x_i}{h} = \frac{1}{h}(y_{\sigma^\star(i)} - x_i),$$

and so this partial gradient of our objective function is nothing more than (a scaled copy of) the negative of our approximate velocity field $\mathbf{v}_t^h$. This means that our Euler-type method of taking steps in the direction described by the velocity field $\mathbf{v}_t^h$ can actually be interpreted as running gradient descent with the partial gradient $X - \Pi^\star Y$. The insight that the present functional-gradient formulation provides, is that it is now clear how to apply more sophisticated methods for minimizing the objective function (i.e., finding steady states), such as Adam, which we can interpret as a natural extension of our previous work.

There remains one more computational bottleneck in this Wasserstein-minimization algorithm, and that is the computation of the optimal transport map $\Pi^\star$. In one dimension, sorting both clouds and matching in order yields $\Pi^\star$ in $O(N \log N)$ time. This is the theoretically optimal complexity. In multiple dimensions, one may use a linear assignment solver which is typically very expensive, i.e., $O(N^3)$ [BDM12, PC19]. We will showcase the Wasserstein-minimization approach on both one- and two-dimensional examples in the next section.

For any initial set of particles X, we can evaluate $F(X)$ and its gradient $\partial_X F(X)$. We can therefore minimize F with any first-order scheme; we use Adam because of its stability under mild noise. A typical iteration is as follows

1. Given X_k and random numbers ξ_k, compute $Y_k = \phi_h(X_k; \xi_k)$.

2. Compute an optimal plan $\Pi_k^\star = \Pi^\star(X_k, Y_k)$.

3. Form the gradient $g_k = X_k - \Pi_k^\star Y_k$ (optionally average over a small batch of independent runs for variance reduction).

4. Apply Adam to obtain X_{k+1}.

We use PyTorch's Adam optimizer in our implementation. One advantage of PyTorch is that we can use its call graph to directly compute gradients of $F(X)$. The first term of that gradient can then be obtained by detaching Y from the call graph.

In summary, minimizing $F(X) = \frac{1}{2}W_2^2(X, \phi_h(X; \xi))$ with Adam yields a fast $O(N \log N)$ procedure that is robust to stochasticity and faithful to the Wasserstein fixed-point formulation, while avoiding the instability of second-order information at coupling changes.

4.2.4 Three Examples

The Wasserstein particle flow is a well-defined, easy-to-implement, and reliable approach to compute steady-state distributions of particle methods. Combining gradient flow with the Adam optimizer also has significant advantages such as steady decrease of the W_2^2-loss. In this section, we demonstrate the combined method on three examples: bacterial chemotaxis (section 4.2.4.1), the nonlinear economic agents model (section 4.2.4.2), and a two-dimensional overdamped Langevin dynamics with non-trivial potential energy profile (section 4.2.4.3).

4.2.4.1 Bacterial Chemotaxis

Consider a collection of bacteria X_t that eat food from a substrate $S(x)$ on the domain $[-L, L]$. The function $S(x)$ represents the amount of food at position x, and we assume

that S remains unchanged over time. Furthermore, there is a chemotactic sensitivity $\chi(S)$ that indicates the level of activation of the bacteria, solely a function of the local food supply. Finally, we also incorporate a diffusion constant D to model the random motion. The bacterial positions X_t over time are then governed by the overdamped Langevin equation [FJ17]

$$dX_t = \chi(S(X_t))S_x(X_t)dt + \sqrt{2D}dW_t, \tag{4.2.4.1}$$

where S_x is the food gradient and W_t is the standard Brownian motion. The bacteria also cannot leave $[-L, L]$ so we impose reflective or no-flux (Neumann) boundary conditions.

We discretize the overdamped Langevin dynamics in equation (4.2.4.1) using an Euler-Maruyama method with step size 10^{-3}, and per evaluation of the timestepper $\phi_h(\cdot)$ we integrate up to $h = 1$ seconds. Reflective boundary conditions are applied after every Euler-Maruyama step. We run the Adam optimizer with standard parameters ($\beta_1 = 0.9, \beta_2 = 0.99$) and an initial 'learning rate' of 10^{-1}. We decrease the learning rate by a factor 10 every 100 epochs, for a total of 300 epochs. The initial distribution is a truncated Gaussian with a mean 5 and a standard deviation of 2.

As we see in Figure 4.1, the Wasserstein-Adam optimizer reliably converges to a final loss of $2\ 10^{-5}$. The loss gradient also decreased to a similar value, indicating strong convergence. The histogram of the optimized particles (right figure, orange) also matches well with the analytic invariant distribution. Finally, we see in the figure on the left that the Wasserstein-Adam optimizer reached this steady-state already after about 150 epochs - or a total simulation time of 150 seconds. This is because the objective function is evaluated only once per Adam epoch. Compared to computing the steady-state by long-time evolution of (4.2.4.1), which takes about 500 seconds of in-simulation time, the Wasserstein-Adam method is much faster.

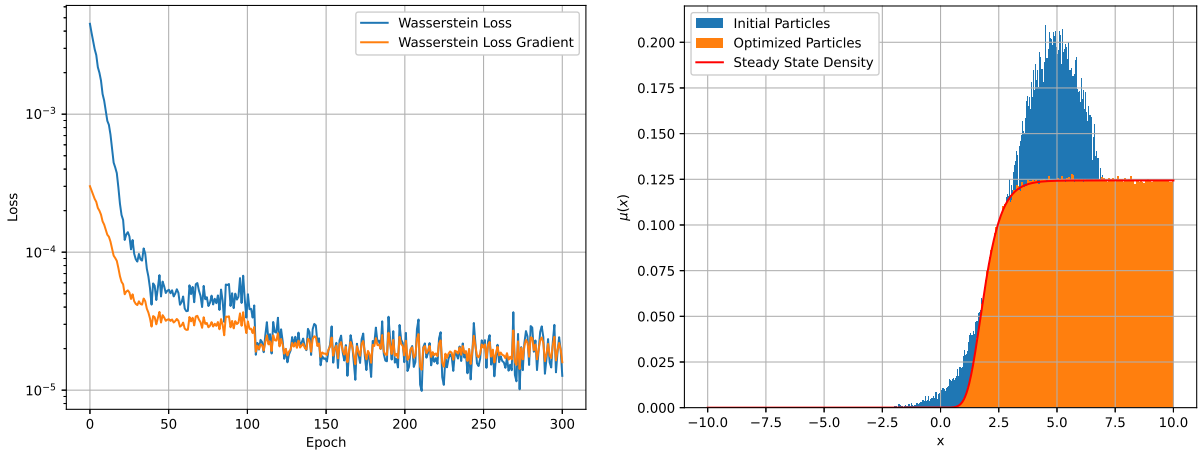


Figure 4.1: Numerical results for the Wasserstein-Adam optimizer on the chemotaxis model. Left: Wasserstein loss (blue) and gradient-norm (orange) per epoch. Right: Histogram of the initial particles in blue, the Wasserstein-Adam optimized particles in orange, and the analytic steady-state density (see equation (4.3.1.4)) in red.

4.2.4.2 Economic Agents

The Wasserstein-Adam method also works well for nonlinear models. Let us consider an example of economic agents trading stocks. Each agent is represented by a value $X_i(t) \in (-1, 1)$ that gives the tendency to buy ($X_i = 1$) or sell ($X_i = -1$) a certain stock. More details on the model can be found in [FEC+24]. We use the timestepper available at [Eva25] to integrate the agents up to $h = 1$ seconds at a time. This timestepper is unbiased for the dynamics of the agents' dynamics and uses the discretization presented in [FEC+24]. We again use an Adam optimizer with an initial learning rate of 10^{-1} and with standard parameters to find the steady-state distribution of the particles. The learning rate is decreased by a factor 10 every 100 epochs for a total of 300 epochs. The initial distribution is a Gaussian with zero mean and a standard deviation of 0.1. The steady-state distribution is wider and approximately normal.

Figure 4.2 displays the Wasserstein loss and gradient on the left, as well as optimized agents compared to the steady-state distribution on the right. The initial distribution of agents is shown too. We observe an excellent match between the histogram of optimized agents and the invariant distribution. The loss stagnates near $2\ 10^{-6}$, nearly four orders of

magnitude lower than the initial loss, demonstrating a clear convergence.

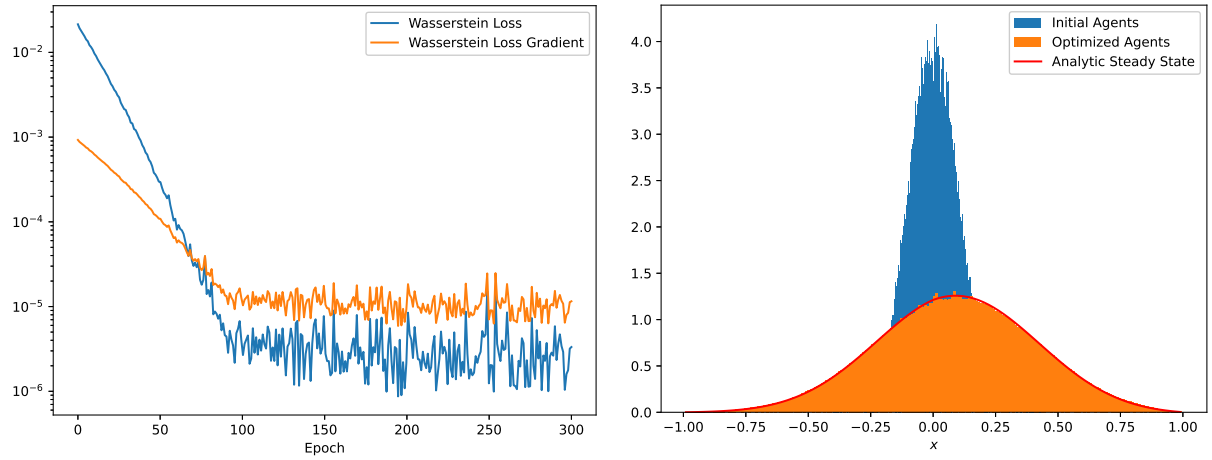


Figure 4.2: Numerical results for the Wasserstein-Adam optimizer on the economic agents model. Left: Wasserstein loss (blue) and gradient-norm (orange) per epoch. Right: Histogram of the initial particles in blue, the Wasserstein-Adam optimized particles in orange, and the analytic steady-state density (see Figure 4.6) in red.

4.2.4.3 The Half-Moon Potential

In multiple dimensions, one cannot exploit the monotone rearrangement used in 1D (sorting), so computing W_2 requires solving a discrete assignment problem between the two point clouds. We use the linear assignment solver, whose worst-case complexity is $O(M^3)$ for M paired points; applied to the full cloud, this would be $O(N^3)$. To keep costs manageable, we adopt mini-batches of size B: each step propagates B particles through the timestepper and solves an $O(B^3)$ assignment problem. These batches are drawn uniformly from the original N particles, and We process N/B batches per iteration, yielding a total $O(NB^2)$ complexity. The smaller the batch size, the smaller the computational cost. The rest of the Wasserstein–Adam flow is unchanged from 1D: we form $Y_b = \varphi_h(X_b)$, compute an optimal match $\Pi_b^\star$ on each batch b, use the detached gradient $\partial_X \frac{1}{2} W_2^2(X_b, Y_b) = X_b - \Pi_b^\star Y_b$, and update the particles X_b with Adam.

We note that the effective result of this is equivalent to solving a *constrained* optimal assignment problem, where we require the $N \times N$ permutation matrix Π to be block-

diagonal (i.e., the N/B diagonal $B \times B$ blocks are the permutation matrices Π_b^*). This permutation matrix is in general not the exact optimal coupling for the overall $N \times N$ assignment problem, but if each batch is chosen randomly, the batched coupling will be a close approximation of the optimal coupling — in particular, it still gives a good descent direction for use in Adam.

For this example, we consider a two-dimensional half-moon potential. The potential energy function $U(x, y)$ is given by

$$U(x, y) = A\,(r(x, y) - R)^2 + B \exp\left(-\alpha(y - y_s)\right) \tag{4.2.4.2}$$

with $A = 2, B = 0.5, R = 2, \alpha = 1.5$, and $y_s = -0.5$. The function $r(x, y) = \sqrt{x^2 + y^2}$ measures the distance from the origin. A two-dimensional color plot of the corresponding steady-state distribution

$$\mu(x, y) = Z^{-1} \exp\left(-U(x, y)\right) \tag{4.2.4.3}$$

is shown in Figure 4.3.

To compute the steady-state distribution using the Wasserstein-Adam method, we construct an overdamped Langevin dynamics of the form

$$d(X, Y) = -\nabla U(X, Y)dt + \sqrt{2}dW_t \tag{4.2.4.4}$$

where W_t is a two-dimensional Brownian motion with independent components. The steady-state distribution of this dynamics is precisely (4.2.4.3). We also implement reflective boundary conditions (e.g., Neumann) to keep all particles within the domain $[-4, 4]\times[-4, 4]$. We discretize this overdamped Langevin dynamics using the Euler-Maruyama timestepper with step size 10^{-3}, and integrate up to time $h = 0.1$ seconds. The initial distribution is a standard bivariate Gaussian, which has minimal overlap with (4.2.4.3).

To compute the Wasserstein loss, we divide the $N = 10^5$ particles into batches of

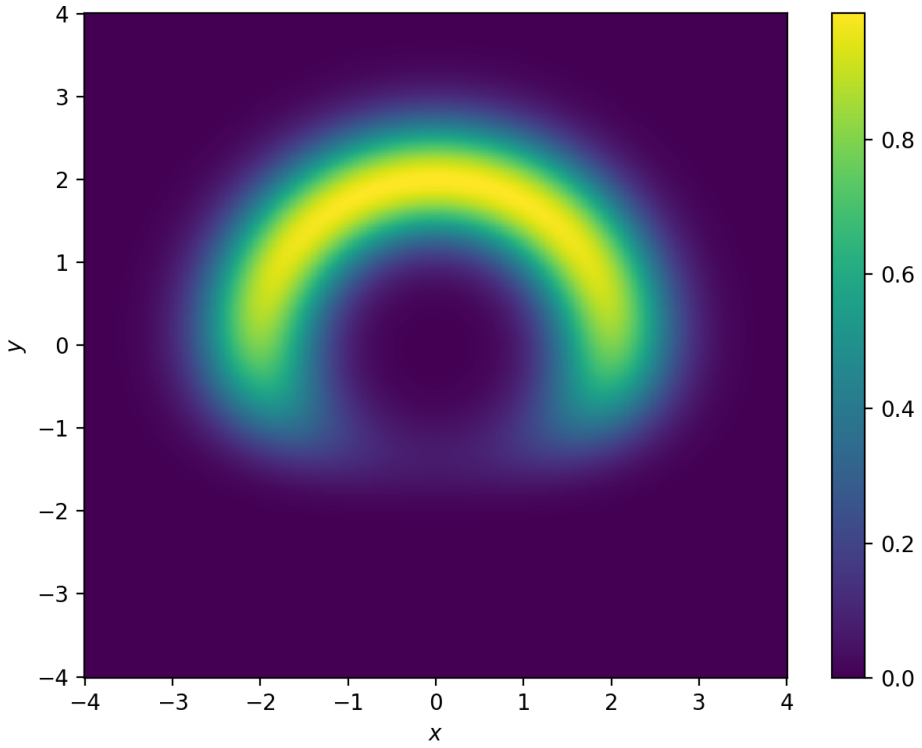


Figure 4.3: A color plot of the half-moon potential, see equation (4.2.4.2) for more details.

size 1000. For each batch, we propagate the particles through the timestepper and evaluate the loss using the OT coupling algorithm. We feed these batches to the Adam optimizer, which uses an initial learning rate of 10^{-2}, decreasing by a factor of 10 every 100 epochs, up to 300 epochs.

The optimization results are shown in Figure 4.4. The left panel reports the objective and gradient-norm trajectories: the initial $\frac{1}{2}W_2^2$ of 0.51233 decreases monotonically to 1.54×10^{-2}, effectively reaching the stochastic noise floor, and the loss gradient decays in tandem, indicating stable convergence. The right panel displays a 2D histogram (color map) of the optimized particles, which shows an excellent match to the analytic steady-state distribution from figure 4.3 with no visible systematic bias.

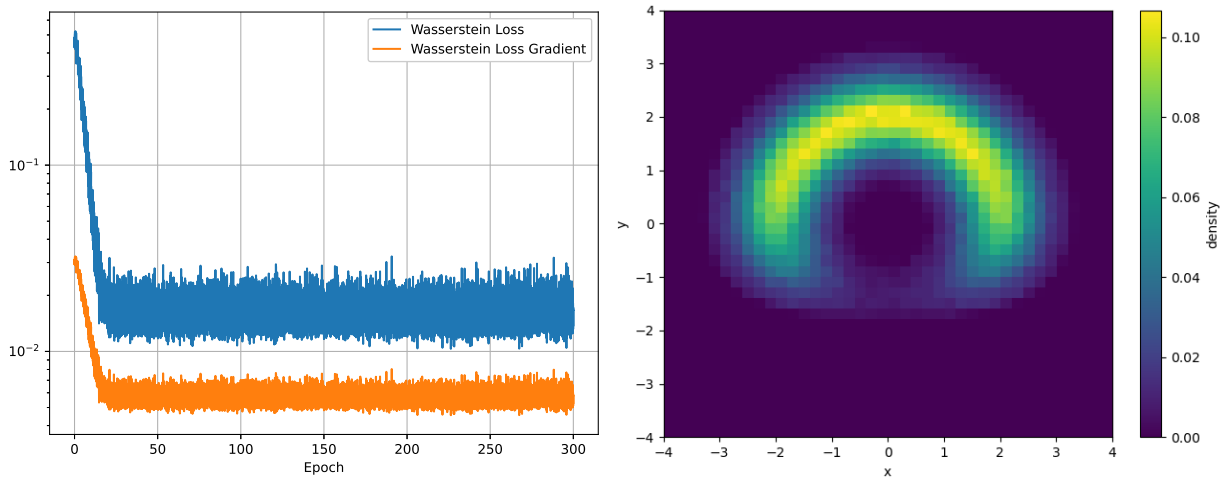


Figure 4.4: Numerical results for the Wasserstein-Adam optimizer on the half-moon potential. Left: Wasserstein loss (blue) and gradient-norm (orange) per epoch. Right: Colormap of the 2D histogram of the optimized particles in orange.

