

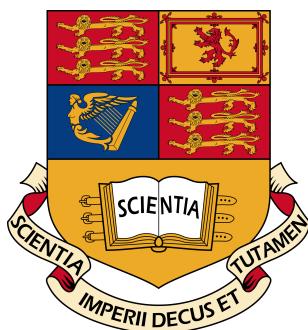
IMPERIAL COLLEGE LONDON

DEPARTMENT OF MATHEMATICS

Numerical methods for Optimal Transport—Schrödinger Bridge Approach and Proximal Splitting Method

Author:
Wenxin Liu

Supervisor:
Dr. Dante Kalise



Submitted in partial fulfillment of the requirements for the MSc degree in Master of Science of Imperial College London

September 2025

Abstract

This dissertation explores the theory and numerical methods for solving the optimal transport(OT) problem, particularly focusing on optimal transport problem with obstacles. We begin with various numerical algorithms for direct solve of OT problems, including dynamic energy minimization techniques, entropic regularization for Sinkhorn distances. This is followed by a discussion of Schrödinger bridges(including gradient flows and reflected Schrödinger bridges) and stochastic optimal transport methods, highlighting their potential and efficiency for addressing OT problems. The dissertation also covers the implementation of these algorithms, and novelly adapts reflected Schrödinger bridges for solving the OT problem with obstacles. We conclude with a discussion on comparison of these methods in terms of their performance and applicability to different OT problems.

Keywords: Optimal Transport with Obstacles, Schrödinger Bridge, Stochastic Optimal Transport, Sinkhorn Algorithm.

Acknowledgments

This dissertation would not have been possible without the support and guidance of many individuals. I would like to express my deepest gratitude to my supervisor, Dr. Dante Kalise, for his invaluable insights and encouragement throughout this research. He opened up multiple avenues of inquiry, suggested fruitful research directions, and consistently provided relevant literature and rigorous academic guidance that enabled me to delve deeper into the theory and refine my ideas.

I also wish to thank my colleagues and friends at Imperial College London for their stimulating discussions and camaraderie. In particular, I am grateful to Hantang Guo, who always provided helpful feedback and support on the theoretical aspects of my research.

Finally, I am deeply thankful to my family for their unwavering support and belief in my abilities.

Notation

Sets, spaces and basic symbols

\mathbb{R}^n	n -dimensional Euclidean space.
$\Omega \subset \mathbb{R}^n$	Connected, smooth, bounded domain (also denoted \mathcal{X} in some sections).
$\partial\Omega$	Boundary of Ω (also written $\partial\mathcal{X}$).
$\overline{\Omega} = \Omega \cup \partial\Omega$	Closure of Ω .
\mathbb{S}^{n-1}	Unit sphere in \mathbb{R}^n .
$H^1(\Omega), L^2(\Omega)$	Sobolev space, square-integrable space.
$\mathcal{P}_2(\Omega)$	Probability measures on Ω with finite second moment.
$\langle \cdot, \cdot \rangle$	Euclidean inner product; also duality pairing where clear from context.
$\ \cdot\ _2$	Euclidean norm.

Measures, densities and couplings

ρ_0, ρ_1	Prescribed initial/terminal probability densities on Ω .
$\rho(t, x)$	Time-dependent density (law of the state at time t).
$\pi(\rho_0, \rho_1)$	Set of static couplings with marginals ρ_0, ρ_1 .
$\rho_{01}(x, y)$	Joint endpoint density for (X_0, X_1) .
$W_p(\mu, \nu)$	p -Wasserstein distance. W_2 used most frequently.
$c(x, y)$	Transport cost; in OT typically $c(x, y) = \frac{1}{2}\ x - y\ ^2$.

OT: dynamic (Benamou–Brenier) variables

$v(t, x)$	Velocity field in the fluid formulation.
$\rho(t, x)$	Density field; satisfies the continuity equation.
$m(t, x) = \rho v$	Momentum field (convexification variable).
\mathcal{C}	Continuity-constraint set: $\partial_t \rho + \operatorname{div}_x(m) = 0$ with endpoint conditions.
$J(m, \rho)$	Benamou–Brenier integrand, $J = \ m\ ^2/(2\rho)$ for $\rho > 0$.
$1_{\{\cdot\}}$	Indicator function of a set/event.

Schrödinger bridge (SB) and stochastic control

$\varepsilon > 0$	Diffusion parameter (variance scale).
W_t	Standard Brownian motion in \mathbb{R}^n .
X_t	Controlled diffusion (state process).

$f(x, t)$	Drift/prior vector field (possibly nonlinear).
$u(x, t)$	Control field; optimal $u^* = \nabla \lambda$ or $u^* = \varepsilon \nabla \ln \varphi$ in contexts.
\mathbb{E}	Expectation with respect to the relevant probability measure.
$\mathbb{D}(P\ W)$	Kullback–Leibler divergence (relative entropy) between path measures.
$\lambda(t, x)$	Value-function/Lagrange multiplier; solves an HJB-type PDE.
$\varphi = e^{\lambda/\varepsilon}, \hat{\varphi} = \rho e^{-\lambda/\varepsilon}$	Hopf–Cole pair; solve backward/forward heat (Kolmogorov) equations.
$p^W(s, x; t, y)$	Heat kernel (Wiener transition density).

Reflected SB and boundary terms

$\mathbf{n}(x)$	Unit inward normal on $\partial\Omega$.
γ_t	Boundary local time (nondecreasing, increases only when $X_t \in \partial\Omega$).
No-flux BC	$\mathbf{n} \cdot (\varepsilon \nabla \rho - \rho(u + f)) = 0$ on $\partial\Omega$.
Neumann BC	$\mathbf{n} \cdot \nabla \varphi = 0, \mathbf{n} \cdot \nabla \hat{\varphi} = 0$ on $\partial\Omega$.

Differential operators and PDEs

$\nabla, \text{div}, \Delta$	Gradient, divergence, Laplacian (spatial).
∂_t	Time derivative.
Fokker–Planck	$\partial_t \rho + \nabla \cdot (\rho u) = \frac{\varepsilon}{2} \Delta \rho$ (variants with drift f).
HJB / HJ	$\partial_t \lambda + \frac{1}{2} \ \nabla \lambda\ ^2 + \langle \nabla \lambda, f \rangle = -\frac{\varepsilon}{2} \Delta \lambda$.
Kolmogorov (FB)	$\partial_t \varphi = -\nabla \varphi \cdot f - \frac{\varepsilon}{2} \Delta \varphi, \quad \partial_t \hat{\varphi} = -\nabla \hat{\varphi} \cdot f + \frac{\varepsilon}{2} \Delta \hat{\varphi}$.

Optimization and splitting

$\text{prox}_{\tau F}(z)$	Proximal operator $\arg \min_y \{F(y) + \frac{1}{2\tau} \ y - z\ ^2\}$.
SDR	Symmetric Douglas–Rachford splitting.
ADMM	Alternating Direction Method of Multipliers.
Primal–Dual	Chambolle–Pock-type primal–dual splitting.
JKO	Jordan–Kinderlehrer–Otto time-discrete gradient flow step in W_2 .
WPR	Wasserstein–proximal recursion (particle/point-cloud scheme).

Discretization (grids, FEM) and linear algebra

$\Delta t, \Delta x$	Time/space step sizes in finite-difference/staggered grids.
G_x, G_t, G_c	Staggered/centered node sets used for discrete divergence/interpolation.
div_h, Δ_h	Discrete divergence/Laplacian; $\Delta_h = A A^*$ in projections.
I, b, b^*	Interpolation operator; boundary extraction operator and its adjoint.
$V_h \subset H^1(\Omega)$	FEM space of continuous, piecewise-linear (P1) functions.
$\{\psi_i\}_{i=1}^{N_h}$	FEM nodal basis associated with mesh nodes $\{x_i\}$.

M	Mass matrix, $M_{ij} = \int_{\Omega} \psi_i \psi_j dx.$
L	Stiffness matrix, $L_{ij} = \int_{\Omega} \nabla \psi_i \cdot \nabla \psi_j dx.$
Δt (FEM time)	Time step in backward Euler: $(M + \frac{\varepsilon}{2} \Delta t L) u^{k+1} = M u^k.$
h	Mesh size; $h = \max_T \text{diam}(T).$
P	Number of time steps in FD sections (so that $\Delta t = 1/P$).
Id	Identity operator/matrix.
Miscellaneous (used in complexity/algorithms)	
n	Total number of grid DoFs (context-dependent, e.g. $n = N^2 P$).
T	Number of outer iterations for an algorithm (e.g. SDR/PD).
K	Iterations for an inner solver/root-finding (context-dependent).
$\mathcal{O}(\cdot)$	Landau notation for computational complexity.

Remark. Throughout the thesis we occasionally write $\Omega \equiv \mathcal{X}$ and $\partial\Omega \equiv \partial\mathcal{X}$. Symbols are (re)defined at first use; the table collates the most frequently used ones for quick reference.

Contents

Notation	v
1 Introduction	1
1.1 Literature Review	1
1.1.1 From Monge to Kantorovich	1
1.1.2 Static versus Dynamic Formulations	1
1.1.3 Schrödinger Bridge and Its OT Connection	2
1.2 Structures of the thesis	5
1.2.1 Chapter 2: Methods for solving Optimal Transport	5
1.2.2 Chapter 3: Schrödinger Bridge and Stochastic Optimal Transport	5
1.2.3 Chapter 4: Numerical Experiments	6
1.2.4 Chapter 5: Conclusion and Future Work	6
1.2.5 Chapter 6: Appendix	6
1.3 Brief methods summary for Optimal transport	6
2 Methods for solving Optimal Transport	8
2.1 Sinkhorn method for optimal transport problem	9
2.1.1 Principle of Sinkhorn method	9
2.1.2 Complexity of Sinkhorn Algorithm	10
2.2 Dynamic method for solving the Optimal Transport Problem	11
2.2.1 Transform the problem	11
2.2.2 Discretize the problem with Finite Difference Method	12
2.2.3 Solving the Discrete Problem with Symmetric Douglas-Rachford Splitting	14
2.2.4 Complexity Analysis for the SDR Algorithm.	16
2.2.5 Solving the discrete problem with Prime dual Chambolle-Pock Splitting Method	18
2.2.6 Modified L^2 -Wasserstein distance	19
3 Schrödinger bridge and Optimal Transport	20
3.1 Introduction	20
3.2 Background	20
3.3 Schrödinger System	23
3.4 Dynamic Schrödinger Bridge	24
3.5 Nonlinear prior Dynamic Schrödinger Bridge	26
3.5.1 Gradient Flow Dynamic Schrödinger Bridge	29

3.5.2	Proximal Recursion Algorithm	29
3.5.3	Sinkhorn Algorithm for Proximal Recursion	30
3.5.4	Proximal Recursion for Kolmogorov System	32
3.6	Schrödinger Bridge with Reflecting Boundary	34
3.6.1	Problem Formulation	34
3.6.2	Finite Element Method for Reflected Schrödinger Bridge . . .	36
3.6.3	Convergence Analysis and Complexity for FEM in Reflected Schrödinger Bridge	37
4	Experimental Results	39
4.1	Numerical example with simple mass splitting transport.	39
4.1.1	Dynamic transport method for mass splitting	39
4.1.2	Sinkhorn method for mass splitting	40
4.1.3	Modified Sinkhorn algorithm for Schrödinger bridge	42
4.1.4	Recursion method for the Schrödinger system in mass splitting	44
4.1.5	Comparison with three methods	46
4.2	Numerical example with Optimal Transport with Obstacles	47
4.2.1	One dimensional simple example	48
4.2.2	Two dimensional maze-like obstacle example	50
4.2.2.1	Method with Proximal Splitting Algorithm	51
4.2.2.2	Method with Reflected Schrödinger Bridge using Finite Elements	53
4.2.2.3	Comparison between SDR and RSB methods	59
4.3	Numerical example with Gradient Flow Schrödinger Bridge	61
5	Conclusion and Future Work	64
5.1	Methodological contributions.	64
5.2	Method Summary and Takeaways	65
5.3	Limitations and Future Work	66
6	Appendix	67
6.1	Proof of Proposition 2.2.2	67
6.2	Proof of Proposition 2.2.5	68
6.3	Details for Remark 2.2.6	68
6.4	Proof of Proposition 2.2.7	69
6.5	Details for the proof of equivalent problem 3.9	70
6.6	Proof of Proposition 3.4.2	71
6.7	Proof of Theorem 3.5.3	72
6.8	Derivation of problem 3.20	72
6.9	Proof of Theorem 3.6.1	73
6.10	Proof of Proposition 3.6.2	74
6.11	Proximal recursions for the reflected SBP	74
6.11.1	Reflected SDE simulation	75
6.11.2	Proximal recursions for the reflected SBP	75
6.11.3	Numerical explanation for bad performance of the proximal recursion	75

Chapter 1

Introduction

Optimal transport is a mathematical framework that deals with the problem of transporting mass from one distribution to another in an optimal way. It has applications in various fields, including economics, physics, and machine learning. We begin by defining the optimal transport problem and its mathematical formulation, followed by a discussion of its historical development and significance in modern mathematics.

1.1 Literature Review

1.1.1 From Monge to Kantorovich

The classical mass-transport problem dates back to Monge (1781), who asked for the least-cost relocation of a mass distribution (“moving soil”). A modern formulation was introduced by Kantorovich (1948), who relaxed Monge’s map-based problem to a linear program over transport plans. The resulting Monge–Kantorovich problem reads

$$\min_{\pi \in \Pi(\mu, \nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} c(x, y) d\pi(x, y), \quad (1.1)$$

where $\pi(\mu, \nu)$ denotes the set of couplings (transport plans) on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals μ and ν , and $c(x, y)$ is a prescribed cost.

Choosing $c(x, y) = \|x - y\|^p$ yields the p -Wasserstein geometry: the optimal value defines

$$W_p^p(\mu, \nu) = \min_{\pi \in \Pi(\mu, \nu)} \int \|x - y\|^p d\pi(x, y), \quad W_p(\mu, \nu) = (W_p^p(\mu, \nu))^{1/p},$$

which is a bona fide metric on the space of probability measures with finite p -th moments (Villani, 2021).

1.1.2 Static versus Dynamic Formulations

This problem has always been viewed as a static optimal transport problem, where the goal is to find the optimal transport plan that minimizes the transportation cost.

Consider the discrete setting of this problem, and refer to Peyré et al. (2019), the optimal transport problem can be formulated as a linear programming problem:

$$\min_{\pi \in \mathbb{R}^{n \times m}} \langle c, \pi \rangle \quad \text{s.t.} \quad \pi \mathbf{1}_m = a, \quad \pi^\top \mathbf{1}_n = b, \quad \pi \geq 0, \quad (1.2)$$

where c is a finite dimensional cost matrix, $\mathbf{1}_m$ and $\mathbf{1}_n$ are vectors of ones of appropriate dimensions, and a and b are the source and target marginal distributions, respectively. The solution to this problem gives the optimal transport plan that minimizes the transportation cost.

This problem can be solved using various numerical methods like Sinkhorn distance and the entropic regularization method. Adding a relative-entropy term $-\varepsilon H(\pi)$ to the cost in (1.2) produces a strictly convex problem solvable by Sinkhorn's matrix-scaling iterations, which converge geometrically and parallelise on modern hardware. As $\varepsilon \downarrow 0$ the solution concentrates toward OT while retaining numerical stability. While in *Benamou and Brenier* (2000), this static problem was first proved to be equivalent to a dynamic approach, which allows for the computation of the Wasserstein distance between two probability distributions by solving a time-dependent energy minimization problem with a continuity constraint:

$$\begin{aligned} \min_{(v, \rho) \in \mathcal{C}^0} \quad & \frac{1}{2} \int_{[0,1]^d} \int_0^1 \rho(x, t) \|v(x, t)\|^2 dt dx, \\ \mathcal{C}^0 = \Big\{ (v, \rho) \mid & \partial_t \rho + \operatorname{div}_x(\rho v) = 0, \quad \rho(x, 0) = \rho_0(x), \quad \rho(x, 1) = \rho_1(x) \Big\}. \end{aligned} \quad (1.3)$$

This dynamic approach provides a more flexible framework for solving the optimal transport problem, allowing for the incorporation of additional constraints and regularization techniques. In Haber et al. (2010), the authors using staggered grids to discretize the problem and reformulate the Lagrangian as a quadratic programming problem, and solved it by Shur complement techniques. Additionally, referring to Parikh et al. (2014), problem 1.3 can also be viewed through the lens of proximal algorithms. Papadakis et al. (2014) proposed proximal splitting methods based on staggered grid to solve the dynamic optimal transport problem by splitting 1.3 into subparts(details are shown in chapter 2), and then consider Douglas-Rachford (DR) splitting method proposed by Lions and Mercier (1979); Douglas and Rachford Jr (1956) to solve the problem. They also implement prime-dual (PD) splitting method proposed by Chambolle and Pock (2011) to solve the same problem. Their method enables efficient computation of the Wasserstein distance in high-dimensional spaces and can be extended to more complex scenarios, such as those involving time-varying distributions or additional physical constraints.