4.3 Newton–Krylov Framework

First-order optimizers typically converge slowly to the steady state, as they take only small steps in the direction of the local gradient. As a result, they require (relatively) many iterations and repeated evaluations of the Wasserstein objective and gradient, each of which involves the costly computation of an optimal transport map. This makes first-order methods particularly expensive in Wasserstein-based formulations. We want to recover the second-order performance of Newton–Krylov by solving

$$F(X) = \partial_X W_2^2\big(X,\ \phi_h(X)\big) = 0 \tag{4.3.0.1}$$

and approximating Jacobian–vector products as

$$DF(X)\cdot v \approx \frac{F(X+\varepsilon v) - F(X)}{\varepsilon}. \tag{4.3.0.2}$$

In practice this is ill-posed and noisy for particle OT. The mapping $X \mapsto \partial_X W_2^2(X, \phi_h(X, \xi)$ is only piecewise smooth: as particles move, the optimal transport coupling changes combinatorially, creating kinks where $DF(X)$ does not exist. Difference quotients then mix

values across distinct couplings, producing large, non-vanishing errors that blow up as $\varepsilon \to 0$. Stochasticity compounds the problem: tiny perturbations $X \mapsto X + \varepsilon v$ can reshuffle assignments in the empirical OT, injecting high-variance jumps into $F(X + \varepsilon v) - F(X)$. Krylov iterations driven by such JVPs become unstable, and line searches lose reliability. To account for this noise, we propose a "smoothed" representation of particle codes in Section 4.4.

The idea of Newton's method is to iteratively update $u_{k+1} = u_k + s_k$ by choosing s_k such that the first order approximation of $\psi(u_{k+1}) \approx 0$. Explicitly, we want s_k such that $\psi(u_k) + D\psi(u_k)s_k = 0$. The main idea behind the Newton–Krylov method is to approximate the action of the Jacobian of the objective function $\nabla\psi(u_k)$, using finite differences with step size ε

$$D\psi(u_k) \cdot v \approx \frac{\psi(u_k + \varepsilon v) - \psi(u_k)}{\varepsilon}. \tag{4.3.0.3}$$

Since this matrix-free formulation can only approximate matrix-vector products, and not the full Jacobian directly, we must use iterative Krylov methods to solve the linear system

$$D\psi(u_k)s_k = -\psi(u_k) \tag{4.3.0.4}$$

to obtain the next Newton guess $u_{k+1} = u_k + s_k$. Because the Jacobian is not generally symmetric, we will use the GMRES method for solving (4.3.0.4).

4.3.1 Newton–Krylov for Mean-Field Equations

As a first step towards using the Newton–Krylov method for calculating steady-state solutions, let us look at the case when the underlying model for the particle timestepper is a partial differential equation, specifically a mean-field or Fokker-Planck equation. We prefer the terminology of the mean-field equation since the Fokker-Planck equation is only valid for linear models; however, we also consider nonlinear stochastic models here.

We should note that our ultimate goal is to compute steady-state distributions of

particle timesteppers. The mean-field PDE, although a useful mathematical concept, is not immediately known or available for most systems and particle codes. In the few cases where it is known, the mean-field model is usually an approximation through some kind of closure. However, in the settings where the mean-field model is known and exact - like the linear Fokker-Planck equation - steady-state distributions are directly encoded, and we can compute it using a Newton–Krylov method. We investigate this idea in this section.

Suppose we have a collection of particles or agents $\{X_i\}_{i=1}^n$ that follow a stochastic process. We will discuss two examples below. In the limit of $n \to \infty$, the particles or agents are distributed according to a time-dependent probability distribution $\mu_t(x)$, where x represents the spatial locations. This probability distribution, in turn, follows a deterministic and coarse mean-field equation of the form

$$\partial_t \mu_t(x) = F(\mu_t(x)) \tag{4.3.1.1}$$

with the appropriate boundary conditions. On this level, all stochasticity has been removed, and we can simply use the regular Newton–Krylov method to compute the steady-state distributions. Most mean-field models never have their right-hand sides coded, so we will rely on a coarse timestepper $\Phi_h(\mu_t)$ that progresses the current distribution μ_t over a time window of size h to μ_{t+h}. Analogous to equation (4.1.0.2), we will solve

$$\Psi(\mu) = \mu - \Phi_h(\mu) = 0 \tag{4.3.1.2}$$

for μ — the steady-state distribution. We will demonstrate this approach on two examples from last section: the linear chemotaxis model (section 4.3.1.1) and the nonlinear economic agents model (section 4.3.1.2).

4.3.1.1 Bacterial chemotaxis model

The coarse Fokker-Planck equation for the spatial distribution $\mu(x,t)$ of the bacteria from section 4.2.4.1 reads

$$\partial_t \mu = -\partial_x(\chi(S(x))S_x(x)\mu) + D\partial_{xx}\mu, \tag{4.3.1.3}$$

and describes how the concentration of bacteria changes over time. This equation is sometimes known as the Keller-Segel model for chemotaxis [Ste00]. The correct no-flux boundary conditions are $J(\pm L) = 0$ with flux

$$J(x) = \chi(S(x))S_x(x)\mu(x,t) - D\mu_x(x,t).$$

It admits a unique steady-state distribution of the form

$$\mu(x) = Z^{-1} \exp\left(\frac{1}{D}\int_{-1}^{S(x)} \chi(S)dS\right), \tag{4.3.1.4}$$

where Z is the normalization constant.

In our example, $S(x) = \tanh(x)$ and $\chi(S) = 1 + \frac{1}{2}S^2$. We compute the steady-state distribution using the Newton–Krylov method based on a finite volumes discretization of (4.3.1.3) with $N = 1000$ equidistant grid points representing the volume centers. That is, the solution will be a vector $u \in \mathbb{R}^N$ representing the values of the steady-state distribution at the fixed volume centers. For timestepping, we use an explicit Euler time discretization over a time window of size $h = 1$ second.

Figure 4.5(a) shows the steady-state distribution obtained by the Newton–Krylov method alongside the analytic formula (4.3.1.4). We observe an excellent correspondence between these two distributions. Additionally, in Figure (b), we observe that the Newton–Krylov method reaches a residual of 10^{-8}, the minimal obtainable residual according to the analysis in [BWWW08], after only 9 steps.

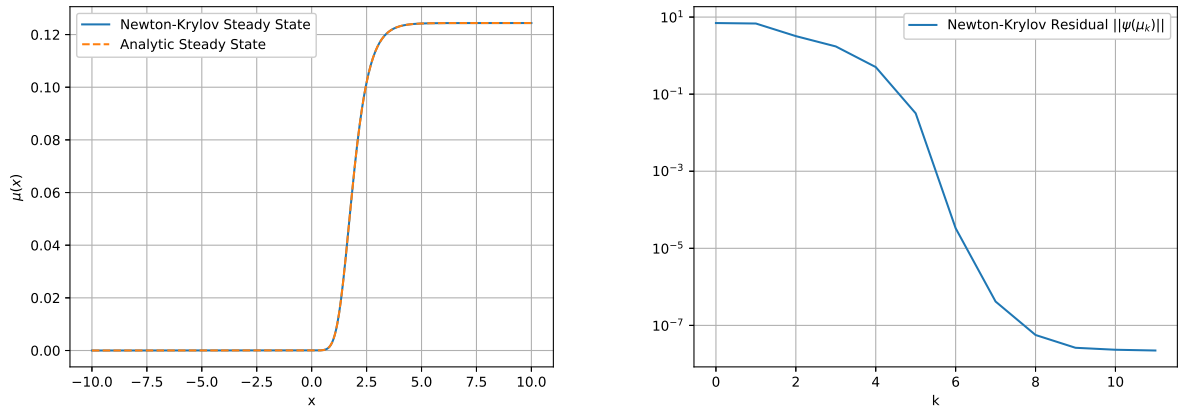


Figure 4.5: (Left) Steady-state distribution of the Keller-Segel model: Newton–Krylov (blue) vs. analytic solution (dashed orange). (Right) Newton–Krylov residual per iteration.

Remark 4.3.1.5. *Although the Fokker-Planck equation* (4.3.1.3) *and the* Ψ*-function are linear in* μ_t*, the Newton–Krylov method is not guaranteed to converge to the steady-state distribution in one step, unlike the regular second-order Newton method. The reasons are twofold. First, the Newton–Krylov method does not solve the exact Jacobian system* $D\psi(x_k)s_k + \psi(x_k) = 0$*, rather an approximation* $D\tilde{\psi}(x_k)s_k + \psi(x_k) = 0$*. Secondly, we typically use a fixed tolerance* $\eta_k = \eta > 0$ *for all linear system solves. So instead of converging to the real steady state distribution, the Newton–Krylov method will stay within a tolerance* η *of the exact steady state.*

4.3.1.2 Economic Agents Model

As a second demonstration of using Newton–Krylov to compute steady-state distributions of mean-field equations, let us reconsider the example of economic agents trading stocks. For this example, we look at the deterministic density $\mu(x,t)$ of the economic agents at location x and time t. It can be shown that an approximate mean-field equation exists, of the form

$$\partial_t \mu = \frac{1}{2}\sigma^2(t)\partial_{xx}\mu + \partial_x\left(b(x,t)\mu\right) + \left(J^+ + J^-\right)\delta(x). \tag{4.3.1.6}$$

In this model, $b(x,t)$ and $\sigma^2(t)$ are the time-dependent drift and diffusivity of the agents, and J^+ and J^- are integral operators with δ the Dirac-delta distribution. Further details on this model can be found in Appendix A of [FEC+24]. Importantly, this mean-field PDE is nonlinear, non-local and admits multiple steady-state distributions.

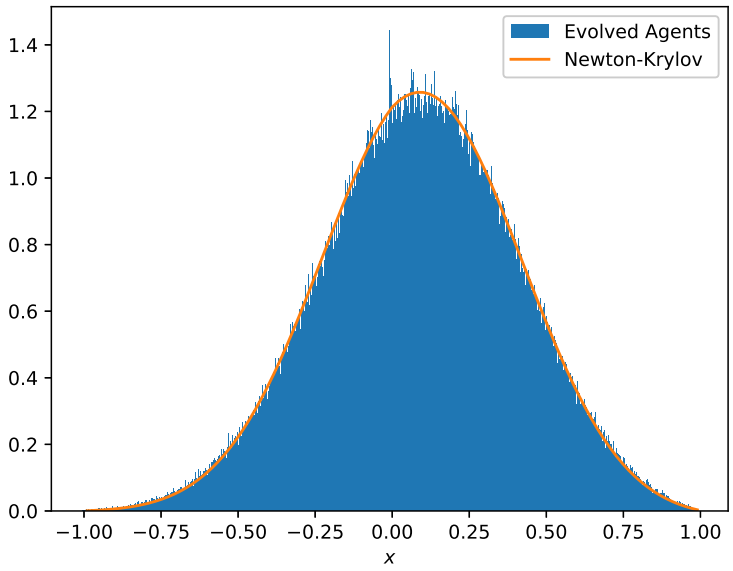


Figure 4.6: Steady-state distribution from Newton–Krylov versus time evolution of stochastic agents.

For the following numerical experiment, we discretized (4.3.1.6) using a finite-volume method with $N = 100$ equidistant volume centers. Our PDE timestepper uses an explicit Euler method with step size 10^{-4}, and we integrate over a time window of size $h = 11$ second. Figure 4.6 shows the steady-state distribution calculated by Newton–Krylov method in orange. It also shows the histogram distributions of the random agents obtained by timestepping the stochastic model to steady state. There is an excellent visual agreement between the two, as well as in the average agent position $\langle X_i \rangle$ of 0.0892.

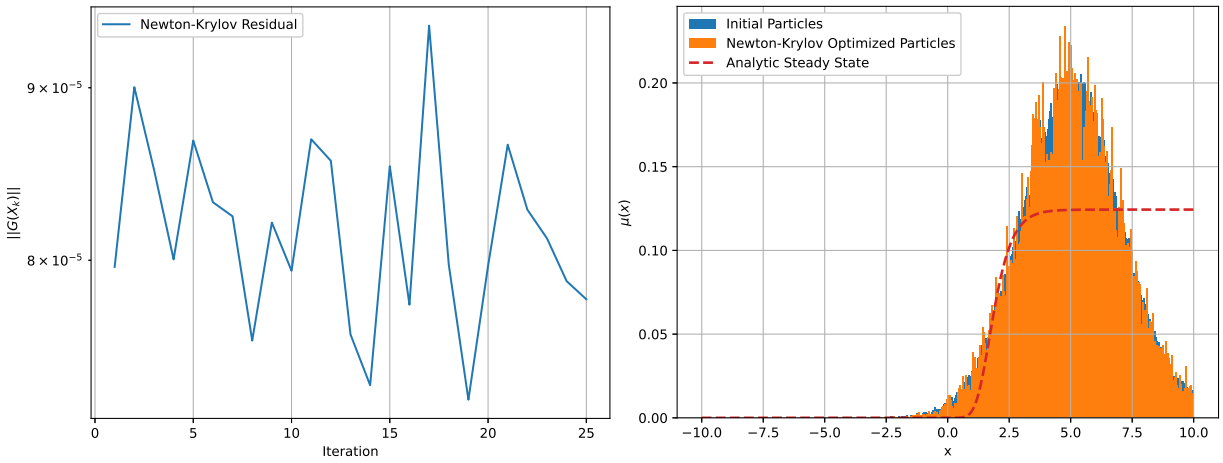


Figure 4.7: Numerical results for the Newton–Krylov optimizer for bacterial chemotaxis. Left: Norm of the objective function, $\left\|\tilde{F}(X_k)\right\|$ per iteration. Right: Initial particles (blue), optimized particles (orange) and analytic steady-state distribution (dashed red). We notice that the optimized distribution has barely moved, which demonstrates that Newton–Krylov at the particle level is too noisy to efficiently find the steady state.

4.4 From Particles to Smooth Representations

To motivate this section, we begin by demonstrating that, even with an optimal finite-difference step size, Newton–Krylov at the particle level cannot discern any gradient information. To demonstrate this effect, let us consider the bacterial chemotaxis example again from section 4.2.4.1. In Figure 4.7 we show the Newton–Krylov optimization result after 100 steps with $N = 10^5$ particles and $\varepsilon = 0.1$. The optimized particles barely moved from their initial distribution. Figure 4.7 shows the norm of the objective function (4.3.0.1) in all iterations, revealing that the error remains essentially unchanged in expectation. The reason for this behavior is that although each application of the timestepper moves the particles closer to the steady state, the subsequent Newton–Krylov update perturbs them in essentially random directions. As a result, the timestepper is forced to repeatedly recover the same progress, preventing any net convergence.

Newton–Krylov may not work on the level of noisy particles (as seen in Figure 4.7), but it can work if we add smoothness to the optimization criterion; enter the Inverse Cumulative Density Function. In one dimension, optimal transport with respect to the

Wasserstein-2 distance admits an especially convenient formulation. Composing the ICDF with the CDF of the particles is exactly equivalent to computing the optimal transport map, ensuring that the geometry of the problem is preserved. Furthermore, if the ICDF is known in fixed (grid-) percentiles, we can interpolate it using splines, making the (interpolated) ICDF a smooth and monotonic representation of the empirical measure. The ICDF is continuously differentiable, so we can employ Newton-type methods to calculate steady states. Finally, evaluating the ICDF at prescribed percentiles corresponds to drawing samples from the distribution.

These combined elements make it possible to define a timestepper directly at the level of the ICDF. Starting from the current ICDF, we first sample N particles by evaluating the ICDF in the (fixed) k/N percentiles. Next, we propagate these samples through the stochastic particle timestepper, after which we construct the new ICDF by retaining some of the (sorted) particles. In this representation, derivatives are well defined, and Newton–Krylov iterations can be applied meaningfully, recovering the fast convergence that is lost in the purely particle-based formulation.

4.4.1 The ICDF-to-ICDF Timestepper

Given a one-dimensional probability density $\mu(x)$ on a fixed interval $[a, b]$, the cumulative density function is defined as

$$F(x) = \int_a^x \mu(y)dy, \quad x \in [a, b] \tag{4.4.1.1}$$

Because F is monotonic, it is injective and inverse cumulative density $F^{-1}(p)$ exists. This ICDF is defined for $p \in [0, 1]$ and $x = F^{-1}(p)$ corresponds to the p-th percentile of μ.