1.1.3 Schrödinger Bridge and Its OT Connection

The *Schrödinger bridge* (SB), posed by Schrödinger (1931–32), seeks the most likely evolution of a diffusive system whose initial and terminal densities are prescribed, which can be seen in Chetrite et al. (2021) and in Schrödinger (1932); it may be viewed as an entropically regularised dynamic OT Chen et al. (2021). Under Brownian prior dynamics, SB reduces exactly to Sinkhorn's iterations in continuous time

(also called *iterative proportional fitting*), which is first proposed by Sinkhorn (1964) and by Sinkhorn (1967). The SB problem is formulated as follows:

$$\begin{aligned} \min_{u_t} \mathbb{E} \left[\int_0^1 \frac{1}{2\varepsilon} \|u_t\|^2 dt \right], \\ \text{subject to } dX_t = u_t dt + \sqrt{\varepsilon} dW_t, \\ X_0 \sim \rho_0, \quad X_1 \sim \rho_1. \end{aligned} \tag{1.4}$$

Additionally, Dai Pra (1991) showed that the SB problem can be reformulated using Ito formulation into a stochastic control problem:

$$\begin{aligned} \min_{(v, \rho) \in \mathcal{C}^0} \quad & \frac{1}{2} \int_{[0,1]^d} \int_0^1 \rho(x, t) \|v(x, t)\|^2 dt dx, \\ \mathcal{C}^0 = \left\{ (v, \rho) \mid \partial_t \rho + \operatorname{div}_x(\rho v) = \frac{\varepsilon}{2} \Delta \rho, \quad \rho(x, 0) = \rho_0(x), \quad \rho(x, 1) = \rho_1(x) \right\}. \end{aligned} \tag{1.5}$$

It's worth noting that 1.3 can also be viewed as a fluid dynamics problem similar to 1.3, but with an additional diffusion term $\frac{\varepsilon}{2} \Delta \rho$ in the continuity equation. This diffusion term accounts for the stochastic nature of the underlying process, which is modeled as a Brownian motion with variance ε . The presence of this diffusion term allows for a more flexible and robust formulation of the optimal transport problem, as it can handle situations where the distributions being transported are not perfectly aligned or have some inherent uncertainty.

If we consider the optimality condition for this problem, we can obtain the following system of coupled partial differential equations:

$$\begin{aligned} \partial_t \lambda + \frac{1}{2} \|\nabla \lambda\|^2 + \frac{\varepsilon}{2} \Delta \lambda &= 0, \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \nabla \lambda) &= \frac{\varepsilon}{2} \Delta \rho. \end{aligned}$$

The first equation is called Hamiltonian-Jacobi-Bellman (HJB) equations, and the second one is continuity equation with diffusion term. The existence and uniqueness of solutions to these equations are proved in Beurling (1960), Fortet (1940), and Jamison (1975). The direct solve for this system is really challenging since it is a coupled system of nonlinear PDEs. However, by Hopf-Cole transformation Hopf (1950); Cole (1951),

$$\varphi = e^{\lambda/\varepsilon}, \quad \hat{\varphi} = \rho e^{-\lambda/\varepsilon},$$

these equations can be transformed into forward-backward heat equations,

$$\begin{aligned} \partial_t \varphi + \frac{\varepsilon}{2} \Delta \varphi &= 0, \\ \partial_t \hat{\varphi} - \frac{\varepsilon}{2} \Delta \hat{\varphi} &= 0, \end{aligned}$$

providing a powerful framework for analyzing the dynamics of the system. Recent work extends SB to nonlinear drifts and reflected SB. The following stochastic differential equation (SDE) given by

$$dX_t = (f + u_t) dt + \sqrt{\varepsilon} dW_t, \tag{1.6}$$

is an example of SDE with nonlinear drift, where f is a nonlinear drift term. This extension allows for the modeling of more complex systems and has applications in various fields, including robotics and swarm control. We can still consider the similar approach as before to reduce the problem into a system of forward-backward Kolmogorov partial differential equations:

$$\begin{aligned}\partial_t \varphi &= -\nabla \varphi \cdot f - \frac{\epsilon}{2} \Delta \varphi, \\ \partial_t \hat{\varphi} &= -\nabla \hat{\varphi} \cdot f + \frac{\epsilon}{2} \Delta \hat{\varphi}.\end{aligned}\tag{1.7}$$

In particular, Jordan et al. (1998) introduced the JKO variational time-stepping scheme, showing that the Fokker–Planck equation with gradient drift can be advanced stably by successive minimization of the free energy in the 2-Wasserstein metric. Building on this idea, Caluya and Halder (2019) proposed the *Wasserstein–proximal recursion* (WPR), which alternates a W_2 -proximal step with pointwise updates enforcing end-point marginals to solve Kolmogorov-type PDEs. Caluya and Halder (2021b) extended WPR to forward–backward Kolmogorov systems and proved convergence via a contraction in Hilbert’s projective metric; their scheme evolves weighted point clouds without spatial grids, enabling Schrödinger-bridge solvers for nonlinear stochastic dynamics and swarm-density control. In parallel, Carrillo et al. (2022) developed a primal–dual splitting framework for Wasserstein gradient flows: each time step is discretized by JKO and reformulated through the Benamou–Brenier dynamic optimal-transport characterization.

For reflected SB:

$$dX_t = f(X_t, t) dt + u(X_t, t) dt + \sqrt{2\epsilon} dW_t + \mathbf{n}(X_t) d\gamma_t,$$

Bushell (1973) shows that under certain condition(details shown in 3.6.1), the process X_t is Feller continuous and strongly Markov. Motivated by JKO steps, Caluya and Halder (2021a) also introduced the WPR methods in reflected Schrödinger bridge problem (RSBP):

$$\begin{aligned}\min_u \quad & \mathbb{E} \left[\int_0^1 \frac{1}{2} \|u(X_t, t)\|^2 dt \right] \\ \text{s.t. } & dX_t = f(X_t, t) dt + u(X_t, t) dt + \sqrt{2\epsilon} dW_t + \mathbf{n}(X_t) d\gamma_t, \\ & X_0 \sim \rho_0, \quad X_1 \sim \rho_1, \\ & X_t \in \overline{\Omega} \quad \text{for } t \in [0, 1],\end{aligned}\tag{1.8}$$

where $f : [0, 1] \times \overline{\Omega} \rightarrow \mathbb{R}^n$ is a smooth drift field and γ_t is the boundary *local time* associated with X_t —a continuous, nondecreasing process with $\gamma_0 = 0$ that increases only when $X_t \in \partial\Omega$. The map $\mathbf{n} : \partial\Omega \rightarrow \mathbb{S}^{n-1}$ denotes the inward unit normal vector). They transform the RSBP 1.8 into stochastic control problem subject to

reflecting boundary conditions:

$$\begin{aligned} \min_{(\rho, u) \in \mathcal{P}_2(\bar{\Omega}) \times \mathcal{U}} \quad & \int_0^1 \int_{\bar{\Omega}} \frac{1}{2} \|u(t, x)\|^2 \rho(t, x) dx dt \\ \text{s.t.} \quad & \partial_t \rho + \nabla \cdot (\rho(u + f)) = \varepsilon \Delta \rho \quad \text{in } \Omega \times (0, 1), \\ & \mathbf{n} \cdot (\varepsilon \nabla \rho - \rho(u + f)) = 0 \quad \text{on } \partial\Omega \times (0, 1), \\ & \rho(0, \cdot) = \rho_0, \quad \rho(1, \cdot) = \rho_1. \end{aligned} \quad (1.9)$$

Following similar steps as before, they reduced the problem into a system of forward-backward Kolmogorov partial differential equations with Neumann boundary conditions:

$$\begin{aligned} \partial_t \varphi &= -\nabla \varphi \cdot f - \frac{\epsilon}{2} \Delta \varphi, \\ \partial_t \hat{\varphi} &= -\nabla \hat{\varphi} \cdot f + \frac{\epsilon}{2} \Delta \hat{\varphi}. \\ \langle \nabla \varphi, n \rangle &= 0, \quad \langle \nabla \hat{\varphi}, n \rangle = 0 \end{aligned} \quad (1.10)$$

Then they follow the idea of WPR method to solve this system 1.10.

1.2 Structures of the thesis

In this dissertation, we explore various numerical methods for solving the optimal transport problem, including the Sinkhorn algorithm, dynamic energy minimization techniques, and Schrödinger bridge approaches. Besides the implementation of these algorithms, this dissertation novelly adapts reflected Schrödinger bridges for solving the optimal transport problem with obstacles and proposed both proximal splitting method and finite elements methods to solve the dynamic optimal transport problem with obstacles. Here is an outline of the dissertation:

1.2.1 Chapter 2: Methods for solving Optimal Transport

In this chapter, we will discuss classical numerical optimal transport methods, starting with the definition of optimal transport and then introducing various methods with detailed proof and analysis for solving optimal transport problems, including the Sinkhorn algorithm and dynamic energy minimization techniques like Symmetric Douglas-Rachford (SDR) splitting method and Primal-Dual (PD) splitting method.