For a discrete set of (sorted) particles $\{X_n\}_{n=1}^N$, the cumulative density function is

piecewise continuous with jumps at the locations

$$F(X_n) = \frac{n}{N}. \tag{4.4.1.2}$$

The inverse cumulative density function is also piecewise continuous defined in the discrete percentiles $F^{-1}(n/N) = X_n$. To reduce the impact of precise particle locations X_n, we propose to evaluate the ICDF in a fixed percentile grid $p_k, k = 1, \ldots, K$ - typically uniform between 0 and 1 - with $K \ll N$. This allows us to construct a coarse ICDF-to-ICDF timestepper

$$\Phi_h\left(F_t^{-1}\right) = F_{t+h}^{-1} \tag{4.4.1.3}$$

in four stages:

1. Interpolate $F^{-1}(t)$ on the fixed grid $p_k, k = 1, \ldots, K$ using a differentiable spline;
2. Sample $\{X_n(t)\}$ by evaluating $F^{-1}(t)$ in uniform percentiles n/N;
3. Propagate the particles to $\{X_n(t+h)\}$ using the particle timestepper ϕ_h;
4. Construct the new ICDF by sorting $\{X_n(t+h)\}$ and retaining every K-th particle.

A schematic of the ICDF-to-ICDF timestepper is shown in Figure 4.8.

By construction, both the ICDF representation and its spline approximation ensure that the timestepper maps one smooth ICDF into another. In this setting, finite-difference approximations of the directional derivative $D\Phi_h \cdot v$ no longer suffer from variance-amplifying $1/\varepsilon$ terms, since the noise inherent at the particle level has been averaged out by the smooth representation. As a result, Newton–Krylov iterations regain their expected fast convergence toward the steady-state distribution, now expressed consistently through the ICDF. The associated ICDF-based objective function is given by

$$\Psi(F^{-1}) = F^{-1} - \Phi_h(F^{-1}). \tag{4.4.1.4}$$

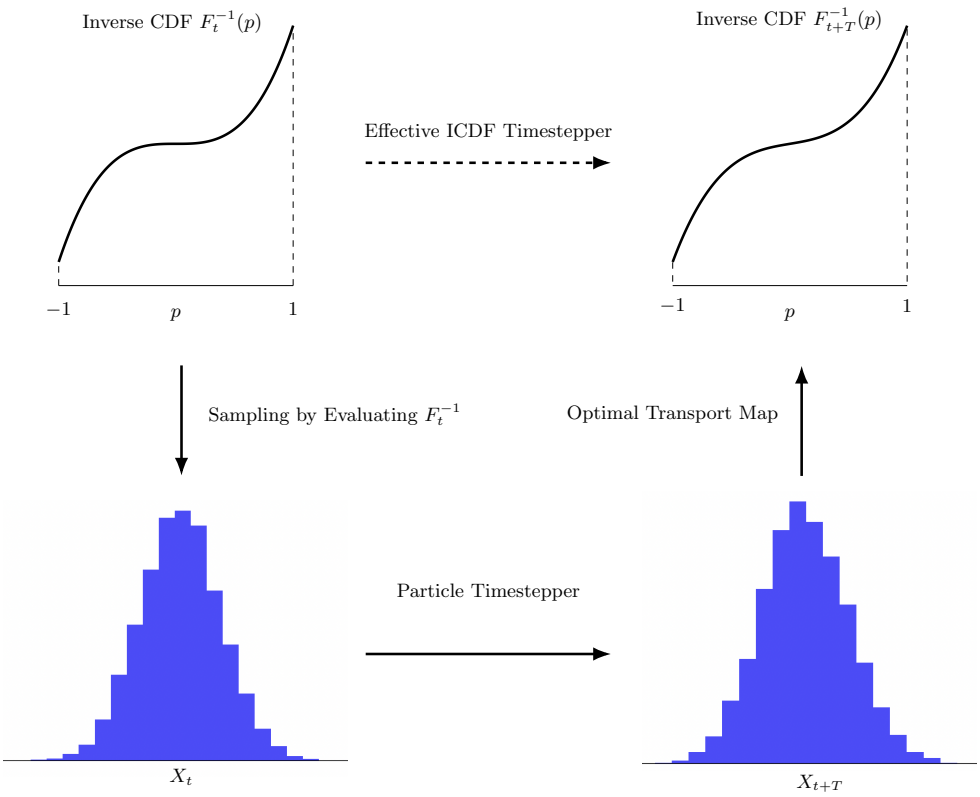


Figure 4.8: Schematic of the effective ICDF–to-ICDF timestepper.

which is zero at steady state. Since the ICDF provides a smooth aggregate representation of the ensemble, the noise is drastically reduced and Newton–Krylov succeeds in efficiently finding the steady state, as we demonstrate below.

4.4.2 Three Numerical Illustrations

We now demonstrate the performance of the Newton–Krylov method on the ICDF-to-ICDF timestepper through three examples. These examples illustrate the accuracy, robustness, and noise tolerance of Newton–Krylov on smooth particle representations across increasingly complex systems.

4.4.2.1 The Bimodal Distribution

As a first demonstration of the Newton–Krylov method applied to the ICDF-to-ICDF timestepper, we consider a bimodal distribution defined by

$$\mu(dx) = \frac{1}{Z} \exp\left(-\frac{1}{2}\left(x^2 - 1\right)^2\right). \tag{4.4.2.1}$$

The particle timestepper is based on an Euler-Maruyama discretization of the overdamped Langevin dynamics

$$dX_t = \nabla \log \mu(X_t)dt + \sqrt{2}dW_t \tag{4.4.2.2}$$

with step size 10^{-3}. The time-integration horizon is $h = 0.1$.

The unknowns in the ICDF-to-ICDF formulation are the values of the ICDF on a fixed percentile grid $p_k = (k - 0.5)/100$ for $1 \le k \le 100$. During each evaluation of the timestepper, we interpolate the ICDF using piecewise linear functions, sample $N = 10^5$ particles at equidistant percentiles, propagate them forward using the microscopic timestepper ϕ_h over a time window of length $h = 0.1$, and reconstruct the new ICDF by retaining every 1000th particle (i.e., $N/100$) in sorted order (i.e. after applying the optimal transport map). These retained particle locations define the updated ICDF on the fixed percentile grid at time h. Jacobian–vector products are approximated by finite differences with a step size of $\varepsilon = 10^{-2}$.

The convergence behavior of Newton–Krylov on the ICDF timestepper is shown in Figure 4.9. The left panel compares the ICDF of the initial guess—corresponding to a standard Gaussian—with the steady-state ICDF obtained after convergence. The right panel displays the histogram of $N = 10^5$ particles sampled from this steady-state ICDF, showing excellent agreement with the analytical bimodal invariant distribution. For completeness, we also report the residual norm $\left\|\Psi\left((F^{-1})^{(k)}\right)\right\|$ per Newton–Krylov iteration k, which decreases rapidly to a noise floor around 10^{-2}.

The Newton–Krylov method converges to the noise floor in 11 iterations, for a total

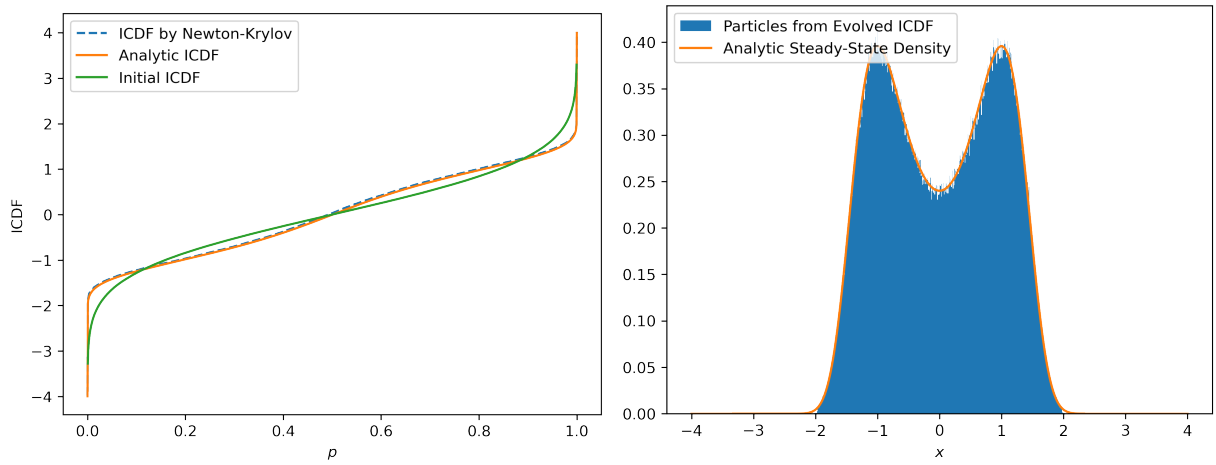


Figure 4.9: (Left) Initial ICDF (green), true steady-state (orange) and steady-state ICDF computed by Newton–Krylov (dashed blue). (Right) Particles sampled from the Newton–Krylov ICDF (blue) and corresponding steady-state bimodal density.

of 63 evaluations of the ICDF timestepper (GMRES needs a few function evaluations per global Newton iteration). This technically corresponds to 6.3 in-simulation seconds of propagating particles to reach the steady state distribution - *much less than the hundreds of seconds that would be required to reach steady state using the Euler-OT timestepper*. The total in-simulation time is the most important metric for comparing optimizers.

4.4.2.2 Bacterial Chemotaxis

As a second example, we revisit the bacterial chemotaxis model introduced in section 4.2.4.1. The ICDF is again discretized on a fixed percentile grid $p_k = (k - 0.5)/100$, for $1 \leq k \leq 100$. At each timestepper evaluation, we interpolate the ICDF using piecewise linear functions, sample $N = 10^5$ particles at equidistant percentiles, and propagate them over a time horizon of $h = 1$ second using the microscopic particle timestepper described in section 4.2.4.1. The propagated ensemble is then projected back onto the ICDF representation by sorting the particle locations. The initial distribution is again taken to be a Gaussian with mean 5 and standard deviation 2, and Jacobian–vector products are again evaluated using finite differences with step size $\varepsilon = 10^{-1}$.

Figure 4.10 illustrates the convergence of the Newton–Krylov method on this

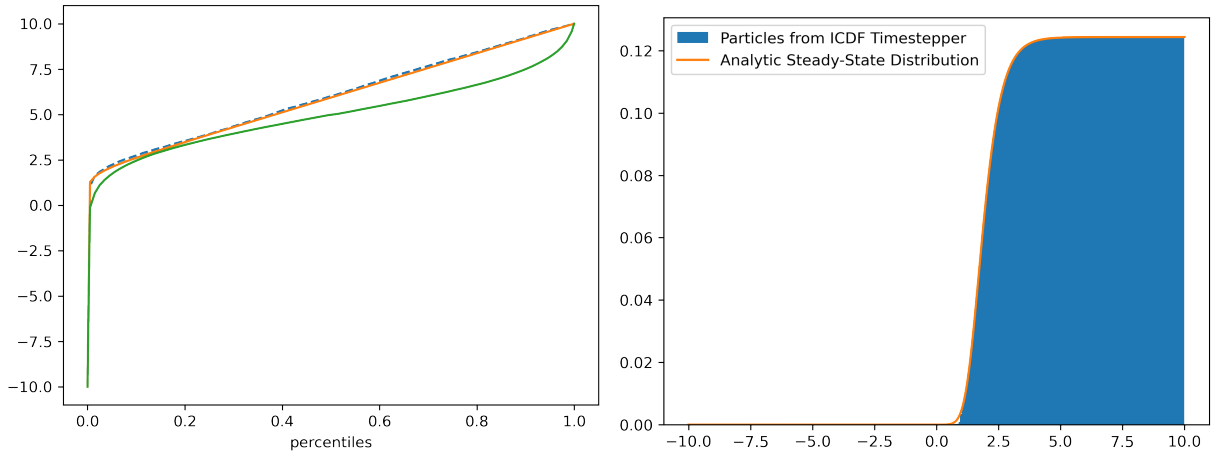


Figure 4.10: (Left) Initial ICDF(green), true steady-state (orange) and steady-state ICDF computed by Newton–Krylov (dashed blue). (Right) Particles sampled from the Newton–Krylov ICDF (blue) and corresponding steady-state density (4.3.1.4) in orange.

ICDF-to-ICDF timestepper. The left panel shows the initial ICDF (corresponding to the Gaussian guess), the true steady-state ICDF obtained from long-time simulation, and the ICDF recovered by Newton–Krylov. The latter two curves show close agreement, confirming that the method correctly identifies the stationary distribution. The right panel displays the histogram of $N = 10^5$ particles sampled from the steady-state ICDF computed by Newton–Krylov, which closely matches the invariant chemotactic density. The Newton–Krylov method is able to reach this steady-state profile after 11 Newton iterations and 114 objective evaluations, corresponding to 114 seconds of in-simulation time ($h = 1$ second). Regular simulation using the Euler-OT timestepper requires about 300 seconds to reach steady state from the same initial distribution.

4.4.2.3 Economic Agents

As a final example, we apply the ICDF-to-ICDF timestepper framework to the economic agents model introduced in section 4.2.4.2. The microscopic timestepper ϕ_h for this system is the McKean–Vlasov Euler scheme described in [FEC+24]. This model is nonlinear and non-local, and its corresponding mean-field equation (4.3.1.6) is known to admit multiple steady-state distributions.

The construction of the ICDF-to-ICDF timestepper is identical to the prior two examples. That is, we discretize the ICDF on a fixed percentile grid $p_k = (k - 0.5)/100$, for $1 \leq k \leq 100$ and $N = 10^5$ particles are sampled uniformly in percentile space. The microscopic timestepper ϕ_h is then applied to evolve these particles over a time horizon of $h = 1$ second. After propagation, the updated ICDF is reconstructed by sorting the particle ensemble retaining every 1000th particle. The initial ICDF corresponds to a Gaussian distribution centered around $x = 0$ with standard deviation 0.1. Jacobian–vector products of the timestepper are again computed using finite differences with step size $\varepsilon = 10^{-1}$.

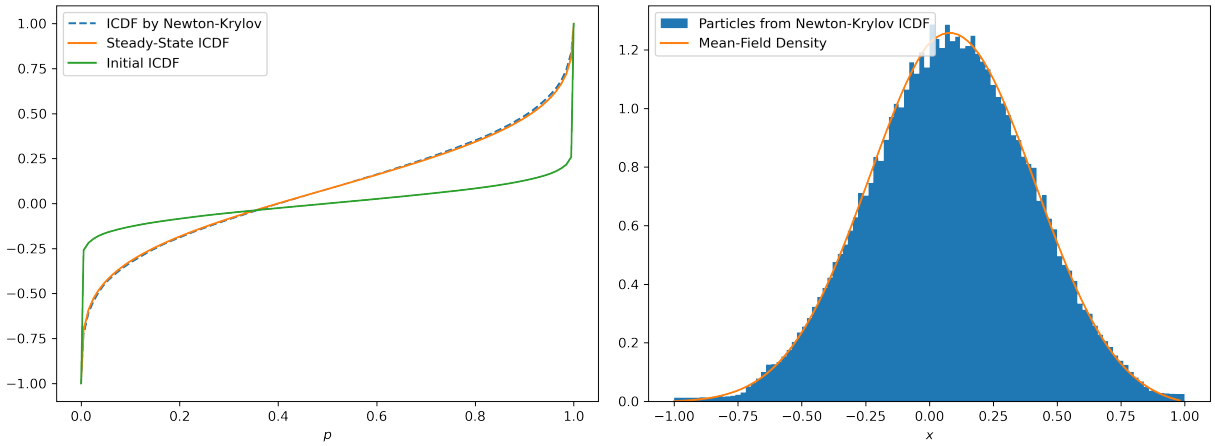


Figure 4.11: (Left) Initial ICDF (green), true steady-state (orange) and steady-state ICDF computed by Newton–Krylov (dashed blue). (Right) Particles sampled from the Newton–Krylov ICDF (blue) and steady-state density corresponding to the mean-field PDE.

Figure 4.11 illustrates the results obtained with Newton–Krylov on this ICDF timestepper. The left panel shows the initial ICDF, the invariant ICDF obtained from long-time microscopic simulation, and the steady-state ICDF recovered by Newton–Krylov. The close agreement between the two confirms that the algorithm correctly identifies the stable steady-state distributions of the system. The right panel displays the histogram of $N = 10^5$ particles sampled from the Newton–Krylov steady-state CDF, which closely matches the invariant distribution of the mean-field PDE (4.3.1.6). For completeness we also show the Newton–Krylov error as a function of the iteration number in Figure 4.12. We see that the error first decreases quadratically as is typical for Newton-like schemes,

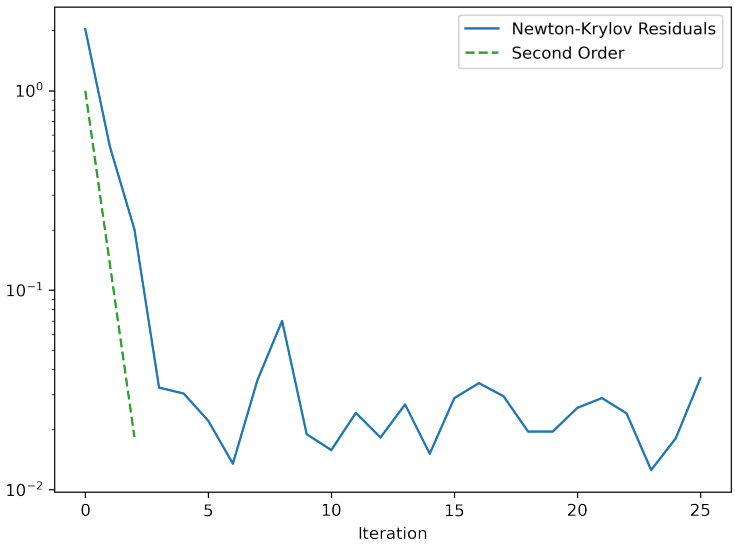


Figure 4.12: Newton–Krylov ICDF residual $\Psi\left((F^{-1})^{(k)}\right)$ per iteration for the economics agents model (blue); Second-order convergence rate (green).

before settling down in the noise regime.