1.2.2 Chapter 3: Schrödinger Bridge and Stochastic Optimal Transport

In this chapter, we will explore the Schrödinger bridge problem and its connection to optimal transport. We will discuss the mathematical formulation of the Schrödinger bridge problem, its relation to stochastic optimal transport problem and using this

approach to solve the optimal transport problem. This chapter will also cover numerical methods for solving nonlinear prior drift Schrödinger bridge problem and reflected Schrödinger bridge problem (RSBP), including using the Wasserstein-proximal recursion (WPR) method and finite element methods.

1.2.3 Chapter 4: Numerical Experiments

In this chapter, we will present numerical experiments to demonstrate the effectiveness of the methods discussed in previous chapters. We will compare the performance of different algorithms for solving optimal transport problems, including the Sinkhorn algorithm, dynamic energy minimization techniques, and Schrödinger bridge approaches with optimal mass splitting example. We will also showcase the application of reflected Schrödinger bridges for solving optimal transport problems with obstacles, mainly focusing on the comparison of the proximal splitting method and finite element method for solving the dynamic optimal transport problem with obstacles. Finally, we provide an example of gradient flow using Wasserstein proximal recursion (WPR) method.

1.2.4 Chapter 5: Conclusion and Future Work

In this chapter, we will summarize the main findings of the dissertation and discuss potential future research directions. We will highlight the strengths and limitations of the methods presented in previous chapters and suggest possible improvements and extensions for further exploration in the field of optimal transport and Schrödinger bridge problems.

1.2.5 Chapter 6: Appendix

In this chapter, we will provide supplementary material, including detailed proofs, additional explanations and one numerical realization result with Wasserstein proximal recursion(WPR) method for one Optimal transport problem with obstacles.

1.3 Brief methods summary for Optimal transport

In this section, we make a simple summary of methods used for OT. Generally speaking, Optimal Transport (OT) admits a *static* (Kantorovich) view and a *dynamic* (Benamou–Brenier) view. While the static view focuses more on the optimal transport plan between two distributions, the dynamic view considers the evolution of the continuity equation over time.

Entropic regularization of the static problem yields the *Sinkhorn* algorithm, which is extremely fast and a good surrogate for true OT on simple geometries, but it is not well suited to hard obstacles or OT variants with prior drift (gradient-flow structure). The dynamic formulation enforces the continuity equation and can be solved accurately by *proximal splitting* (e.g., SDR, ADMM, primal–dual), which flexibly incor-

porates constraints and deformations, though convergence is typically slower with only first order convergence rates.

The *Schrödinger Bridge (SB)* provides a probabilistic relaxation linked to entropic OT and the dynamic continuity equation. In PDE form, SB can be advanced by *finite elements* (forward/backward Kolmogorov equations), which preserves mass/positivity and handles complex boundaries, but requires repeatedly solving large sparse systems and becomes onerous in high dimensions. Alternatively, *Wasserstein proximal recursion* (particle/point-cloud) is mesh-free and lightweight, performing well when boundaries are simple and drifts are smooth, but it struggles to capture maze-like obstacles and reflection with high fidelity. Here is a summary figure of the relationship between these methods:

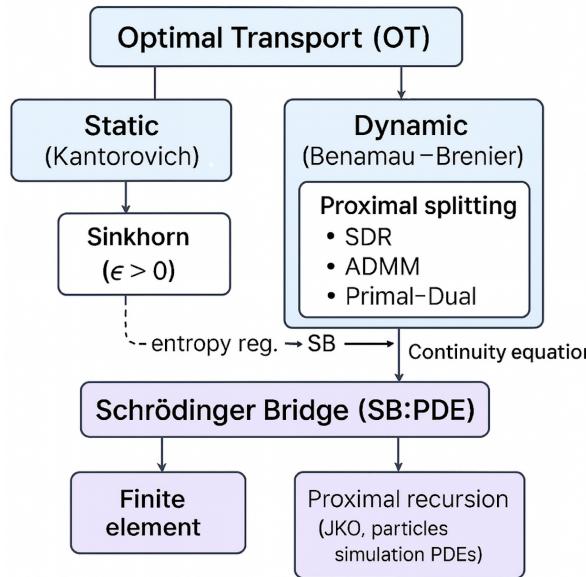


Figure 1.1: Summary of methods for Optimal Transport

Chapter 2

Methods for solving Optimal Transport

In this chapter, we will discuss classical numerical optimal transport methods. We will start with the definition of optimal transport and then introduce various methods for solving optimal transport problems.

Definition 2.0.1. (*Optimal Transport*) A valid transport map T is a map that pushes forward the measure $\rho_a(x) dx$ onto $\rho_b(x) dx$. In terms of the densities at positions a and b , this gives the constraint

$$\rho_a(x) = \rho_b(T(x)) |\det(\partial T(x))| \quad (2.1)$$

where $\partial T(x) \in \mathbb{R}^{d \times d}$ is the Jacobian matrix of T at x . This relation is often called the gradient equation. We denote by $\mathcal{T}(\rho_a, \rho_b)$ the set of all transport maps satisfying this constraint. An optimal transport map T then solves

$$\min_{T \in \mathcal{T}(\rho_a, \rho_b)} \int C(x, T(x)) \rho_a dx \quad (2.2)$$

where $C(x, y) \geq 0$ is the cost of sending x to y . In the case $C(x, y) = \|x - y\|^2$, the optimal value of (2.2) is called the L^2 -Wasserstein distance between the densities ρ_a and ρ_b .

We Consider a discretized problem: the assignment procedure can be described with a coupling matrix $\mathbf{P} \in \mathbb{R}_+^{n \times m}$, where $\mathbf{P}_{i,j}$ describes the amount of mass flowing from source i to target j . The matrix \mathbf{P} is a coupling matrix if it satisfies the following conditions:

- $\mathbf{P} \geq 0$ (non-negativity)
- $\mathbf{P}\mathbf{1}_m = \mathbf{a}$ (marginal constraint for sources)
- $\mathbf{P}^\top \mathbf{1}_n = \mathbf{b}$ (marginal constraint for targets)

where $\mathbf{1}_m$ and $\mathbf{1}_n$ are vectors of ones of appropriate dimensions, and \mathbf{a} and \mathbf{b} are the source and target marginal distributions, respectively. Since we use the discrete

setting, we express the change of mass from source i to target j as a matrix $\mathbf{P} \in \mathbb{R}^{n \times m}$. Thus, we have the following definition of the Kantorovich's optimal transport problem.

Definition 2.0.2. (*Kantorovich's optimal transport problem*) *The Kantorovich's optimal transport problem is defined as the following optimization problem:*

$$\begin{aligned} & \min_{\mathbf{P} \in \mathbb{R}_+^{n \times m}} \langle \mathbf{C}, \mathbf{P} \rangle \\ \text{s.t. } & \mathbf{P}\mathbf{1}_m = \mathbf{a} \\ & \mathbf{P}^\top \mathbf{1}_n = \mathbf{b} \end{aligned} \tag{2.3}$$

where $\mathbf{C} \in \mathbb{R}^{n \times m}$ is the cost matrix, $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product, and \mathbf{a}, \mathbf{b} are the source and target marginal distributions, respectively.

2.1 Sinkhorn method for optimal transport problem

We now consider the method to directly solve Kantorovich's optimal transport problem. It is obvious that (2.0.2) is not numerically stable since the result may be sensitive to small perturbations in the cost matrix \mathbf{C} or the marginal distributions \mathbf{a} and \mathbf{b} . To address this issue, we can introduce entropy regularization to the Kantorovich's optimal transport problem, called the Sinkhorn distance, which is a more stable and computationally efficient version of the optimal transport problem.

2.1.1 Principle of Sinkhorn method

Definition 2.1.1. (*Sinkhorn distance*) *The Sinkhorn distance is defined as the following optimization problem:*

$$\begin{aligned} & \min_{\mathbf{P} \in \mathbb{R}_+^{n \times m}} \langle \mathbf{C}, \mathbf{P} \rangle + \epsilon H(\mathbf{P}) \\ \text{s.t. } & \mathbf{P}\mathbf{1}_m = \mathbf{a} \\ & \mathbf{P}^\top \mathbf{1}_n = \mathbf{b} \end{aligned} \tag{2.4}$$

where $H(\mathbf{P}) = \sum_{i=1}^n \sum_{j=1}^m \mathbf{P}_{i,j} (\log(\mathbf{P}_{i,j}) - 1)$ is the entropy of the coupling matrix \mathbf{P} , and $\epsilon > 0$ is a regularization parameter.