4.5 Extending Smooth Representations to Multiple Dimensions

In one dimension, the ICDF approach allows us to handle the issue presented by noisy particles and still develop an appropriate Newton–Krylov method. Extending this idea beyond one dimension requires additional care. There is not an analog of the ICDF that appropriately captures the optimal transport maps. In particular, an inverse cumulative distribution function cannot be defined in a straightforward manner for $d \geq 2$. The reason is that the CDF is not injective: for a given probability level $p \in [0, 1]$, the level set

$$\left\{(x, y) \in \mathbb{R}^2 \mid F_{X,Y}(x, y) = p\right\} \tag{4.5.0.1}$$

typically forms a one-dimensional manifold (a curve) rather than a single point. Hence, the mapping $(F_{X,Y})^{-1}(p)$ is not a well-defined function from $[0, 1] \to \mathbb{R}^2$. Consequently, the

notion of "evaluating the ICDF" becomes ambiguous in higher dimensions. It follows that the one-dimensional ICDF-based timestepper illustrated in Figure 4.8 cannot be extended to multivariate distributions in a direct manner. Even if a generalized notion of inverse CDF were introduced (e.g., by parameterizing level sets), its numerical evaluation would be computationally inefficient, since sampling points uniformly on such two-dimensional level sets is considerably more expensive than evaluating the scalar inverse $F^{-1}(p)$ in one dimension.

One must instead construct alternative smooth representations that retain the essential link to Optimal Transport theory. An option is to work with the two-dimensional cumulative distribution function (CDF), which generalizes the one-dimensional case and retains links to optimal transport. An alternative is the sliced Wasserstein distance. By projecting the distribution onto many one-dimensional directions, each projection yields an ICDF that is well defined, and these are aggregated to approximate the full Wasserstein geometry. Therefore, both the CDF and the sliced Wasserstein framework act as higher-dimensional analogues of the one-dimensional ICDF, providing the smoothness needed to construct well-behaved timesteppers and to restore the accelerated convergence of Newton–Krylov methods in the multidimensional setting.

It is worth emphasizing, though, that the reason we need an alternative approach is purely for computational feasibility. If we had an oracle that could instantaneously compute the optimal transport map between any two distributions of arbitrary size, our original approach would still work in multiple dimensions. Indeed, if we were able to compute the associated vector field in the case of continuous distributions, then there would not be any noise, and we could run the Newton–Krylov method exactly as expected. However, due to the noisy nature of the timestepper, our Newton–Krylov approach needs more particles in the simulation than would be reasonable to pass to an explicit OT solver.

This computational issue only appears in the setting of multiple ambient dimensions precisely because optimal transport maps are extremely cheap to compute in one dimension.

For multiple dimensions, computing an optimal transport map requires solving a linear program which is no longer practical for the number of particles we would need in order for the desired effects to become observable. As we have discussed, the signal-to-noise ratio in the computation of the Jacobians is extremely poor for small numbers of particles, and in order to make the ratio high enough for our Newton–Krylov method to work, we need to make the number of particles so high that solving the linear program becomes computationally infeasible. Additionally, the statistical rate of convergence of discrete optimal transport maps depends poorly on the dimension (typically $n^{-1/d}$, where n is the number of points and d is the ambient dimension), meaning the number of particles needed also scales with the dimension. This creates a bad tradeoff with computational complexity that cannot be resolved without unreasonable computational power. Thus, the reason we need additional alternative approaches in multiple dimensions is the practical computational ability, not some fundamental flaw with the approach of running Newton–Krylov on the OT velocity fields.

We discuss two such alternatives in detail here: the multidimensional cumulative distribution function and the sliced-Wasserstein distance. We discuss the former idea in section 4.5.1 and the latter in section 4.5.2.

4.5.1 The CDF-to-CDF Timestepper

A natural choice is to work directly with the two-dimensional cumulative distribution function $F_{X,Y}(x, y)$, which provides a continuous and differentiable description of the underlying particle ensemble. The notion of a cumulative distribution function (CDF) extends naturally to multiple dimensions. For clarity of exposition, we focus on the bivariate case, although all subsequent algorithms and results generalize to arbitrary number of dimensions. Let (X, Y) be a random vector with joint distribution. Its cumulative

distribution function is defined by

$$F_{X,Y}(x,y) = P(X \leq x, Y \leq y) = \int_{-\infty}^{x}\int_{-\infty}^{y} \mu(u,v)dudv. \tag{4.5.1.1}$$

Sampling from a two-dimensional cumulative distribution function can be performed through its marginal and conditional components. Following the approach described by [ZKG05, ZKG06], the joint CDF $F_{X,Y}(x,y)$ can be decomposed into the marginal CDF $F_X(x)$ of one variable and the conditional CDF $F_{Y|x}(y|x)$ of the other. This decomposition allows sampling to proceed through a sequence of one-dimensional operations: one first draws a sample x from the marginal cumulative distribution $F_X(x)$, and subsequently samples y from the conditional cumulative distribution $F_{Y|X}(y)$. In this way, multidimensional sampling reduces to iterated evaluations of smooth one-dimensional ICDFs, which are mathematically well-defined.

The two-dimensional sampling algorithm works as follows. First we construct the marginal CDF of the x-coordinates, given by

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x}\int_{-\infty}^{\infty} \mu(u,v)dudv = F_{X,Y}(x,\infty). \tag{4.5.1.2}$$

This one-dimensional CDF can be easily inverted to generate samples $x_1, x_2, \ldots, x_n$ at equidistant percentiles $p_i = (i-0.5)/n$. Next, for each x-sample x_i we construct the one-dimensional CDF of the conditional distribution $\mu(y|x_i)$. It can be shown [Pap65, Bil95] that this conditional cumulative distribution function is given by the formula

$$F_{Y|X_i}(y) = \frac{\partial_x F_{X,Y}(X_i, Y)}{\mu_X(X_i)}. \tag{4.5.1.3}$$

Given any smooth representation of $F_{X,Y}(x,y)$ this partial derivative is readily available, and sampling the y-coordinates X_i can also be achieved by inverting $F_{Y|X_i}(y)$ and evaluating it in equidistant percentiles $q_j = (j-0.5)/m$. The end product of this staged sampling algorithm

are N particles $(X_n, Y_n)_{n=1}^N$. After propagating the particles through the timestepper

$$(\tilde{X}_n, \tilde{Y}_n) = \phi_h(X_n, Y_n), \tag{4.5.1.4}$$

we reconstruct the updated two-dimensional CDF by computing, at each grid node (x_i, y_j), the fraction of particles $(\tilde{X}_n, \tilde{Y}_n)_{n=1}^N$ located in the lower-left quadrant relative to that point:

$$F_{X,Y}(x_i, y_j) = \#\left\{n \mid \tilde{X}_n \leq x_i \text{ and } \tilde{Y}_n \leq y_j\right\}/N \tag{4.5.1.5}$$

Starting from the CDF $F_{X,Y}^t$ at time t, the three steps

1. Sampling (X_n, Y_n) from marginal and conditional CDFs;
2. Forward particle propagation using the particle timestepper $(\tilde{X}_n, \tilde{Y}_n) = \phi_h(X_n, Y_n)$;
3. Restriction to the 2D CDF by counting particles in the lower right quadrant of each grid point;

define a consistent CDF-to-CDF timestepper

$$\Phi_h\left(F_{X,Y}^t\right) = F_{X,Y}^{t+h}. \tag{4.5.1.6}$$

See Figure 4.13 for a schematic of this timestepper. The associated residual map reads

$$\Psi\left(F_{X,Y}^t\right) = F_{X,Y}^t - \Phi_h(F_{X,Y}^t), \tag{4.5.1.7}$$

which is identical to 0 at every grid point (x_i, y_j) in steady state.

We note that constructing the two-dimensional CDF of a particle ensemble and re-sampling it through the marginal and conditional one-dimensional CDFs, as we outlined above, yields a monotone triangular map that is close to the OT solution under many practical conditions. In fact, it corresponds to the Knothe–Rosenblatt rearrangement [Ros52,

Kno57, San15, PC19], a sequential, measure-preserving transformation that orders variables one at a time through their conditional distributions. While the Knothe-Rosenblatt map does not minimize the standard quadratic OT cost (and hence is not itself an optimal transport map in the usual sense), it can be expressed as a limit of optimal transport maps for different cost functions [CGS10]. In particular, it shares key structural properties and can be viewed as a computationally efficient approximation to the optimal transport map in high-dimensional sampling contexts. We do not require the exact optimal transport map in our Newton-Krylov calculations; any fast and reasonably accurate alternative that preserves the steady state is sufficient. The Knothe-Rosenblatt rearrangement offers a good alternative that is computationally efficient and preserves the steady state.

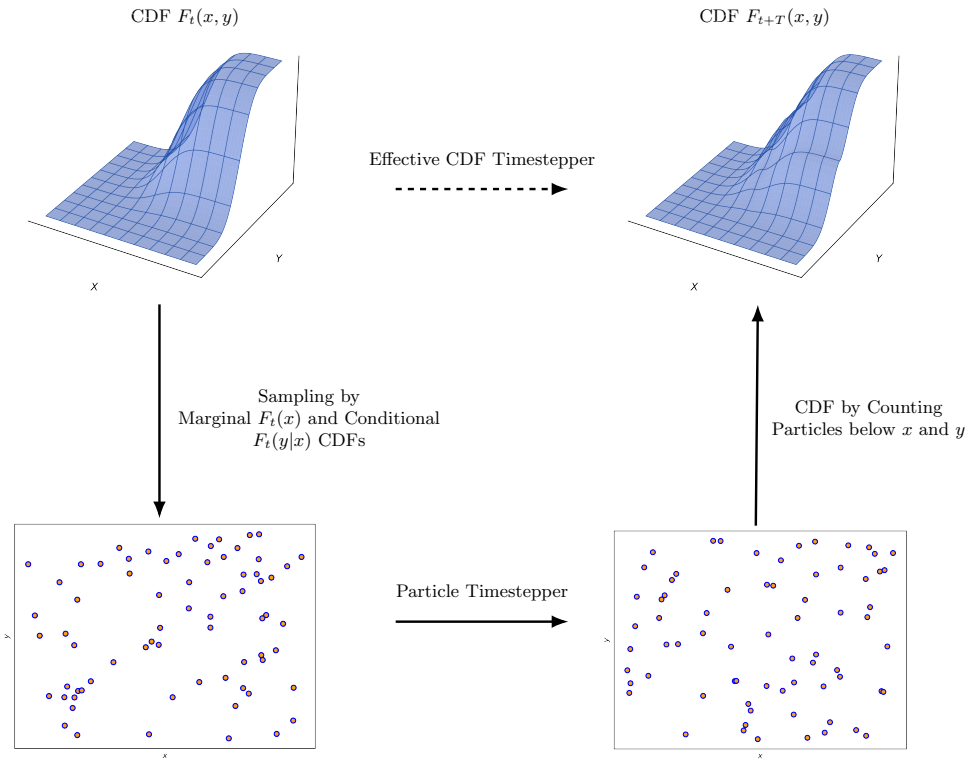


Figure 4.13: Schematic of the effective two-dimensional CDF-to-CDF timestepper.

Numerical Example Consider the two-dimensional half-moon distribution (4.2.4.2) as a representative example of our approach. We evaluate the empirical two-dimensional

cumulative distribution function (CDF) on a uniform grid $(x_i, y_j)_{i,j}$ of $n_X = n_Y = 100$ points, equally spaced between -4 and 4. The CDF-to-CDF timestepper is constructed using $N = 10^4$ particles, and sampling is performed by inverting the marginal and conditional one-dimensional cumulative distribution functions. To increase regularity, we interpolate the two-dimensional CDF with a piecewise-linear spline and solve for the corresponding percentiles. Specifically, for every percentile pair (p_i, q_j), we determine the coordinates $(X_i, Y_j|X_i)$ such that

$$F_X(X_i) = p_i,\ F_{Y|X_i}(Y_j) = q_j. \tag{4.5.1.8}$$

This inversion can be implemented efficiently by vectorizing the evaluation of the marginal and conditional CDFs. Because the two-dimensional CDF is represented as a smooth spline, its partial derivatives—such as those appearing in (4.5.1.3)—are spline functions as well and need to be computed only once, independent of p_i, q_j, X_i or $Y_j|X_i$.

We next apply the Newton–Krylov scheme to find the steady-state distribution of the CDF-to-CDF timestepper. The timestepper is based on an Euler-Maruyama discretization of the dynamics (4.2.4.4) with time step 10^{-3}. We integrate this dynamics up to time $h = 1$ second. The initial condition for the Newton–Krylov method is the standard bivariate Gaussian distribution (see the left of Figure 4.14). The optimized distribution obtained by Newton–Krylov is displayed on the right of Figure 4.14. One can see that Newton–Krylov can recover the true steady-state distribution even though the initial distribution is quite far from equilibrium. Figure 4.15 shows how the 'loss' $\|\Psi(F_{X,Y})\|$ decreases steadily per nonlinear iteration and settles down to the noise level, a clear signal of convergence.

4.5.2 The Sliced Wasserstein Timestepper

For completeness, we also consider an alternative smooth representation of multidimensional particle systems based on the sliced Wasserstein distance (SWD). The key idea is to project probability distributions in $\mathbb{R}^d$ onto one-dimensional subspaces defined by

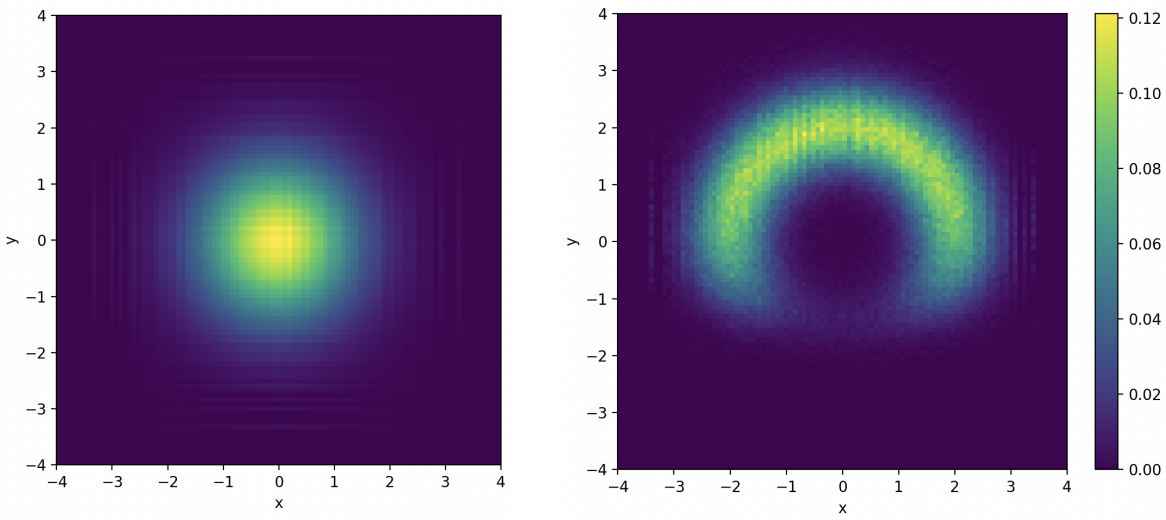


Figure 4.14: Histogram heatmaps of particle density: (left) initial Gaussian condition; (right) distribution after Newton–Krylov optimization using the 2D CDF smooth representation.

unit vectors $\theta \in \mathbb{S}^{d-1}$ at percentiles of the angular CDF, and to compute the Wasserstein distance between the resulting one-dimensional projected measures. Formally, for two distributions μ and ν,

$$SW_d^2(\mu, \nu) = \int_{\mathbb{S}^{d-1}} W_{d-1}^2 \left(P_\theta \# \mu, P_\theta \# \nu\right) \, \mathrm{d}\theta, \tag{4.5.2.1}$$

where $P_\theta(x) = \langle x, \theta \rangle$ is the projection onto the line spanned by θ and $P_\theta \# \mu$ denotes the pushforward of μ by P_θ. In two dimensions, we approximate the integral with a finite set of angles $\{\theta_i\}_{i=1}^{N_\theta}$:

$$SW_2^2(\mu, \nu) \approx \frac{1}{N_\theta} \sum_{i=1}^{N_\theta} W_2^2\big(P_{\theta_i} \# \mu, \; P_{\theta_i} \# \nu\big) = \frac{1}{N_\theta} \sum_{i=1}^{N_\theta} \int \left| F^{-1}_{P_{\theta_i} \mu}(r) - F^{-1}_{P_{\theta_i} \nu}(r) \right|^2 \mathrm{d}r, \tag{4.5.2.2}$$

where $F_{P_{\theta_i} \mu}$ is the one-dimensional CDF of the projected measure $P_{\theta_i} \# \mu$. We note that the sliced Wasserstein distance lower bounds the ordinary Wasserstein distance, and it also gives an upper bound up to a constant that depends on the ambient dimension [BRPP15]. Thus, the sliced Wasserstein distance is indeed a reasonable approximation, preservers the steady state, and we can still use it effectively build a similar timestepper.

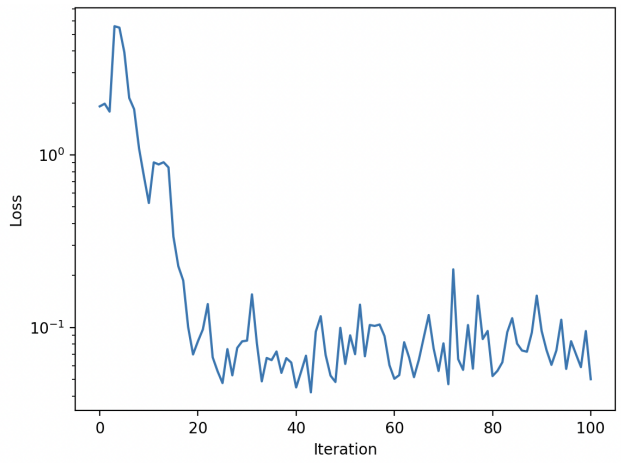


Figure 4.15: Newton–Krylov loss $\left\|\Psi\left(F^{(k)}_{X,Y}\right)\right\|$ per iteration k. The loss decreases up until a noise level induced by the finite number of particles.

The sliced Wasserstein idea provides an alternative to build smooth timesteppers from an underlying particle timestepper. We retain a rectangular grid representation of the two-dimensional CDF, but sampling proceeds through directional projections. To achieve a consistent sampling, we need to first sample directions θ_i from the marginal angular CDF F_Θ and, for each θ_i, generate radii $r_j|\theta_i$ from the conditional CDF $F_{R|\theta_i}(r)$. However, unlike the Cartesian marginal and conditional CDFs, there are no explicit expressions analogous to equations (4.5.1.2) and (4.5.1.3) for the marginal angular and conditional radial CDFs. We need to first explicitly construct the two-dimensional density

$$\mu(x, y) = \nabla F_{X,Y}(x, y) \tag{4.5.2.3}$$

which can be obtained efficiently from a spline representation of $F_{X,Y}$. Then the marginal angular CDF is given by

$$F_\Theta(\theta) = \int_0^\theta \int_0^\infty \mu\left(r\cos\phi,\, r\sin\phi\right)\, r\, dr\, d\phi. \tag{4.5.2.4}$$

and the conditional radial CDF for any angle θ reads

$$F_{R|\theta}(r) = \int_0^r \mu(s\cos\theta, s\sin\theta)\; s\; ds. \tag{4.5.2.5}$$

Sampling the 2D CDF consistently in a sliced Wasserstein-inspired way can then be achieved through

1. Generating a set of angles $\{\theta_i\}_{i=1}^{N_\Theta} \subset [0, 2\pi)$ by inverting the marginal angular CDF (4.5.2.4) in fixed percentiles $p_i = (i - 0.5)/N_\Theta$;

2. For each θ_i, constructing the radial conditional CDF $F_{R|\theta_i}(r)$ using the projection (4.5.2.5);

3. Inverting each radial conditional CDF in fixed percentiles $\{r_j\}_{j=1}^{N_r} \subset (0, 1)$ through a bisection-like scheme

$$\rho_{ij} \;=\; F^{-1}_{R|\theta_i}(r_j).$$

;

4. Mapping these 1D samples back to $\mathbb{R}^2$ via the inverse projection

$$X_n = \rho_{ij}\cos\theta_i, \qquad Y_n = \rho_{ij}\sin\theta_i,$$

and collecting all (X_n, Y_n) across i, j.

After sampling we proceed as with the CDF-to-CDF timestepper by propagating the samples through the particle timestepper to obtain new particles $(\tilde{X}_n, \tilde{Y}_n)$ and restricting back to the CDF by counting the number of $(\tilde{X}_n, \tilde{Y}_n)$ to the lower left of each CDF grid point (x_i, y_j). Together, this sliced representation captures much of the transport geometry while remaining computationally light, providing a scalable surrogate for the full multidimensional Wasserstein map within our Newton–Krylov framework. The complete sliced Wasserstein-to-sliced Wasserstein timestepper is shown schematically in Figure 4.16.

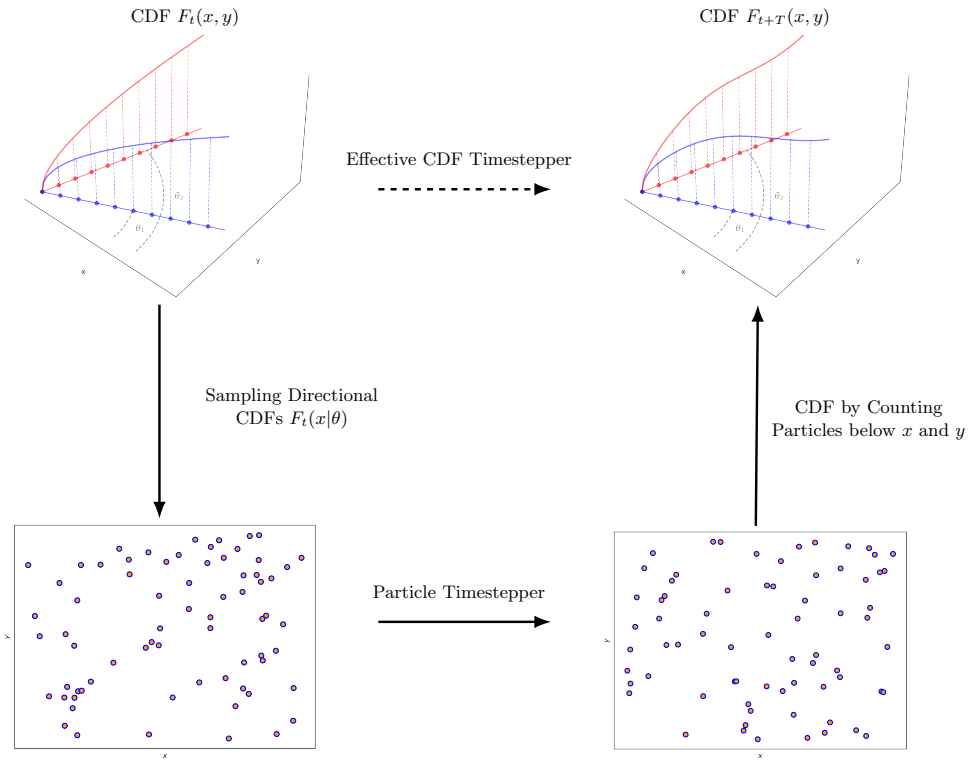


Figure 4.16: Schematic of the effective sliced Wasserstein to sliced Wasserstein timestepper.

Finally we also show how the Newton–Krylov method on the sliced-Wasserstein representation of particles converges to the steady-state distribution of the half-moon potential. The initial condition and timestepper parameters are the same as in section 4.5.1, and sampling is done by first generating $N_\Theta = 100$ angular samples and then $N_R = 100$ radial samples for each angle. The Newton–Krylov loss per iteration is shown in the left subfigure of Figure 4.17 and the resulting steady-state distribution is shown on the right.

As shown, there is no significant difference in either the convergence rate or the resulting steady state between the sliced Wasserstein and the direct CDF representations. This outcome is expected, since both formulations are mathematically equivalent. The key insight is that employing smooth representations enables the use of higher-order optimization schemes, while the specific choice of representation is comparatively unimportant. The decisive factor is the computational efficiency of sampling.

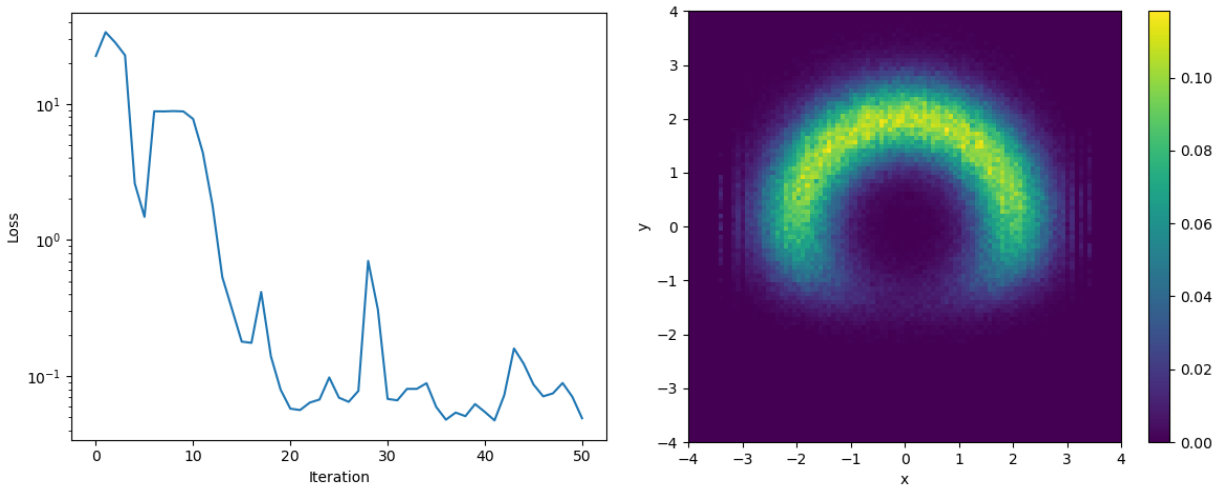


Figure 4.17: (left) Loss $\left\|\Psi(F^{(k)}_{X,Y}\right\|$ per iteration; (right) Histogram heatmap of resulting steady-state particle density of the Newton–Krylov method applied to the Sliced Wasserstein representation.

4.6 Discussion and Outlook

We have presented a unified, matrix-free framework for computing steady-state distributions of (stochastic) particle timesteppers. The key idea is to reformulate a steady state in the language of optimal transport. On the deterministic side, we revisited the residual formulation $\psi(u) = u - \phi_h(u)$ and showed how the Newton–Krylov method efficiently recovers steady states of the Fokker-Planck equation. On the stochastic side, we introduced a first-order Adam-Wasserstein method for calculating steady states directly on the particle level. We also demonstrated that a naive particle-level extension fails to higher-order optimizers such as Newton–Krylov.

To address this limitation, we introduced smooth distributional timesteppers—first in one dimension through the ICDF-to-ICDF map, and then in multiple dimensions through CDF-to-CDF and sliced Wasserstein formulations. These representations aggregate microscopic variability into smooth macroscopic objects on which Newton–Krylov regains its fast, second-order convergence. Numerical results confirm that these smooth timesteppers yield comparable steady states with markedly reduced stochastic fluctuations, enabling

accurate steady-state computations even in noisy particle systems. The central message is that smoothness in representation, rather than in the underlying dynamics, is the key to robust, matrix-free solvers for stochastic steady states.

One of the main questions we want to address during further research will be to reduce the stochastic error in the finite-differences approximation of the Jacobian. Variance reduction at the particle level through correlated samples could further stabilize Jacobian-vector products and allow smaller finite-difference steps. In the case of central finite differences, antithetic variates might be a natural variance reduction technique since paired perturbations with opposite randomness can cancel leading-order stochastic fluctuations while preserving the deterministic directional derivative signal.

Beyond computing single steady states, an important next step is to apply Newton–Krylov to compute steady-state branches of parameter-dependent particle systems. Such numerical continuation algorithms require consistent residual evaluations between successive Newton-Krylov steps. Therefore, improving variance reduction at the particle level—through correlated sampling, common-random-number strategies, or smoother estimators will be crucial to enable robust and efficient continuation of stochastic steady states, including the reliable detection of folds and bifurcations in distribution space.

Finally, scaling these approaches to large particle ensembles and to systems with potentially thousands of dimensions remains a major challenge. A key open question is how to identify the most effective sampling strategy for multidimensional CDFs (or their smooth equivalents) constructed via percentile evaluations. What constitutes "best" in this context is not yet defined, but it will likely involve a balance between proximity to the true optimal transport map (much like the Knothe-Rosenblatt rearrangement approximates it) and the computational efficiency of the resulting sampling algorithm. Establishing this balance will be essential for extending smooth timestepper frameworks to high-dimensional stochastic systems like molecular dynamics [CDGK04] or real economic systems.

Acknowledgements

Chapter 4, in full, is a reprint of material in [VKCK25]: Hannes Vandecasteele, Nicholas Karris, Alexander Cloninger, and Ioannis G. Kevrekidis. Optimal transport, timesteppers, Newton-Krylov methods and steady states of collective particle dynamics. Preprint, arXiv:2512.24567, 2025. The dissertation author is a primary author and investigator of this paper, which is currently under revision for publication in the Journal of Computational Physics.

Chapter 5

Image Interpolation Using Wasserstein Geodesics

It can be shown that Stable Diffusion has a permutation-invariance property with respect to the rows of Contrastive Language-Image Pretraining (CLIP) embedding matrices. This inspired the novel observation that these embeddings can naturally be interpreted as point clouds in a Wasserstein space rather than as matrices in a Euclidean space. This perspective opens up new possibilities for understanding the geometry of embedding space. For example, when interpolating between embeddings of two distinct prompts, we propose reframing the interpolation problem as an optimal transport problem. By solving this optimal transport problem, we compute a shortest path (or geodesic) between embeddings that captures a more natural and geometrically smooth transition through the embedding space. This results in smoother and more coherent intermediate (interpolated) images when rendered by the Stable Diffusion generative model. We conduct experiments to investigate this effect, comparing the quality of interpolated images produced using optimal transport to those generated by other standard interpolation methods. The novel optimal transport–based approach presented indeed gives smoother image interpolations, suggesting that viewing the embeddings as point clouds (rather than as matrices) better

reflects and leverages the geometry of the embedding space.

5.1 Introduction

The study of the manipulation and interpolation of the inputs to and outputs of generative diffusion-based text-to-image models like Stable Diffusion and latent diffusion models [RBL+22, EKB+24] has attracted increased attention in recent years. These models produce novel images by denoising a random noise latent conditioned on a prompt input [HJA20]. The prompt embeddings obtained from the raw prompt text, such as those obtained by CLIP [RKH+21], are the input observed by the model and used for training and sampling. Understanding these prompt embedding spaces is desirable for a variety of creative and functional applications, including prompt optimization [GAA+23, WLHG24, ZYHT07], improved sampling diversity [DPP24], image and prompt inversion [ZWW+24, LZW+25], concept ablation and unlearning [LvdWH+24, KZW+23], and image interpolation [WG23].

Specifically, interest in image interpolation in the context of diffusion models has grown. Applications include anticipated domains such as video transitions [ZCL+25], but also extend to a variety of unexpected use-cases, such as improving echocardiography image quality [SNB+24] or classifying plant health [Lee25]. Methods to obtain smooth image interpolations in the context of generative diffusion models include latent-space interpolation [YSY+25, SM25] a mixture of latent and prompt embedding interpolation [WG23, YYX+24, ZZX+24], and even attention interpolation [HWLY24]. Most of these approaches (excepting [HWLY24]) focus on starting with real images, inverting to the embedding space, interpolating between embeddings, and then generating images for the interpolated embeddings.

In parallel, there has been a growing interest in the geometry of learned latent spaces for frontier models like the work of [BHB25, Lee23, AHH18, SST+24], and even for

movement and interpolation between images in [PKC+23]. Geometric perspectives are being leveraged extensively in the the interpretability literature [VB20, BB20]. Additionally, there is research in the broader community looking at how geometric properties are connected to topics like hallucination [YBS+25, PWM+25] and deep-fake [XHFS25, BSC+23, SS24] detection. Much of this work focuses on clustering, structure, and dominant modes or directions within the learned embedding space where the representations of the data are vectors or matrices.

Rather than focus on the geometry of *latent* space as in, e.g., [PKC+23], we look to connect geometric perspectives on the textual *embedding* space with the image interpolation task in Stable Diffusion. This novel work leverages a permutation invariance property of the embedding-to-image generator used in Stable Diffusion to produce a new way of exploiting the geometry of embeddings in image interpolation. Rather than understanding relationships between prompts based on Euclidean or matrix-based relationships, we consider learned embeddings as unordered point-clouds. This point-cloud provides a different way to interpolate between embeddings using optimal transport. While optimal transport techniques have been incorporated into diffusion frameworks for concept optimization [LWLL25], as well as interpolation methods [ZYHT07, YYX+24], to the best of our knowledge, no studies explore the effect of using optimal transport to pair prompt embedding tokens for interpolation within a diffusion framework.

In our work, we compare the performance gained by using optimal transport for interpolation between prompt embeddings from a point-cloud view. Although there is evidence that interpolating only between prompt embeddings results in sub-par interpolations [HWLY24], by fixing the latent seed and focusing only on prompt embedding interpolation, we are able to isolate properties and features of the prompt embedding space which otherwise might be masked by interpolating the latents and the prompts simultaneously. Beyond simply improving interpolation methods, this research provides a potential mechanism to improve foundational understanding of the prompt embedding

space and emergent behavior.

In Section 5.2 we present the diffusion and embedding models used, underlying cross attention permutation invariance assumptions required for the application of optimal transport, and the optimal transport methodology. Our novel approach and experimental hypotheses are shared in Section 5.3. Section 5.4 establishes our experimental design and presents our results of probing the effect of optimal transport for embedding interpolation. A discussion of the results and potential future work are provided in Section 5.5.

5.2 Preliminaries and theoretical motivation

In this section we present a high-level overview of relevant concepts. We begin with diffusion models, stable diffusion, and Contrastive Language-Image Pretraining (CLIP) in Section 5.2.1 followed by discussions of CLIP permutation invariance and optimal transport in Sections 5.2.2 and 5.2.3, respectively.