Proposition 2.1.2. (*Convergence with ϵ*). *The unique solution \mathbf{P}_ϵ converges to the optimal transport plan as $\epsilon \rightarrow 0$. Additionally, if $\epsilon \rightarrow \infty$, the solution \mathbf{P}_ϵ converges to $\mathbf{ab}^T = (a_i b_j)_{i,j}$.*

The proof for this proposition is shown in Peyré et al. (2019), which demonstrating the convergence of the Sinkhorn distance to the optimal transport plan as the regularization parameter ϵ approaches zero. We now use Lagrange multipliers to solve the Sinkhorn distance problem. We introduce Lagrange multipliers $\mathbf{u} \in \mathbb{R}^n$ and $\mathbf{v} \in \mathbb{R}^m$ for the marginal constraints, and the Lagrangian function is given by:

$$\mathcal{L}(\mathbf{P}, \mathbf{u}, \mathbf{v}) = \langle \mathbf{C}, \mathbf{P} \rangle + \epsilon H(\mathbf{P}) - \langle \mathbf{u}, \mathbf{P}\mathbf{1}_m - \mathbf{a} \rangle - \langle \mathbf{v}, \mathbf{P}^\top \mathbf{1}_n - \mathbf{b} \rangle \tag{2.5}$$

Consider the first order optimality conditions, we have:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial P_{i,j}} &= C_{i,j} + \epsilon \log(P_{i,j}) - u_i - v_j = 0 \\ \Rightarrow P_{i,j} &= \exp\left(\frac{u_i + v_j - C_{i,j}}{\epsilon}\right) = \exp\left(\frac{u_i}{\epsilon}\right) \exp\left(\frac{v_j}{\epsilon}\right) \exp\left(-\frac{C_{i,j}}{\epsilon}\right)\end{aligned}$$

We rescale the variables \mathbf{u}, \mathbf{v} and define $\mathbf{K} = \exp\left(-\frac{C}{\epsilon}\right)$, then we have:

$$P_{i,j} = u_i v_j K_{i,j} \quad (2.6)$$

where $\mathbf{u} = \exp\left(\frac{\mathbf{u}}{\epsilon}\right)$ and $\mathbf{v} = \exp\left(\frac{\mathbf{v}}{\epsilon}\right)$. Substituting this expression into the marginal constraints, we obtain the following equations:

$$\begin{aligned}P\mathbf{1}_m &= \mathbf{u}\mathbf{v}^\top \mathbf{K}\mathbf{1}_m = \mathbf{u} \left(\sum_{j=1}^m v_j K_{i,j} \right) = \mathbf{a} \\ \Rightarrow \mathbf{u} &= \frac{\mathbf{a}}{\mathbf{K}\mathbf{v}}\end{aligned}$$

Similarly, for the second marginal constraint, we have:

$$\begin{aligned}P^\top \mathbf{1}_n &= \mathbf{v}\mathbf{u}^\top \mathbf{K}^\top \mathbf{1}_n = \mathbf{v} \left(\sum_{i=1}^n u_i K_{i,j} \right) = \mathbf{b} \\ \Rightarrow \mathbf{v} &= \frac{\mathbf{b}}{\mathbf{K}^\top \mathbf{u}}\end{aligned}$$

Thus, we can iteratively update \mathbf{u} and \mathbf{v} until convergence. The Sinkhorn algorithm is summarized as follows:

Algorithm 1 Sinkhorn Algorithm

- 1: **Input:** Cost matrix C , source marginal a , target marginal b , regularization parameter ϵ , tolerance τ
 - 2: Initialize $\mathbf{u} = \mathbf{1}_n$, $\mathbf{v} = \mathbf{1}_m$
 - 3: Compute $\mathbf{K} = \exp\left(-\frac{C}{\epsilon}\right)$
 - 4: **repeat**
 - 5: Update $\mathbf{u} = \frac{\mathbf{a}}{\mathbf{K}\mathbf{v}}$
 - 6: Update $\mathbf{v} = \frac{\mathbf{b}}{\mathbf{K}^\top \mathbf{u}}$
 - 7: **until** $\|P\mathbf{1}_m - \mathbf{a}\| < \tau$ and $\|P^\top \mathbf{1}_n - \mathbf{b}\| < \tau$
 - 8: Compute the optimal transport plan $P = \text{diag}(\mathbf{u}) \cdot \text{diag}(\mathbf{v}) \cdot \mathbf{K}$
 - 9: **return** P
-

2.1.2 Complexity of Sinkhorn Algorithm

In our implementation, the Sinkhorn algorithm operates on a grid of size $N \times N$, leading to $n = N^2$ discrete locations. The cost matrix $C \in \mathbb{R}^{n \times n}$ and the associated Gibbs kernel $K = \exp(-C/\epsilon)$ are both fully dense. As a result, each Sinkhorn iteration involves two matrix-vector multiplications of cost $\mathcal{O}(n^2)$.

Assuming a fixed number of iterations T , the overall computational complexity is therefore given by:

$$\mathcal{O}(n^2 \cdot T),$$

where T is the number of Sinkhorn updates required to reach a desired accuracy. In our experiments, we use a fixed iteration count $T = 500$, making the complexity effectively dominated by the quadratic cost per iteration.

This quadratic scaling with respect to n can be mitigated in practice using techniques such as kernel truncation, FFT-based convolutions (for translation-invariant costs), or multiscale entropic solvers.

2.2 Dynamic method for solving the Optimal Transport Problem

In this section, we will explore the dynamic methods to solve the optimal transport problem.

2.2.1 Transform the problem

We can obtain the density at any time $t \in [0, 1]$ with respect to initial density ρ_0 .

$$\rho(x, t) = \rho_0(T_t(x)) |\det(\partial T_t(x))|, \quad T_t = (1-t)\text{Id}_d + tT,$$

where $t \in [0, 1]$, T is the optimal transport map from ρ_0 to ρ_1 , Id_d is the d -dimensional identity, and $\partial T_t(x) \in \mathbb{R}^{d \times d}$ is the Jacobian of T_t at x .

To solve L^2 -Wasserstein distance, Benamou and Brenier (2000) proved that the following (non-convex) minimization problem over a density field $\rho(x, t) \in \mathbb{R}$ and a velocity field $v(x, t) \in \mathbb{R}^d$ satisfying the constraint \mathcal{C}^0 is equivalent to the optimal transport problem (2.2):

$$\begin{aligned} & \min_{(v, \rho) \in \mathcal{C}^0} \frac{1}{2} \int_{[0,1]^d} \int_0^1 \rho(x, t) \|v(x, t)\|^2 dt dx, \\ & \mathcal{C}^0 = \left\{ (v, \rho) \mid \begin{array}{l} \partial_t \rho + \text{div}_x(\rho v) = 0, \quad v(x, 0) = v(x, 1) = 0, \\ \rho(x, 0) = \rho_0(x), \quad \rho(x, 1) = \rho_1(x) \end{array} \right\}. \end{aligned} \tag{2.7}$$

We first introduce change of variables to transform the non-convex problem (2.7). We define the momentum field $m(x, t) = \rho(x, t)v(x, t)$ and the density field $\rho(x, t)$. The change of variables gives us the following convex optimization problem:

$$\min_{(m, \rho) \in \mathcal{C}} \mathcal{J}(m, \rho) = \int_{[0,1]^d} \int_0^1 J(m(x, t), \rho(x, t)) dt dx \tag{2.8}$$

where $J(m, \rho)$ is defined as follows:

$$J(m, \rho) = \begin{cases} \frac{\|m\|^2}{2\rho}, & \rho > 0, \\ 0, & (m, \rho) = (0, 0), \\ +\infty, & \text{otherwise,} \end{cases} \quad (2.9)$$

and the set \mathcal{C} is defined as follows:

$$\mathcal{C} = \left\{ (m, \rho) \mid \begin{array}{l} \partial_t \rho + \operatorname{div}_x(m) = 0, \quad m(0, t) = m(1, t) = 0, \\ \rho(x, 0) = \rho_0(x), \quad \rho(x, 1) = \rho_1(x) \end{array} \right\}. \quad (2.10)$$

Using similar method as interior point method, we use indicator function to transform the problem into a non-constraint problem. The indicator function is defined as follows:

$$\mathcal{I}(m, \rho) = \begin{cases} 0, & \text{if } (m, \rho) \in \mathcal{C}, \\ +\infty, & \text{otherwise.} \end{cases} \quad (2.11)$$

The overall optimization problem can now be expressed as:

$$\min_{(m, \rho)} \mathcal{J}(m, \rho) + \mathcal{I}(m, \rho) \quad (2.12)$$

2.2.2 Discretize the problem with Finite Difference Method

To solve the optimization problem, we discretize the space and time domains using finite difference methods. We define a grid in space and time, and approximate the derivatives using central difference to keep second order accuracy. Following Harlow and Welch Harlow et al. (1965), we introduce a uniform staggered mesh with spatial step $\Delta x = 1/N$ and temporal step $\Delta t = 1/P$. Define the two interleaved node sets as

$$\mathcal{G}_x = \left\{ (x_{i+\frac{1}{2}}, t_j) \mid x_{i+\frac{1}{2}} = (i + \frac{1}{2}) \Delta x, t_j = j \Delta t, -1 \leq i \leq N, 0 \leq j \leq P \right\}, \quad (2.13)$$

$$\mathcal{G}_t = \left\{ (x_i, t_{j+\frac{1}{2}}) \mid x_i = i \Delta x, t_{j+\frac{1}{2}} = (j + \frac{1}{2}) \Delta t, 0 \leq i \leq N, -1 \leq j \leq P \right\}. \quad (2.14)$$

The discrete unknowns are then collected as

$$U = (\bar{m}, \bar{\rho}) \in \mathcal{E}_h = \mathbb{R}^{\mathcal{G}_x} \times \mathbb{R}^{\mathcal{G}_t},$$

where $\bar{m}_{i,j}$ are the momentum values on \mathcal{G}_x and $\bar{\rho}_{i,j}$ the density values on \mathcal{G}_t . Then the time derivative is approximated as:

$$\partial_t \rho(x_i, t_j) \approx \frac{\rho(x_i, t_{j+1/2}) - \rho(x_i, t_{j-1/2})}{\Delta t}, \quad (2.15)$$

and the divergence of the momentum field m using a similar central difference approach:

$$\operatorname{div}_x(m)(x_i, t_j) \approx \frac{m(x_{i+1/2}, t_j) - m(x_{i-1/2}, t_j)}{\Delta x} \quad (2.16)$$

where Δx and Δt are the spatial and temporal step sizes, respectively. Additionally, we introduce the centered grid

$$\mathcal{G}_c = \{(x_i, t_j) \mid x_i = i \Delta x, t_j = j \Delta t, 0 \leq i \leq N, 0 \leq j \leq P\},$$

where $\Delta x = 1/N$ and $\Delta t = 1/P$. The discrete unknowns are then collected as

$$V = (m, \rho) \in \mathcal{E}_c = \mathbb{R}^{\mathcal{G}_c} \times \mathbb{R}^{\mathcal{G}_c},$$

Thus, we can interpolate the values of the momentum and density fields from the staggered grid to the centered grid, we define the interpolation operator as

$$\mathcal{I} : \mathcal{E}_h \rightarrow \mathbb{R}^{\mathcal{G}_c} \times \mathbb{R}^{\mathcal{G}_c} \quad (2.17)$$

$$\forall 0 \leq i \leq N, 0 \leq j \leq P, \quad \begin{cases} m_{i,j} = \frac{1}{2}(\bar{m}_{i-1,j} + \bar{m}_{i,j}), \\ \rho_{i,j} = \frac{1}{2}(\bar{\rho}_{i,j-1} + \bar{\rho}_{i,j}). \end{cases} \quad (2.18)$$

Additionally, we deal with the boundary conditions. We can rewrite $\partial_t \rho + \operatorname{div}_x(m) = 0$ as the space-time divergence operator $\operatorname{div}_h : \mathcal{E}_h \rightarrow \mathbb{R}^{\mathcal{G}_c}$ denoted as

$$\forall 0 \leq i \leq N, 0 \leq j \leq P, \quad (\operatorname{div}_h U)_{i,j} = \frac{\bar{m}_{i,j} - \bar{m}_{i-1,j}}{\Delta x} + \frac{\bar{\rho}_{i,j} - \bar{\rho}_{i,j-1}}{\Delta t}. \quad (2.19)$$

And the Dirichlet boundary conditions can be expressed as the boundary operator

$$b : \mathcal{E}_h \longrightarrow \underbrace{\mathbb{R}^{P+1} \times \mathbb{R}^{P+1}}_{\text{momentum}} \times \underbrace{\mathbb{R}^{N+1} \times \mathbb{R}^{N+1}}_{\text{density}},$$

defined by

$$b(\mathcal{I}(U)) = \left(\{m_{0,j}\}_{j=0}^P, \{m_{N,j}\}_{j=0}^P, \{\rho_{i,0}\}_{i=0}^N, \{\rho_{i,P}\}_{i=0}^N \right). \quad (2.20)$$

We then enforce the discrete boundary conditions

$$b(\mathcal{I}(U)) = b_0 := (\mathbf{0}, \mathbf{0}, \rho^0, \rho^1),$$

where $\rho^0, \rho^1 \in \mathbb{R}^{N+1}$ are the prescribed initial and final densities.

Finally, we can express the optimization problem in a discrete form. The discrete version of the optimization problem is given by:

$$\min_{U \in \mathcal{E}_h} \mathcal{J}_h(U) + \mathcal{I}_h(U) \quad (2.21)$$

where $\mathcal{J}_h : \mathcal{E}_h \rightarrow \mathbb{R}$ is the discrete cost functional defined as

$$\mathcal{J}_h(U) = \sum_{(x_i, t_j) \in \mathcal{G}_c} J(\mathcal{I}(U)(x_i, t_j)) \Delta x \Delta t \quad (2.22)$$

and $\mathcal{I}_h : \mathcal{E}_h \rightarrow \mathbb{R}$ is the discrete indicator function defined as

$$\mathcal{I}_h(U) = \begin{cases} 0, & \text{if } U \in \mathcal{C}_h, \\ +\infty, & \text{otherwise,} \end{cases} \quad (2.23)$$

where \mathcal{C}_h is the discrete version of the constraint set \mathcal{C} defined as

$$\mathcal{C}_h = \{ U \in \mathcal{E}_h \mid \operatorname{div}_h(U) = 0, b(\mathcal{I}(U)) = b_0 \}. \quad (2.24)$$

2.2.3 Solving the Discrete Problem with Symmetric Douglas-Rachford Splitting

To solve the discrete optimization problem, we can use the Symmetric Douglas-Rachford Splitting method. This method is effective for solving convex optimization problems with separable structures. We can decompose the problem into two subproblems: one for the momentum field m and another for the density field ρ . The Symmetric Douglas-Rachford Splitting algorithm iteratively updates the variables m and ρ while enforcing the constraints. The updates are performed as follows:

Algorithm 2 Symmetric Douglas-Rachford Splitting for Optimal Transport

- 1: **Input:** Initial guess $W^{(0)} = (m^{(0)}, \rho^{(0)})$, step size $\tau > 0$, number of iterations K

2: **for** $k = 0$ to $K - 1$ **do**

3: $W^{(k+1/2)} = W^{(0)} + w_1 \text{prox}_{\tau J}(W^{(k)}) + w_2 \text{prox}_{\tau \mathcal{I}_h}(W^{(k)})$, (2.25)

4: $W^{(k+1)} = W^{(0)} + \mu(2W^{(k+1/2)} - W^{(k)} - \text{prox}_{\tau J}(W^{(k)}) - \text{prox}_{\tau \mathcal{I}_h}(W^{(k)}))$ (2.26)

5: Update the variables:

$U^{(k+1)} = U^{(k)} + \mu(W^{(k+1/2)} - U^{(k)})$ (2.27)

6: **end for**

7: **Output:** Final solution $U^{(K)} = (m^{(K)}, \rho^{(K)})$

We then discuss the proximal operator for the cost functional J and the indicator function \mathcal{I} . The proximal operator is defined as follows:

Definition 2.2.1. The proximal operator of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined as

$$\text{prox}_{\tau f}(x) = \arg \min_{y \in \mathbb{R}^n} \left(\tau f(y) + \frac{1}{2} \|y - x\|^2 \right) \quad (2.28)$$

where $\tau > 0$ is a parameter that controls the step size.

Using this definition, we can derive the proximal operator for the cost functional J as follows:

Proposition 2.2.2. The proximal operator for the cost functional J is given by

$$\text{prox}_{\tau J}(m, \rho) = \begin{cases} \frac{\bar{\rho}m}{\bar{\rho}+2\tau}, & \text{if } \bar{\rho} > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (2.29)$$

where $\bar{\rho}$ is the largest real root of third order polynomial equation in X :

$$P(X) \equiv (X - \bar{\rho})(X + 2\tau)^2 - \tau \|m\|^2 = 0. \quad (2.30)$$

Additionally, for indicate function, we can divide the space into two parts, one is the feasible set with divergence form and the other is the set with Dirichlet boundary conditions. The indicator function \mathcal{I}_s satisfying divergence form is defined as follows:

Definition 2.2.3. *The indicator function \mathcal{I}_s is defined as*

$$\mathcal{I}_s(U) = \begin{cases} 0, & \text{if } U \in \mathcal{C}, \\ \infty, & \text{otherwise,} \end{cases} \quad (2.31)$$

where \mathcal{C} is the set of feasible pairs $U = (\bar{m}, \bar{\rho}) \in \mathcal{E}_h$ that satisfy $\operatorname{div} U = 0$.