5.2.1 Diffusion models, stable diffusion, and CLIP

Diffusion models are a class of probabilistic generative models that have garnered significant attention for their ability to produce high-quality images that closely adhere to user-specified criteria, unlocking applications that range from automating artistic content creation to aiding researchers in synthetic data generation, design and modeling complex systems and processes [CSJ+23, MA22]. At their core, diffusion models operate through two complementary processes: (*i*) a forward "diffusion" process where an original image is corrupted through the iterative addition of Gaussian noise, and (*ii*) a reverse-diffusion process that reconstructs the original image by estimating and removing the noise added at each diffusion step. More formally, in the diffusion process, the data $\mathbf{x}_0$ are perturbed at

each step t through the Markov chain

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}),$$

where $\beta_t \in (0,1)$ is a variance schedule determining the amount of noise added at step t, and $\mathcal{N}(\cdot; \mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. The reverse-diffusion process to recover $\mathbf{x}_0$ through $\mathbf{x}_T$ is modeled by a separate Markov chain

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(t)),$$

where $\mu_\theta(\mathbf{x}_t, t)$ is the predicted mean (typically learned using a neural network (U-Net)), and $\Sigma_\theta(t)$ is the variance (often fixed, but can be learned).

While several variations of diffusion models have been developed [CTG+23, HJA20, DN21, RBL+22, SSDK+21, SME21], our interest lies in the widely studied *latent* diffusion model Stable Diffusion 1.5 (SD) by Stability AI [RBL+22]. Like traditional diffusion models, SD employs a forward and reverse diffusion process, but for SD, these processes operate in a latent space rather than the original data space. A pre-trained variational autoencoder (VAE) first maps input data $\mathbf{x}_0$ to a lower-dimensional latent representation $\mathbf{z}_0$ via an encoder E, which substantially improves computational efficiency of the diffusion/reverse-diffusion processes since the latent space is much lower dimensional than that of the original data. The reconstructed representation obtained through the latent reverse-diffusion process is then decoded back into the data space through the VAE decoder D.

The U-Net architecture of the reverse-diffusion process in SD differs from traditional diffusion in that it has been enhanced with attention mechanisms, and in particular cross-attention between the latent space and a textual embedding space. SD uses the Contrastive Language-Image Pretraining (CLIP) model developed by OpenAI [RKH+21], which maps images and text strings to a shared embedding space in such a way that paired texts and images are nearby and unrelated pairs are farther apart. Thus, the Stable Diffusion 1.5

pipeline for producing an image corresponding to a particular prompt string is to compute a CLIP embedding of the string (padded if necessary), which yields a collection of 77 tokens vectors in $\mathbb{R}^{768}$, canonically packaged as a matrix in $\mathbb{R}^{77\times 768}$. This is the textual embedding space that is used in the cross-attention blocks in the U-Net architecture for image generation.

In summary, the Stable Diffusion pipeline has two essential "functions" around which we will center our focus: (*i*) the textual prompt embedding via CLIP and (*ii*) a reverse-diffusion processes wherein embeddings are denoised and decoded into an image influenced by the CLIP-encoded text. These two functions can be described as

- $f : \{\text{strings}\} \to \mathbb{R}^{77\times 768}$, where $s \mapsto f(s)$, the CLIP embedding of the prompt s,
- $g : \mathbb{R}^{77\times 768} \to \{\text{images}\}$, where $e \mapsto g(e)$, the image generated from embedding e

Note that the image generation function g is hiding a tremendous amount of computational complexity. In particular, it contains the entire U-Net and cross-attention architecture that underlies the "denoising" process, which involves many more initial parameters than just the CLIP embedding matrix. Throughout this chapter, we will consider all other parameters of this architecture fixed (e.g., the initial "noisy" latent z_T), meaning we are only interested in how the image that results from the overall pipeline (i.e., the image that results from decoding the final denoised latent) changes as a function of the CLIP embedding input. Thus, while the underlying architecture is extremely intricate, it is helpful to consider it as a single function g that takes as input a 77×768 embedding matrix and returns an image.

5.2.2 Permutation invariance

We briefly detail two observations that give rise to an important permutation-invariance property of the function g above.

The first is a general property about the ubiquitous attention operation. Given two matrices of arbitrary dimension, X and X', a common formulation of the attention operation [VSP+17] is

$$A(Q, K, V) = \text{softmax}_{row}\left(\frac{QK^T}{\sqrt{D}}\right)V \tag{5.2.2.1}$$

where the input matrices are a query ($Q = XW_Q$), key ($K = X'W_K$), and value ($V = X'W_V$) matrix constructed as the product of X or X' and a matrix of learned weights. Equation (5.2.2.1) describes self-attention when $X = X'$ and cross-attention when $X \neq X'$. It is a known property that cross-attention is invariant under joint permutation of the rows of K and V, and hence also of X'. At a high level, this is because softmax operates row-wise, and permuting the columns simply permutes the probabilities in the same way; for more detail, we defer to [Fle21, JXG19].

The second critical observation is that the U-Net backbone of Stable Diffusion 1.5 is constructed as a series of self-attention blocks on the image latent and cross-attention blocks between the image latent and the CLIP embedding matrix [RBL+22]. In the cross-attention blocks, the image latent z corresponds to X and the CLIP embedding e corresponds to X'. That is, the function g defined in 5.2.1, as a function of the CLIP embeddings e, is nothing but a series of cross-attention blocks with $X' = e$. Together with the fact that cross-attention is permutation-invariant on the rows of X', we conclude that the function g is permutation-invariant on the rows of e.

This means that the *order* of the 77 token vectors is not relevant, only the tokens themselves as points in $\mathbb{R}^{768}$, which suggests that one can view these embeddings not as *matrices* but instead as *point clouds*. That is, we can view g as a function $g : \mathcal{P}_{77}^{\text{unif}}(\mathbb{R}^{768}) \to \{\text{images}\}$, where $\mathcal{P}_N^{\text{unif}}(\mathbb{R}^d)$ denotes the set of uniform measures on N (possibly nondistinct) discrete points in $\mathbb{R}^d$ (i.e., measures of the form $\frac{1}{N}\sum_{i=1}^N \delta_{x_i}$, where $x_i \in \mathbb{R}^d$). For notational clarity, when viewing embeddings as matrices, we will denote them with Roman letters (e.g., e_0 or e_1), and when viewing them as point clouds, we will use Greek letters (e.g., μ_0

or μ_1). Note that there is a many-to-one equivalence between matrices and point clouds, where a matrix e and point cloud μ are equivalent if the rows of e are exactly the points in μ.

5.2.3 Optimal transport and Wasserstein space

This new point-cloud interpretation of the embedding space comes with a new natural distance metric: the Wasserstein distance. For two point clouds $\mu = \frac{1}{N}\sum_{i=1}^{N}\delta_{x_i}$ and $\nu = \frac{1}{N}\sum_{i=1}^{N}\delta_{y_i}$ in $\mathcal{P}_N^{\text{unif}}(\mathbb{R}^d)$, we define the Wasserstein distance between μ and ν to be

$$W_2(\mu,\nu) := \min_{\sigma\in S_N}\left(\sum_{i=1}^{N}\|x_i - y_{\sigma(i)}\|_2^2\right)^{1/2}, \tag{5.2.3.1}$$

where the optimization is over all permutations $\sigma \in S_N$, the symmetric group on N elements.[1] A map T that satisfies $T(x_i) = y_{\sigma^*(i)}$ for some optimal permutation σ^* is called an "optimal transport map" from μ to ν and is denoted T_μ^ν. When the support of μ is N *distinct* points (which in practice is almost always true for our CLIP embeddings), then at least one such optimal transport map exists (Proposition 2.1 in [PC19]). Though not necessarily unique, the notation T_μ^ν will refer to a particular choice of an optimal map, which we will call "the" optimal transport map.

While we lose the Euclidean space structure that comes with the matrix perspective, the point-cloud perspective comes with an analogous mathematical structure of a (formal) Riemannian manifold [AGS08, Ott01] Specifically, W_2 is a metric on $\mathcal{P}_N^{\text{unif}}(\mathbb{R}^d)$, and $\mathcal{P}_N^{\text{unif}}(\mathbb{R}^d)$ is a subset of the "Wasserstein manifold" of measures with finite second moment, which itself comes with some useful geometric properties. As we begin to explore the landscape

[1] In general, one typically needs to define the Wasserstein distance between discrete measures using the Kantorovich formulation, which allows for the possibility that the optimal "plan" may not be induced by a "map". However, Proposition 2.1 in [PC19] guarantees that, in the case of uniform discrete measures on the same number of points, there exists a permutation that is optimal. This is the only case of interest in this chapter, so we make the definition of Wasserstein distance using permutations, rather than more general plans.

of embedding space with this new perspective, the key geometric property of interest to this chapter is the nice relationship between barycenters and geodesics. Specifically, if $T_{\mu_0}^{\mu_1}$ is an optimal transport map between $\mu_0, \mu_1 \in \mathcal{P}_N^{\text{unif}}(\mathbb{R}^d)$, then there is an explicit expression for a "constant-speed geodesic" $\mu : [0,1] \to \mathcal{P}_N^{\text{unif}}(\mathbb{R}^d)$ from μ_0 to μ_1 given by $t \mapsto \mu_t$ with

$$\mu_t = ((1-t)\,\text{id} + tT_{\mu_0}^{\mu_1})_{\#}\mu_0 = \frac{1}{N}\sum_{i=1}^{N} \delta_{(1-t)x_i + t y_{\sigma^*(i)}}, \tag{5.2.3.2}$$

where the notation $T_{\#}\mu$ denotes the pushforward of μ by the map $T : \mathbb{R}^d \to \mathbb{R}^d$ [AGS08]. In particular, for $t \in [0,1]$, μ_t is the weighted Wasserstein barycenter; i.e., μ_t satisfies

$$\mu_t = \underset{\mu \in \mathcal{P}_N^{\text{unif}}(\mathbb{R}^d)}{\text{argmin}} \left[(1-t)W_2(\mu_0, \mu)^2 + tW_2(\mu_1, \mu)^2\right]. \tag{5.2.3.3}$$

This barycenter definition is exactly analogous to a "weighted average" in Euclidean space, which means these geodesics between point clouds on the Wasserstein manifold are exactly analogous to straight lines — they trace out the shortest path between two points in the space. Thus, in essence, this says that the OT-inspired way to interpolate two N-point clouds μ_0, μ_1 is to pair off points in μ_0 with points in μ_1 using the optimal transport "coupling" (i.e., the pairing of x_i with $T_{\mu_0}^{\mu_1}(x_i) = y_{\sigma^*(i)}$) and then to linearly interpolate between each pair simultaneously. Despite losing the nice structure of the vector space of matrices, the point-cloud interpretation of the embedding space still exhibits a natural way to interpolate between embeddings. Exploiting the point-cloud interpretation of the learned embedding space provides the basis for our novel, geometrically-informed image interpolation technique presented in Section 5.3

5.3 Geometry-informed Image Interpolation Approach

In this work we explore how the permutation-invariance property described in Section 5.2.2 can inform novel image interpolation techniques by operating on the embeddings. Specifically, the discussion above suggests that the natural way to interpolate between embeddings (viewed as point clouds) is to use (5.2.3.2) with the optimal coupling σ^*. This gives the shortest path through the embedding space between the two prompts, meaning that any other method of interpolating the embeddings gives a longer (or at least not shorter) path through Wasserstein space. For example, there is a natural way to interpolate between embeddings e_0, e_1 viewed as matrices, which is to linearly interpolate by setting $e_t = (1-t)e_0 + te_1$. By viewing e_t now as a point cloud, this method traces out a path through Wasserstein space also described by (5.2.3.2), except instead of using the optimal coupling σ^*, it uses the coupling induced by the order of the rows in the CLIP matrices, which we call the "CLIP coupling." In fact, the length of this path is exactly the cost of the associated coupling (consequence of Theorem 8.3.1 in [AGS08]), and for the CLIP coupling, that cost is the standard 2-norm of the difference of corresponding rows. Thus, not only do we know that the CLIP interpolating path is longer (since the optimal coupling cost is at least as small as the CLIP coupling cost), we know precisely how much longer.

One way to assess whether our point-cloud perspective on embedding space is "more natural" than the matrix perspective is to investigate how these path lengths through embedding space relate to notions of similarity in image space. Specifically, for a particular interpolation path through embedding space, there is an associated path through image space obtained by generating the image that corresponds to each interpolated embedding along the embedding path. For any embedding-interpolation path, the start and end embeddings are fixed, and this means that the associated image path connects the images corresponding to the start and end embeddings. If one embedding-interpolation path is "better" than another, its associated image-interpolation path should be smoother, more natural-looking, and contain images which are more similar. In short, the path through

image space should be "shorter." Thus, if considering the CLIP embeddings as point clouds really is a more faithful geometric interpretation of embedding space, we hypothesize that:

1. The geodesic path through embedding space is shorter for prompts which are more similar (i.e., embeddings are closer in Wasserstein distance when the prompts are more similar), and the associated image paths have lower PPL scores,

2. suboptimal couplings give relatively worse embedding interpolations for prompts which are more similar (because suboptimal couplings are relatively more costly when embeddings are close in Wasserstein distance)

3. for a fixed pair of prompt embeddings, a shorter interpolating path between them gives a better associated image interpolation, and

4. this effect increases for prompts which are more similar because the relative path-length increase is less severe.

To properly test these hypotheses, we need a suitable notion of path length in image space. Rather than use a pixel-wise metric, which promotes structurally smooth but unrealistic image interpolations, we seek an image-similarity score that promotes smooth image transitions while retaining realism and context shifts. To this end, we use Perceptual Path Length (PPL) [KLA19] as a surrogate for path length. PPL is defined as the average of image similarity scores between consecutive equally-spaced images along a path, with the intuition that equally spaced points are closer if the overall path is shorter. The standard image similarity score used for PPL is Learned Perceptual Image Patch Similarity (LPIPS) [ZIE+18], which is a standard method for judging image similarity.

Thus, our high-level approach is as follows, as illustrated in Figure 5.1. Two embedding matrices are obtained from two prompt pairs using CLIP. The embedding matrices are treated as point clouds, and three interpolation couplings (OT, CLIP, and a random coupling) are used to define a path between the embedding point clouds. The

interpolated path in embedding space is sampled along a grid, and resultant embedding point clouds are obtained. Both the original embeddings and the interpolated embeddings are used to produce a corresponding trajectory of images in image space. For each grid sample along the interpolated path in embedding space, we produce a corresponding image with a diffusion model (Stable Diffusion), and the resultant images are compared using PPL, based on the LPIPS scores between the k^{th} and $(k+1)^{th}$ image denoted ℓ_k. In Section 5.4, we present experimental results that use this pipeline to explore the validity of the hypotheses listed above.

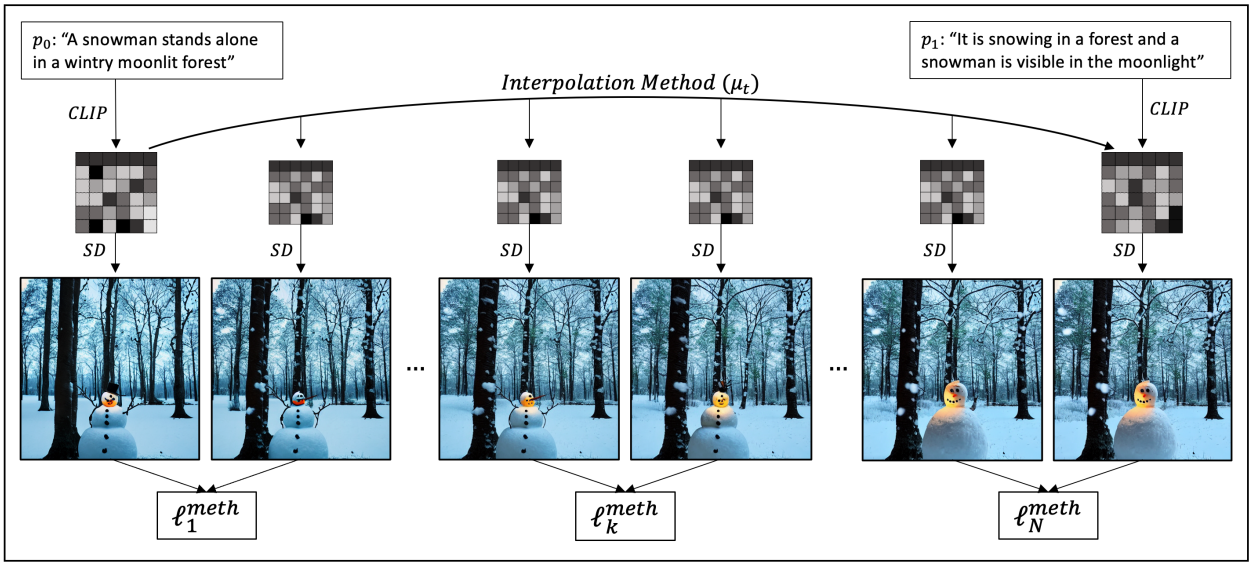


Figure 5.1: Embedding interpolation workflow where SD indicates use of the Stable Diffusion model to produce the associated image.

5.4 Experimental Design and Results

5.4.1 Experimental setup

To evaluate the impact of capturing the geometric structure of CLIP embeddings in image interpolation, we consider many pairs of prompts (see below for more details about how the prompt pairs were selected). For each pair, we follow the pipeline illustrated in

Figure 5.1 for each of three couplings — OT, CLIP, and a random coupling (to create the random coupling, we fix the first row of the CLIP matrices — this row is always the same for any prompt — and randomly pair the last 76 rows in each CLIP matrix). We compute the cost of each coupling and the resulting PPL score for each coupling's image interpolation (we use the AlexNet architecture when computing LPIPS scores for PPL). We note that LPIPS gives lower scores to images that are more similar, and so a smaller PPL score means that the given path through image space is "shorter". Since the OT interpolation method produces a geodesic μ_t^{OT} through embedding space, and each of the other two interpolations μ_t^{clip} and μ_t^{rand} are longer paths, we hypothesize that the LPIPS/PPL scores for the OT method ℓ_k^{ot} will be lower on average than the other methods.

5.4.2 Dataset

To properly test the efficacy of the three interpolation methods, we must produce interpolated images from each method for many pairs. The structure of our hypothesis requires a dataset with prompt pairs that carry a known similarity score. To create such a dataset, we leverage the Crisscrossed Captions dataset [PBC+21], which contains captioned images from the the MS-COCO dataset [LMB+14] that have been manually scored according to their similarity on a scale from 0 to 5, with 5 being most similar. We assume that the similarity score between two images can be extended as a similarity score the associated pair of captions. We discretize the range of similarity scores into bins of width 0.5 and randomly select 1,000 pairs from each similarity group (excluding any pairs where the two captions were identical). These captions are then used as prompts in the SD model with a fixed seed. The results shown reflect the experiment described in Section 5.4.1 applied to 10,000 caption/prompt pairs. The Crisscrossed Captions dataset has a custom, open source license at https://github.com/google-research-datasets/Crisscrossed-Captions/blob/master/LICENSE. The MS-COCO dataset has a Creative Commons Attribution 4.0 License, and the SD model has a CreativeML Open

RAIL-M License. CPU and GPU workers were used on an internal cluster and the total compute cost for the experiments was 1,094 CPU hours and 2,172 GPU hours. Preliminary and failed experiments accounted for an additional approximately 7,000 CPU hours and 575 GPU hours of compute resources.

5.4.3 Results

Figure 5.2: Image interpolations for each method across four selected prompt pairs of varying similarity.

To help build the reader's intuition, we start by showing the qualitative impact of the geometric relationships being considered. Figure 5.2 shows a selection of interpolated images, where for each of the four panels, the sub-images on the far left and far right correspond to the images produced for each prompt pair, i.e., the endpoints of our path in image space. For each panel, the rows of sub-figures are ordered from top to bottom as OT, CLIP, and random. Examination of the image trajectories shows cases where the CLIP and/or random method appears to hallucinate objects in the interpolated trajectory. The prompt pairs shown were selected based on having quantitatively larger differences in the associated path lengths for illustration purposes. However, as will be shown subsequently, the OT path is found to perform comparably or above the CLIP or random couplings across

the board.

For a quantitative evaluation, we compute the perceptual path length (PPL) (as defined above in Section 5.4.1) [KLA19] for each interpolation method (OT, CLIP, random) as a measurement of transition smoothness between the trajectory of interpolated images for all 10,000 pairs of prompts. Smaller PPL values are desirable and correspond to higher smoothness across the interpolated image trajectories. In the subsequent plots, we report PPL and coupling costs over the collection of all 1000 prompt pairs of each similarity level.

The boxplots of PPL scores and coupling costs are plotted for each method and similarity group in Figure 5.3. To test our hypothesis, we assess the statistical significance of both the difference in median PPL and the difference in median coupling cost between the OT interpolation results and the CLIP or random couplings, respectively within each similarity group. Significance of p-values for testing the difference in median PPL within each interpolation method and similarity group are reported in Table 5.1. As hypothesized, the impact of embedding path is more significant for more similar pairs of prompts. All tests for the difference in median coupling cost between optimal transport and each other interpolation method are highly significantly different, $p < 0.0001$, for all similarity groups. Additionally, the path length decreases as a function of similarity and the significance of an optimal coupling grows as the similarity increases as hypothesized based on geometric intuition. The PPL scores and coupling costs were determined to be not normally distributed, so for both sets of tests we employed a Wilcoxon signed-rank test to test the null hypothesis that the median of the difference is zero. In addition to the typical assumptions of independence and randomness, the Wilcoxon signed-rank test assumes continuous data distributed symmetrically about the median [RT21].

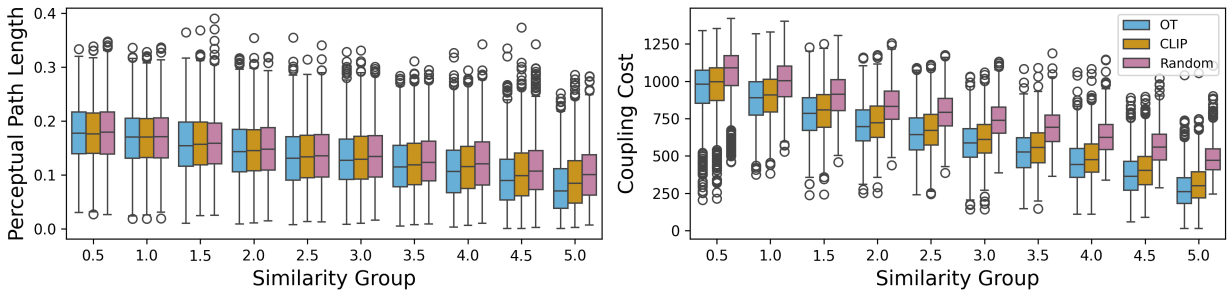


Figure 5.3: PPL (left) and coupling cost (right) by interpolation method and similarity group.

Table 5.1: Significance of p-values for the Wilcoxon test comparing the median of the PPL scores. ($p < 0.05^{*}$, $p < 0.01^{**}$, $p < 0.001^{***}$)

	Similarity Group									
	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
OT vs. CLIP PPL	-	-	-	-	***	*	***	***	***	***
OT vs. Random PPL	*	-	**	***	***	***	***	***	***	***

5.5 Conclusions and Future Directions

In this work we observe that interpolating using the OT coupling in general results in shorter image paths than using the linear and random couplings, and that the improvement is more substantial for more similar prompts. Both of these observations are consistent with the hypothesis that path length through Wasserstein space is reflected by "path length" through image space. We also notice that in many cases where OT drastically outperforms the other two methods, it does so because the other methods produce interpolated images which are relatively very different than either end image. This suggests that embedding space has a "convexity" property with respect to Wasserstein distance that it does not have when the embedding matrices are handled as matrix objects. Concretely, we find given two embeddings of a particular class, an embedding "between" them in Wasserstein space is more likely to be in the same class than an embedding "between" them in matrix space. Our experimental results suggest that the point-cloud perspective indeed does a better job of capturing the "geometry" of embedding space in ways that reflect more desirable

properties in image interpolation.

In particular, it is perhaps surprising that the random couplings, while noticeably worse than the optimal couplings, still result in reasonably good image interpolations. Additionally, anecdotal experiments suggest that even maximally-costly couplings (i.e., couplings that maximize the objective quantity in (5.2.3.1)) still give image interpolations which are not catastrophic. In terms of the geometry of the Wasserstein manifold, these interpolating paths using different couplings are all "straight" (even though they are not all "geodesics"). This suggests some redundancy in the embeddings' geometry and raises a natural question: How far can we push the interpolating paths before they result in catastrophic image behavior? Investigating this more general question could help shed even more light on the geometry of embedding-space and how it interacts with the "geometry" of image-space.

There are also more general questions one could ask about interpolating between more than two embeddings. The notion of a Wasserstein barycenter is perfectly well-defined when there are more than two measures (the definition is exactly analogous to (5.2.3.3)), which means one could compute the Wasserstein barycenter of many embeddings and observe the corresponding image outputs. Investigating these multi-way interpolations would provide much deeper insight into the "manifold" structure of the embedding-space, which is difficult to fully investigate with only one-parameter curves. While computing Wasserstein barycenters of multiple measures is substantially more expensive than computing an optimal transport map (there is no closed-form equation like (5.2.3.2)), one could use the techniques of linearized optimal transport [LKKC25, MC23, CHKM25] to approximate the Wasserstein barycenter much more computationally efficiently.

Deeper exploration of these questions could uncover properties and conditions under which prompt interpolation results in thematically consistent and smooth images, giving rise not only to improved methods for image interpolation that rely on prompt interpolation [WG23], but also informing scenarios where prompt interpolation or manipulation is an

auxiliary step, such as variant refinement [DPP24]. Another area for future work is in developing additional metrics to capture pair-wise changes in images. The path length surrogate considered in this work leverages a standard image similarity score, LPIPS, but that score does not explicitly capture or penalize for contextual and content shifts between image pairs. Additional work in this area would enable more direct connections with applications like hallucination detection and associated mitigation strategies.

Finally, the scope of this work focused solely on Stable Diffusion 1.5, whose architecture differs from the latest Stable Diffusion model (3.5) available at the time of writing. While architectural innovations in the latest models present barriers to direct extension of the proposed approach, the point cloud perspective may still better preserve underlying geometry and desired invariances than performing operations like token concatenations used in Stable Diffusion 3.5. Therefore, an important future direction would be extending and evaluating the point cloud framing for shared, multi-modal token spaces and transformer-based denoising architectures like those of current state-of-the-art models. However, although it is not the current "state-of-the-art" model, the fact that it can be run locally means there is a substantial benefit to investigating Stable Diffusion 1.5 specifically. A better understanding of the geometry underlying this model could provide insights that could be thoroughly explored without the need for heavy resources, thus providing a theoretical framework for a large class of geometric experiments that can still be run on an accessible testbed.

Acknowledgements

Chapter 5, in full, is a reprint of the material in [KDFE26]: Nicholas Karris, Luke Durell, Javier Flores, and Tegan Emerson. Which way from B to A: The role of embedding geometry in image interpolation for Stable Diffusion. In *Proceedings of the 1st Conference on Topology, Algebra, and Geometry in Data Science (TAG-DS 2025)*, volume 321 of *Proceedings*

of Machine Learning Research, pages 114–125. PMLR, 2026. The dissertation author is the primary author and investigator of this work, which was conducted under the Laboratory Directed Research and Development Program at PNNL, a multi-program national laboratory operated by Battelle for the U.S. Department of Energy under contract DE-AC05-76RL01830.

Bibliography

[AG13] Luigi Ambrosio and Nicola Gigli. A user's guide to optimal transport. In Benedetto Piccoli and Michel Rascle, editors, *Modelling and Optimisation of Flows on Networks: Cetraro, Italy 2009*, volume 2062 of *Lecture Notes in Mathematics*, pages 1–155. Springer, Berlin, Heidelberg, 2013.

[AGS08] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics. ETH Zürich. Birkhäuser, Basel, 2nd edition, 2008.

[AHH18] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: On the curvature of deep generative models. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*, 2018.

[AK21] Hassan Arbabi and Ioannis G. Kevrekidis. Particles to partial differential equations parsimoniously. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(3):033137, 2021.

[Atk89] Kendall E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley & Sons, New York, 2nd edition, 1989.

[BB00] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[BB20] Randall Balestriero and Richard G Baraniuk. Mad max: Affine spline insights into deep learning. *Proceedings of the IEEE*, 109(5):704–727, 2020.

[BDM12] Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment problems: revised reprint*. SIAM, 2012.

[BHB25] Randall Balestriero, Ahmed Imtiaz Humayun, and Richard G Baraniuk. On the geometry of deep learning. *NOTICES OF THE AMERICAN MATHEMATICAL SOCIETY*, 72(4), 2025.

[Bil95] Patrick Billingsley. *Probability and Measure*. Wiley Series in Probability and Statistics. Wiley-Interscience, 3rd edition, 1995.

[BL19] Sergey G. Bobkov and Michel Ledoux. One-dimensional empirical measures, order statistics, and Kantorovich transport distances. *Memoirs of the American Mathematical Society*, 261(1259), 2019.

[BR25] Riccardo Bonalli and Alessandro Rudi. Non-parametric learning of stochastic differential equations with non-asymptotic fast rates of convergence. *Foundations of Computational Mathematics*, 25:1–56, 2025.

[Bre91] Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on Pure and Applied Mathematics*, 44(4):375–417, 1991.

[BRPP15] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

[BS08] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer, 3rd edition, 2008.

[BSC+23] Giorgio Barnabò, Federico Siciliano, Carlos Castillo, Stefano Leonardi, Preslav Nakov, Giovanni Da San Martino, and Fabrizio Silvestri. Deep active learning for misinformation detection using geometric deep learning. *Online Social Networks and Media*, 33:100244, 2023.

[BWWW08] Peter N. Brown, Homer F. Walker, Rebecca Wasyk, and Carol S. Woodward. On using approximate finite differences in matrix-free newton–krylov methods. *SIAM Journal on Numerical Analysis*, 46(4):1892–1911, 2008.

[CDGK04] Ligang Chen, Pablo G. Debenedetti, William C. Gear, and Ioannis G. Kevrekidis. From molecular dynamics to coarse self-similar solutions: a simple example using equation-free computation. *Journal of non-Newtonian Fluid Mechanics*, 120(1-3):215–223, 2004.

[CGP16] Yongxin Chen, Tryphon T. Georgiou, and Michele Pavon. On the relation between optimal transport and schrödinger bridges: A stochastic control viewpoint. *Journal of Optimization Theory and Applications*, 169(2):671–691, 2016.

[CGS10] G. Carlier, A. Galichon, and F. Santambrogio. From knothe's transport to brenier's map and a continuation method for optimal transport. *SIAM Journal on Mathematical Analysis*, 41(6):2554–2576, 2010.

[CHKM25] Alexander Cloninger, Keaton Hamm, Varun Khurana, and Caroline Moosmüller. Linearized Wasserstein dimensionality reduction with approximation guarantees. *Applied and Computational Harmonic Analysis*, 74:101718, 2025.

[CNWR25] Sinho Chewi, Jonathan Niles-Weed, and Philippe Rigollet. *Statistical Optimal Transport*. Lecture Notes in Mathematics. Springer, 2025.

[CSJ+23] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. DiffDock: Diffusion Steps, Twists, and Turns for molecular Docking, 2023. Preprint, arXiv:2210.01776.

[CTG+23] Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z. Li. A survey on generative diffusion model, 2023. Preprint, arXiv:2209.02646.

[DN21] Prafulla Dhariwal and Alex Nichol. Diffusion models Beat GANs on image synthesis, 2021. Preprint, arXiv:2105.05233.

[DPP24] Niklas Deckers, Julia Peters, and Martin Potthast. Manipulating Embeddings of Stable Diffusion Prompts. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, pages 7636–7644. International Joint Conferences on Artificial Intelligence Organization, 2024.

[EE03] Weinan E and Bjorn Engquist. The heterogeneous multiscale methods. *Communications in Mathematical Sciences*, 1(1):87–132, 2003.

[EGH00] Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. Finite volume methods. In J. L. Lions and Philippe Ciarlet, editors, *Solution of Equation in Rn (Part 3), Techniques of Scientific Computing (Part 3)*, volume 7 of *Handbook of Numerical Analysis*, pages 713–1020. Elsevier, 2000.

[EKB+24] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 12606–12633. JMLR.org, 2024.

[ELVE05] Weinan E, Di Liu, and Eric Vanden-Eijnden. Analysis of multiscale methods for stochastic differential equations. *Communications on Pure and Applied Mathematics*, 58(11):1544–1585, 2005.

[Eva25] Nicolas Evangelou. Agent-based modeling repository. `https://gitlab.com/nicolasevangelou/agent_based/-/tree/main?ref_type=heads`, 2025. Accessed: October 26, 2025.

[FEC+24] Gianluca Fabiani, Nikolaos Evangelou, Tianqi Cui, Juan M Bello-Rivas, Cristina P Martin-Linares, Constantinos Siettos, and Ioannis G Kevrekidis. Task-oriented machine learning surrogates for tipping points of agent-based models. *Nature communications*, 15(1):4117, 2024.

[FJ17] Nicolas Fournier and Benjamin Jourdain. Stochastic particle approximation of the Keller–Segel equation and two-dimensional generalization of Bessel processes. *The Annals of Applied Probability*, 27(5):2807–2861, 2017.

[Fle21] François Fleuret. Deep learning attention mechanisms. Technical report, University of Geneva, Geneva, Switzerland, 2021.

[Fri64] Avner Friedman. *Partial Differential Equations of Parabolic Type*. Prentice-Hall, 1964.

[Fri75] Avner Friedman. *Stochastic Differential Equations and Applications*, volume 1. Academic Press, 1975.

[FVG+25] Gianluca Fabiani, Hannes Vandecasteele, Somdatta Goswami, Constantinos Siettos, and Ioannis G. Kevrekidis. Enabling local neural operators to perform equation-free system-level analysis. Preprint, arXiv:2505.02308, 2025.

[GAA+23] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-Or. An image is Worth One Word: Personalizing text-to-image generation using Textual Inversion. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*, 2023.

[GBYK23] Somdatta Goswami, Aniruddha Bora, Yue Yu, and George Em Karniadakis. Physics-informed deep neural operator networks. In Timon Rabczuk and Klaus-Jürgen Bathe, editors, *Machine Learning in Modeling and Simulation: Methods and Applications*, Computational Methods in Engineering & the Sciences, pages 219–254. Springer, Cham, 2023.

[Gea01] William C. Gear. Projective integration methods for distributions. *NEC Report*, 2001.

[GGK09] C. W. Gear, D. Givon, and I. G. Kevrekidis. Virtual slow manifolds: The fast stochastic case. *AIP Conference Proceedings*, 1168(1):17–20, 2009.