And the indicator function \mathcal{I}_c satisfying Dirichlet boundary conditions is defined as follows:

Definition 2.2.4. *The indicator function \mathcal{I}_c is defined as*

$$\mathcal{I}_c(U) = \begin{cases} 0, & \text{if } b(\mathcal{I}(U)) = b_0, \\ \infty, & \text{otherwise,} \end{cases} \quad (2.32)$$

where b_0 is the prescribed boundary condition defined in (2.20).

Proposition 2.2.5. *The proximal operator for the indicator function \mathcal{I}_s is given by*

$$\operatorname{prox}_{\tau \mathcal{I}_s}(U) = (\operatorname{Id} - A^* \Delta^{-1} A) U \quad (2.33)$$

where A is the divergence operator defined in (2.19) and Δ is the discrete Laplacian operator.

Remark 2.2.6. *We give a detailed explanation why $\Delta = AA^*$ is the discrete Laplacian operator and how \mathcal{I}_s is calculated numerically. Details can be found in Appendix 6.3.*

We then discuss the proximal operator for the indicator function \mathcal{I}_c satisfying Dirichlet boundary conditions.

Proposition 2.2.7. *The proximal operator for the indicator function \mathcal{I}_c is given by*

$$\operatorname{prox}_{\tau \mathcal{I}_c}(U) = U - \mathcal{I}^* b^* (b \mathcal{I} \mathcal{I}^* b^*)^{-1} (b(\mathcal{I}(U)) - b_0), \quad (2.34)$$

where b^* is the adjoint operator of the boundary operator b defined in (2.20).

Remark 2.2.8. *We now give a detailed explanation of how \mathcal{I}_c is calculated numerically. Given any $U \in \mathcal{E}_h$ and corresponding $V = \mathcal{I}(U) \in \mathcal{E}_c$, for each kind of boundary condition in $m(0, t) = m(1, t) = 0$ and $\rho(x, 0) = \rho^0, \rho(x, 1) = \rho^1$, we only need to deal with the start and end points of the momentum and density fields. Thus, for momentum boundary condition, we set*

$$A \begin{pmatrix} \bar{m} \\ m \end{pmatrix} = \begin{pmatrix} \mathcal{I} & -I_n \\ 0 & e_1^T \\ 0 & e_n^T \end{pmatrix} \begin{pmatrix} \bar{m} \\ m \end{pmatrix} = \begin{pmatrix} 0_n \\ 0 \\ 0 \end{pmatrix} \quad (2.35)$$

And for density boundary condition, we set

$$A \begin{pmatrix} \bar{\rho} \\ \rho \end{pmatrix} = \begin{pmatrix} \mathcal{I} & -I_n \\ 0 & e_1^T \\ 0 & e_n^T \end{pmatrix} \begin{pmatrix} \bar{\rho} \\ \rho \end{pmatrix} = \begin{pmatrix} 0_n \\ \rho^0 \\ \rho^1 \end{pmatrix} \quad (2.36)$$

where e_i is the i -th standard basis vector in \mathbb{R}^n . Thus, combine the momentum and density boundary conditions, we can write the general form as:

$$A \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} \mathcal{I} & -I_n \\ 0 & e_1^T \\ 0 & e_n^T \end{pmatrix} \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} 0_n \\ a \\ b \end{pmatrix} = g, \quad (2.37)$$

where a and b are the prescribed boundary conditions for momentum and density fields, respectively. The proximal operator for the indicator function \mathcal{I}_c can then be computed as the projection onto $Ax = g$:

$$\text{prox}_{\tau\mathcal{I}_c}(U) = (\text{Id} - A^*(AA^*)^{-1}A)U + A^*(AA^*)^{-1}g, \quad (2.38)$$

2.2.4 Complexity Analysis for the SDR Algorithm.

For this algorithm, we consider the 3-D analysis (containing x, y, t three dimensions) as the example. We have three proximal operators to compute:

- Proximal operator for cost function: $\text{prox}_{\tau J_\beta}(m, \rho)$.
- Proximal operator for divergence form: $\text{prox}_{\tau\mathcal{I}_s}(U)$.
- Proximal operator for Dirichlet boundary condition: $\text{prox}_{\tau\mathcal{I}_c}(U)$.

The complexity of each proximal operator is as follows:

- **Proximal operator for cost function:** The procedure is mainly dependent on the root-finding algorithm for the equation

$$P(X) = X^{1-\beta}(X - \bar{\rho})(X^\beta + 2\tau)^2 - \tau\beta\|m\|^2 = 0. \quad (2.39)$$

When $\beta = 1$, the equation becomes a cubic polynomial equation, which can be solved pointwise in $\mathcal{O}(N^2P)$ time, where $N \times N$ is the spatial grid size and P is the number of time steps. Let $n = N^2P$, the total complexity is $\mathcal{O}(n)$.

For $\beta \in (0, 1)$, Newton's method is used to find the root at each grid point. Each iteration involves elementwise operations of complexity $\mathcal{O}(n)$, and assuming K iterations are needed for convergence, the total complexity becomes

$$\boxed{\mathcal{O}(Kn)}.$$

- **Proximal operator for divergence form:** The projection onto the divergence-free constraint is implemented by solving a Neumann boundary Poisson equation. It consists of two main steps. First, compute the divergence $\nabla \cdot u$ using finite differences in each spatial direction. This is a local operation and requires $\mathcal{O}(n)$ time, where $n = N^2P$ denotes the total number of grid points in the (x, y, t) domain.

Then, solve the Poisson equation $\Delta p = -\nabla \cdot u$ with Neumann boundary conditions using a discrete cosine transform (DCT) method. Both the forward and inverse DCTs are performed over the 3D grid, yielding a computational complexity of $\mathcal{O}(n \log n)$.

Hence, the total complexity for the proximal operator associated with the divergence constraint is

$$\boxed{\mathcal{O}(n \log n)}.$$

- **Proximal operator for Dirichlet boundary condition:** This projection step enforces interpolation consistency between staggered variables under fixed Dirichlet boundary values. The implementation applies a dense projection matrix $B \in \mathbb{R}^{(2N+1) \times (2N+1)}$ to k column vectors, where N is the number of grid points along the projection direction, and k is the number of independent slices in the orthogonal directions. Take the horizontal direction as an example, the complexity is $\mathcal{O}(N^3 P)$.

For a 3D grid of size $N \times N \times P$, the projection is applied separately along each of the three coordinate directions. Each direction contributes a computational cost as follows:

- Horizontal direction (x): $\mathcal{O}(N^3 P)$,
- Vertical direction (y): $\mathcal{O}(N^3 P)$,
- Temporal direction (t): $\mathcal{O}(P^2 N^2)$.

Summing over all three directions, the total complexity is

$$\mathcal{O}(N^3 P + N^3 P + P^2 N^2) = \mathcal{O}(N^3 P + P^2 N^2).$$

Factoring this, we obtain

$$\mathcal{O}(N^3 P + P^2 N^2) = \mathcal{O}(N^3 P + P^2 N^2).$$

Assuming a near-cubic domain with $N \approx P$, the complexity simplifies to

$$\boxed{\mathcal{O}(n^{4/3})}.$$

We can summarize the total complexity of the SDR algorithm as follows:

$$\mathcal{O}(T(Kn + n \log n + n^{4/3})), \quad (2.40)$$

where T is the number of iterations for the outer loop of the SDR algorithm, K is the number of iterations for the root-finding algorithm in the proximal operator for cost function, and $n = N^2P$ is the total number of grid points in the 3D domain. This indicates that the algorithm is efficient for moderate values of K and n , especially when the number of iterations is not excessively large.