[Gig11] Nicola Gigli. On Hölder continuity-in-time of the optimal transport map towards measures along a curve. *Proceedings of the Edinburgh Mathematical Society*, 54(2):401–409, 2011.

[GK03a] C. W. Gear and Ioannis G. Kevrekidis. Projective methods for stiff differential equations: Problems with gaps in their eigenvalue spectrum. *SIAM Journal on Scientific Computing*, 24(4):1091–1106, 2003.

[GK03b] C. W. Gear and Ioannis G. Kevrekidis. Telescopic projective methods for parabolic differential equations. *Journal of Computational Physics*, 187(1):95–109, 2003.

[GKKZ05] Gear, Kaper, Kevrekidis, and Zagaris. Projecting to a slow manifold: Singularly perturbed systems and legacy codes. *Siam Journal on Applied Dynamical Systems*, 4:711–732, 2005.

[GKSK22] Somdatta Goswami, Katiana Kontolati, Michael D. Shields, and George Em Karniadakis. Deep transfer operator learning for partial differential equations under conditional shift. *Nature Machine Intelligence*, 4(12):1155–1164, 2022.

[GKT02] C. W. Gear, Ioannis G. Kevrekidis, and Constantinos Theodoropoulos. 'Coarse' integration/bifurcation analysis via microscopic simulators: Micro-Galerkin methods. *Computers & Chemical Engineering*, 26(7–8):941–963, 2002.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[HMT+22] Guillaume Huguet, D. S. Magruder, Alexander Tong, Oluwadamilola Fasina, Manik Kuchroo, Guy Wolf, and Smita Krishnaswamy. Manifold interpolating optimal-transport flows for trajectory inference. *Advances in Neural Information Processing Systems*, 35:29705–29718, 2022.

[HWLY24] Qiyuan He, Jinghao Wang, Ziwei Liu, and Angela Yao. AID: Attention interpolation of text-to-image diffusion. *Advances in Neural Information Processing Systems*, 37:97766–97799, 2024.

[JXG19] Shuiwang Ji, Yaochen Xie, and Hongyang Gao. A Mathematical View of Attention models in deep learning. Technical report, Texas A&M University, College Station, TX, USA, 2019.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, 2015.

[KDFE26] Nicholas Karris, Luke Durell, Javier Flores, and Tegan Emerson. Which way from B to A: The role of embedding geometry in image interpolation for Stable Diffusion. In *Proceedings of the 1st Conference on Topology, Algebra, and Geometry in Data Science (TAG-DS 2025)*, volume 321 of *Proceedings of Machine Learning Research*, pages 114–125. PMLR, 2026.

[KGH+03] Ioannis G. Kevrekidis, C. William Gear, James M. Hyman, Panagiotis G. Kevrekidis, Olof Runborg, and Constantinos Theodoropoulos. Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.

[KGH04] Ioannis G. Kevrekidis, C. William Gear, and Gerhard Hummer. Equation-free: The computer-aided analysis of complex multiscale systems. *AIChE Journal*, 50(7):1346–1355, 2004.

[KKQ04] Carl T. Kelley, Ioannis G. Kevrekidis, and Liang Qiao. Newton-krylov solvers for time-steppers. Preprint, arXiv:math/0404374, 2004.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019.

[KLM24] Nikola B. Kovachki, Samuel Lanthaler, and Hrushikesh Mhaskar. Data complexity estimates for operator learning. Preprint, arXiv:2405.15992, 2024.

[KNK+25] Nicholas Karris, Evangelos A. Nikitopoulos, Ioannis G. Kevrekidis, Seungjoon Lee, and Alexander Cloninger. Using linearized optimal transport to predict the evolution of stochastic particle systems. Preprint, arXiv:2408.01857, 2025.

[Kno57] Herbert Knothe. Contributions to the theory of convex bodies. *Michigan Mathematical Journal*, 4(1):39–52, 1957.

[KNS+19] Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized sliced wasserstein distances. *Advances in neural information processing systems*, 32, 2019.

[KZW+23] Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating Concepts in text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22634–22645, 2023.

[Lee23] Minhyeok Lee. The geometry of feature space in deep learning models: A holistic perspective and comprehensive review. *Mathematics*, 11(10):2375, 2023.

[Lee25] Younghoon Lee. Enhancing plant health classification via diffusion model-based data augmentation. *Multimedia Systems*, 31(2):143, 2025.

[Léo14] Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport. *Discrete and Continuous Dynamical Systems*, 34(4):1533–1574, 2014.

[LJP+21] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021.

[LKGK03] Ju Li, Panayotis G. Kevrekidis, C. William Gear, and Ioannis G. Kevrekidis. Deciding the nature of the coarse equation through microscopic simulations: The baby-bathwater scheme. *Multiscale Modeling & Simulation*, 1(3):391–407, 2003.

[LKKC25] Jun Linwu, Varun Khurana, Nicholas Karris, and Alexander Cloninger. Linearized optimal transport pyLOT Library: A Toolkit for Machine learning on Point Clouds, 2025. Preprint, arXiv:2502.03439.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV 2014)*, pages 740–755. Springer, 2014.

[Lot08] John Lott. Some geometric calculations on Wasserstein space. *Communications in Mathematical Physics*, 277:423–437, 2008.

[LPSK23] Seungjoon Lee, Yorgos M. Psarellis, Constantinos I. Siettos, and Ioannis G. Kevrekidis. Learning black- and gray-box chemotactic PDEs/closures from agent based Monte Carlo simulation data. *Journal of Mathematical Biology*, 87(1):15, 2023.

[LSU68] O. A. Ladyzhenskaya, V. A. Solonnikov, and N. N. Uraltseva. *Linear and Quasilinear Equations of Elliptic Type*, volume 46 of *Translations of Mathematical Monographs*. American Mathematical Society, 1968.

[LvdWH+24] Senmao Li, Joost van de Weijer, Taihang Hu, Fahad Khan, Qibin Hou, Yaxing Wang, and Jian Yang. Get what You Want, not what You Don't: Image Content Suppression for text-to-image diffusion models. In *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*, 2024.

[LWLL25] Shenghao Li, Zhanpeng Wang, Zhongxuan Luo, and Na Lei. An optimal transport-guided diffusion framework with mitigating mode mixture. *Neurocomputing*, 616:128910, 2025.

[LZW+25] Mingzhe Li, Gehao Zhang, Zhenting Wang, Shiqing Ma, Siqi Pan, Richard Cartwright, and Juan Zhai. EDITOR: Effective and Interpretable Prompt Inversion for text-to-image diffusion models, 2025. Preprint, arXiv:2506.03067.

[MA22] François Mazé and Faez Ahmed. Diffusion models Beat GANs on Topology optimization, 2022. Preprint, arXiv:2208.09591.

[MBNWW23] Tudor Manole, Sivaraman Balakrishnan, Jonathan Niles-Weed, and Larry Wasserman. Central limit theorems for smooth optimal transport maps, 2023.

[MBNWW24] Tudor Manole, Sivaraman Balakrishnan, Jonathan Niles-Weed, and Larry Wasserman. Plugin estimation of smooth optimal transport maps. *The Annals of Statistics*, 52(3):966–998, 2024.

[MC23] Caroline Moosmüller and Alexander Cloninger. Linear optimal transport embedding: Provable Wasserstein classification for certain rigid transformations and perturbations. *Information and Inference: A Journal of the IMA*, 12(1):363–389, 2023.

[MDC20] Quentin Mérigot, Alex Delalande, and Frédéric Chazal. Quantitative stability of optimal transport maps and linearization of the 2-Wasserstein space. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3186–3196. PMLR, 2020.

[Mha23] H. N. Mhaskar. Local approximation of operators. *Applied and Computational Harmonic Analysis*, 64:194–228, 2023.

[MNW24] Tudor Manole and Jonathan Niles-Weed. Sharp convergence rates for empirical optimal transport with smooth costs. *The Annals of Applied Probability*, 34(1B):1108–1135, 2024.

[NFM18] Richard A. Norton, Colin Fox, and Malcolm E. Morrison. Numerical approximation of the frobenius–perron operator using the finite volume method. *SIAM Journal on Numerical Analysis*, 56(1):570–589, 2018.

[Ott01] Felix Otto. The geometry of dissipative evolution equations: The porous medium equation. *Communications in Partial Differential Equations*, 26(1–2):101–174, 2001.

[Pap65] Athanasios Papoulis. *Random variables and stochastic processes*. McGraw Hill, 1965.

[Pav14] Grigorios A. Pavliotis. *Stochastic Processes and Applications: Diffusion Processes, the Fokker–Planck and Langevin Equations*, volume 60 of *Texts in Applied Mathematics*. Springer, 2014.

[PBC+21] Zarana Parekh, Jason Baldridge, Daniel Cer, Austin Waters, and Yinfei Yang. Crisscrossed Captions: Extended Intramodal and Intermodal semantic Similarity Judgments for MS-COCO, 2021. Preprint, arXiv:2004.15020.

[PC19] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019.

[PKC+23] Yong-Hyun Park, Mingi Kwon, Jaewoong Choi, Junghyo Jo, and Youngjung Uh. Understanding the latent space of diffusion models through the lens of riemannian geometry. *Advances in Neural Information Processing Systems*, 36:24129–24142, 2023.

[PKKR18] Se Rim Park, Soheil Kolouri, Shinjini Kundu, and Gustavo K. Rohde. The cumulative distribution transform and linear pattern classification. *Applied and Computational Harmonic Analysis*, 45(3):616 – 641, 2018.

[PWM+25] Edward Phillips, Sean Wu, Soheila Molaei, Danielle Belgrave, Anshul Thakur, and David Clifton. Geometric uncertainty for detecting and correcting hallucinations in llms. Preprint, arXiv:2509.13813, 2025.

[QEKK06] Liang Qiao, Radek Erban, Carl T. Kelley, and Ioannis G. Kevrekidis. Spatially distributed stochastic systems: Equation-free and equation-assisted preconditioned computations. *The Journal of chemical physics*, 125(20), 2006.

[RBL+22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2022.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable visual models from natural language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[RMGK03] Ramiro Rico-Martínez, C. Gear, and Ioannis Kevrekidis. Coarse projective kmc integration: Forward/reverse initial and boundary value problems. *Journal of Computational Physics*, 196:474–489, 2003.

[Ros52] Murray Rosenblatt. Remarks on a multivariate transformation. *The annals of mathematical statistics*, 23(3):470–472, 1952.

[RT21] Kandethody M. Ramachandran and Chris P. Tsokos. Chapter 12 - Nonparametric Statistics. In Kandethody M. Ramachandran and Chris P. Tsokos, editors, *Mathematical Statistics with Applications in R (Third Edition)*, pages 491–530. Academic Press, 2021.

[San15] Filippo Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*, volume 87 of *Progress in Nonlinear Differential Equations and Their Applications*. Birkhäuser, Cham, 2015.

[SGK12] C. I. Siettos, C. W. Gear, and I. G. Kevrekidis. An equation-free approach to agent-based computation: Bifurcation analysis and control of stationary states. *Europhysics Letters*, 99(4):48007, 2012.

[SGOK05] S. Setayeshgar, C. W. Gear, H. G. Othmer, and I. G. Kevrekidis. Application of coarse integration to bacterial chemotaxis. *Multiscale Modeling & Simulation*, 4(1):307–327, 2005.

[Sie10] Constantinos I Siettos. System level numerical analysis of a monte carlo simulation of the e. coli chemotaxis. Preprint, arXiv:1011.1467, 2010.

[SM25] Shinnosuke Saito and Takashi Matsubara. Image interpolation with score-based Riemannian metrics of diffusion models, 2025. Preprint, arXiv:2504.20288.

[SME21] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion Implicit models. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*, 2021.

[SNB+24] Anparasy Sivaanpu, Michelle Noga, Harald Becher, Kumaradevan Punithakumar, and Lawrence H Le. Denoising Echocardiography with an Improved diffusion model. In *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1–4, 2024.

[SPK03] C. I. Siettos, C. C. Pantelides, and I. G. Kevrekidis. Enabling dynamic process simulators to perform alternative tasks: A time-stepper-based toolkit for computer-aided analysis. *Industrial & Engineering Chemistry Research*, 42:6795–6801, 2003.

[SS24] M Sivabalamurugan and TR Swapna. Deepfake detection and classification using local surface geometrical features. In *2024 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI)*, pages 1–6, 2024.

[SSDK+21] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic Differential equations. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*, 2021.

[SST+24] Kotaro Sakamoto, Ryosuke Sakamoto, Masato Tanabe, Masatomo Akagawa, Yusuke Hayashi, Manato Yaguchi, Masahiro Suzuki, and Yutaka Matsuo. The geometry of diffusion models: Tubular neighbourhoods and singularities. In *ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024.

[Ste00] Angela Stevens. The derivation of chemotaxis equations as limit dynamics of moderately interacting stochastic many-particle systems. *SIAM Journal on Applied Mathematics*, 61(1):183–212, 2000.

[THW+20] Alexander Tong, Jessie Huang, Guy Wolf, David van Dijk, and Smita Krishnaswamy. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. *Proceedings of Machine Learning Research*, 119:9526–9536, 2020.

[TLGD24] Antonio Terpin, Nicolas Lanzetti, Martín Gadea, and Florian Dorfler. Learning diffusion at lightspeed. In *The Thirty-Eighth Annual Conference on Neural Information Processing Systems*, 2024.

[VB20] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International conference on machine learning*, pages 9786–9796, 2020.

[VE03] Eric Vanden-Eijnden. Numerical techniques for multi-scale dynamical systems with stochastic effects. *Communications in Mathematical Sciences*, 1(2):385–391, 2003.

[VKCK25] Hannes Vandecasteele, Nicholas Karris, Alexander Cloninger, and Ioannis G. Kevrekidis. Optimal transport, timesteppers, Newton-Krylov methods and steady states of collective particle dynamics. Preprint, arXiv:2512.24567, 2025.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[WG23] Clinton Wang and Polina Golland. Interpolating between Images with diffusion models, 2023. Preprint, arXiv:2303.05736.

[WLHG24] Ruochen Wang, Ting Liu, Cho-Jui Hsieh, and Boqing Gong. On Discrete Prompt optimization for diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 50992–51011. PMLR, 2024.

[WSB+13] Wei Wang, Dejan Slepčev, Saurav Basu, John A. Ozolek, and Gustavo K. Rohde. A linear optimal transportation framework for quantifying and visualizing variations in sets of images. *International Journal of Computer Vision*, 101:254–269, 2013.

[XHFS25] Hongyang Xie, Hongyang He, Boyang Fu, and Victor Sanchez. Grdt: Towards robust deepfake detection using geometric representation distribution and texture. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 734–744, 2025.

[XL24] Yewei Xu and Qin Li. Forward-Euler time-discretization for Wasserstein gradient flows can be wrong. Preprint, arXiv:2406.08209, 2024.

[YBS+25] Eric Yeats, John Buckheit, Sarah McGuire Scullen, Brendan Kennedy, Loc Truong, Davis Brown, Bill Kay, Cliff Joslyn, Tegan Emerson, and Michael J Henry. What do geometric hallucination detection metrics actually measure? In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2025.

[YSY+25] Qingtao Yu, Jaskirat Singh, Zhaoyuan Yang, Peter Henry Tu, Jing Zhang, Hongdong Li, Richard Hartley, and Dylan Campbell. Probability Density Geodesics in image diffusion latent space. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27989–27998, 2025.

[YYX+24] Zhaoyuan Yang, Zhengyang Yu, Zhiwei Xu, Jaskirat Singh, Jing Zhang, Dylan Campbell, Peter Tu, and Richard Hartley. IMPUS: Image Morphing with Perceptually-Uniform sampling using diffusion models. In *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*, 2024.

[ZCL+25] Rui Zhang, Yaosen Chen, Yuegen Liu, Wei Wang, Xuming Wen, and Hongxia Wang. TVG: A training-free Transition video generation method

with diffusion models. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(8):7471–7484, 2025.

[ZIE+18] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[ZKG05] Yu Zou, Ioannis Kevrekidis, and Roger Ghanem. Equation-free dynamic renormalization: Self-similarity in multidimensional particle system dynamics. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 72(4):046702, 2005.

[ZKG06] Yu Zou, Ioannis G. Kevrekidis, and Roger G. Ghanem. Equation-free particle-based computations: Coarse projective integration and coarse dynamic renormalization in 2d. *Industrial & engineering chemistry research*, 45(21):7002–7014, 2006.

[ZVG+12] Antonios Zagaris, Christophe Vandekerckhove, C. William Gear, Tasso J. Kaper, and Ioannis G. Kevrekidis. Stability and stabilization of the constrained runs schemes for equation-free projection to a slow manifold. *Discrete and Continuous Dynamical Systems*, 32(8):2759–2803, 2012.

[ZWW+24] Xulu Zhang, Xiao-Yong Wei, Jinlin Wu, Tianyi Zhang, Zhaoxiang Zhang, Zhen Lei, and Qing Li. Compositional inversion for stable diffusion models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, volume 38 of *AAAI'24/IAAI'24/EAAI'24*, pages 7350–7358. AAAI Press, 2024.

[ZYHT07] Lei Zhu, Yan Yang, Steven Haker, and Allen Tannenbaum. An image Morphing Technique based on optimal Mass Preserving mapping. *IEEE Transactions on Image Processing*, 16(6):1481–1495, 2007.

[ZZX+24] Kaiwen Zhang, Yifan Zhou, Xudong Xu, Bo Dai, and Xingang Pan. DiffMorpher: Unleashing the Capability of diffusion models for image Morphing. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7912–7921, 2024.