2.2.5 Solving the discrete problem with Prime dual Chambolle-Pock Splitting Method

To solve the discrete optimization problem, we can also use the Prime dual Chambolle-Pock Splitting method. This method is effective for solving convex optimization problems with separable structures. Since the minimization has the form of

$$\min_{U \in \mathcal{E}_h} \mathcal{J}(\mathcal{I}(U)) + \mathcal{I}_h(U) \quad (2.41)$$

We assign $G_2 = \mathcal{J}$, $A = \mathcal{I}$ and $G_1 = \iota_C$. Thus, the iterations compute a sequence $(U^{(\ell)}, \Upsilon^{(\ell)}, V^{(\ell)}) \in \mathcal{E}_s \times \mathcal{E}_s \times \mathcal{E}_c$ of variables from an initial $(\Upsilon^{(0)}, V^{(0)})$ with the following steps:

Algorithm 3 Prime dual Chambolle-Pock Splitting for Optimal Transport

- 1: **Input:** Initial guess $U^{(0)}$, step size $\tau > 0$, number of iterations K
- 2: **for** $k = 0$ to $K - 1$ **do**
- 3: Compute the proximal operator for G_2^* :

$$V^{(\ell+1)} = \text{Prox}_{\sigma G_2^*} (V^{(\ell)} + \sigma \mathcal{I} \Upsilon^{(\ell)}) \quad (2.42)$$

- 4: Compute the proximal operator for G_1 :

$$U^{(\ell+1)} = \text{Prox}_{\tau G_1} (U^{(\ell)} - \tau \mathcal{I}^* V^{(\ell+1)}) \quad (2.43)$$

- 5: Update the variables:

$$\Upsilon^{(\ell+1)} = U^{(\ell+1)} + \theta (U^{(\ell+1)} - U^{(\ell)}) \quad (2.44)$$

- 6: **end for**
 - 7: **Output:** Final solution $U^{(K)}$
-

We then discuss the proximal operator for G_2^* and G_1 . The dual proximal operator is defined as follows:

Definition 2.2.9. *The dual proximal operator of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined using the Moreau's identity as follows:*

$$\forall w \in \mathcal{H}, \quad \text{Prox}_{\gamma F^*}(w) = w - \gamma \text{Prox}_{F/\gamma}(w/\gamma), \quad (2.45)$$

where $\gamma > 0$ is a parameter that controls the step size.

Remark 2.2.10. *For numerically solving the prime dual problem, since the proximal operator for G_2^* is hard to obtain directly, we can modify the boundary condition with $b(\mathcal{I}(U)) = b_0$ into G_2^* function, this is to say, we consider $\mathcal{J}(V)$, we can show that the $\text{Prox}_{\frac{1}{\tau} J} \left(\frac{V^{(\ell)} - \tau \mathcal{I}^* V^{(\ell+1)} + b_0/2}{\tau} \right) - b_0/2$ is the same as the original one. Thus, the proximal operator for G_2^* is given by our dual proximal operator and the modified boundary condition. And the proximal operator for G_1 is given by the staggered grid divergence projection:*

$$\text{Prox}_{\tau G_1}(U) = U - \nabla \Delta^{-1} \text{div} U \quad (2.46)$$

2.2.6 Modified L^2 -Wasserstein distance

According to A new class of transport distances between measures, we can extend the L^2 -Wasserstein distance to a modified version that incorporates parameter β . The modified L^2 -Wasserstein distance is defined as follows:

$$\begin{aligned} \min_{(v, \rho) \in \mathcal{C}^0} \quad & \frac{1}{2} \int_{[0,1]^d} \int_0^1 \rho^\beta(x, t) \|v(x, t)\|^2 dt dx, \\ \mathcal{C}^0 = \left\{ (v, \rho) \mid \right. & \partial_t \rho + \operatorname{div}_x(\rho^\beta v) = 0, \quad v(x, 0) = v(x, 1) = 0, \\ & \left. \rho(x, 0) = \rho_0(x), \quad \rho(x, 1) = \rho_1(x) \right\}. \end{aligned} \quad (2.47)$$

This modified distance is a generalization of the L^2 -Wasserstein distance, where β controls the degree of regularization applied to the density ρ . When $\beta = 1$, it reduces to the standard L^2 -Wasserstein distance, and when $\beta \approx 0$, it approximates the H^{-1} Sobolev semi-norm over density, which may reflect the mass dissipation in the transport process.

For numerical computation, we extend $\mathcal{J}(\mathcal{I}(U))$ to the following functional:

$$\mathcal{J}_\beta^w(\mathcal{I}(U)) = \sum_{k \in \mathcal{G}_c} w_k J_\beta(m_k, f_k) \quad (2.48)$$

where w_k is the weight of the k -th group, and $J_\beta(m_k, f_k)$ is the cost function defined as:

$$J_\beta(m, \rho) = \begin{cases} \frac{\|m\|^2}{2\rho^\beta} & \text{if } f > 0 \\ 0 & \text{if } (m, \rho) = (0, 0) \\ +\infty & \text{otherwise} \end{cases} \quad (2.49)$$

The proximal operator of the functional $\mathcal{J}_\beta^w(\mathcal{I}(U))$ is defined as: following similar steps, we can easily derive the proximal operator of the functional $\mathcal{J}_\beta^w(\mathcal{I}(U))$.

Proposition 2.2.11. (*Proximal operator of $\mathcal{J}_\beta^w(\mathcal{I}(U))$*) *The proximal operator of the functional $\mathcal{J}_\beta^w(\mathcal{I}(U))$ is given by:*

$$\operatorname{prox}_{\tau J_\beta}(m, \rho) = \begin{cases} \frac{\bar{\rho}^\beta m}{\bar{\rho}^\beta + 2\tau}, & \text{if } \bar{\rho} > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (2.50)$$

where $\bar{\rho}$ is the largest real root of third order polynomial equation in X :

$$P(X) = X^{1-\beta}(X - \bar{\rho})(X^\beta + 2\tau)^2 - \tau\beta\|m\|^2 = 0. \quad (2.51)$$

Chapter 3

Schrödinger bridge and Optimal Transport

3.1 Introduction

We now introduce the Schrödinger bridge to solve the optimal transport problem. The Schrödinger bridge problem is a stochastic control problem that can be solved via its equivalent formulations:

- As a *minimum relative entropy* problem on path space.
- As a *stochastic control* problem with the Fokker–Planck (forward Kolmogorov) equation.
- As an *entropic regularization* of the static optimal transport problem (Sinkhorn).
- As a *fluid-dynamic* variational problem with diffusion.

3.2 Background

In this section, we introduce the foundational concepts necessary for understanding the Schrödinger bridge and its connections to optimal transport.

Definition 3.2.1 (Marginal and Joint Densities). *Let P be a probability measure on $\Omega = C([0, 1]; \mathbb{R}^n)$ with canonical process $X_t(\omega) = \omega(t)$. Define*

$$\rho_0^P(x) dx = P\{X_0 \in dx\}, \quad \rho_1^P(y) dy = P\{X_1 \in dy\},$$

and the joint endpoint density

$$\rho_{01}^P(x, y) dx dy = P\{X_0 \in dx, X_1 \in dy\},$$

so that $\int \rho_{01}^P(x, y) dy = \rho_0^P(x)$ and $\int \rho_{01}^P(x, y) dx = \rho_1^P(y)$.

Definition 3.2.2 (Transition Kernel and Static Couplings). *We define the Transition kernel for mass Transition as*

$$p(s, x; t, y) := W_\rho\{X_t \in dy \mid X_s = x\}/dy,$$

satisfies $\rho_{01}(x, y) = \rho_0(x)p(0, x; 1, y)$. Specially, for the Wiener prior W_ρ with initial density $\rho(x)$, the transition kernel satisfies the heat kernel with variance ε :

$$p^W(s, x; t, y) := \frac{1}{(4\pi\varepsilon)^{n/2}(t-s)} \exp\left(-\frac{\|x-y\|^2}{4\varepsilon(t-s)}\right).$$

The set of static couplings with prescribed marginals ρ_0, ρ_1 is

$$\Pi(\rho_0, \rho_1) = \{\pi(x, y) \geq 0 : \int \pi(x, y) dy = \rho_0(x), \int \pi(x, y) dx = \rho_1(y)\}.$$

Definition 3.2.3 (Disintegration into Bridge Measures). *Based on Radon Nikodym theory and disintegration, any $P \in \mathcal{D}(\rho_0, \rho_1)$ (path measures with $X_0 \sim \rho_0$, $X_1 \sim \rho_1$) admits*

$$P(d\omega) = \rho_{01}^P(x, y) dx dy \otimes P_x^y(d\omega), \quad P_x^y(A) = P[\omega \in A \mid X_0 = x, X_1 = y].$$

Similarly, W_x^y is the Brownian bridge under W . Thus ρ_{01}, P_x^y encode the static coupling and conditioned path law.

Based on these definitions, paths $\omega \in \Omega$ represent continuous evolutions, W encodes the unbiased Brownian prior, and W_x^y, P_x^y describe the “bridge” or conditioned evolution between fixed endpoints.

Definition 3.2.4 (Relative Entropy (Kullback–Leibler Divergence)). *For two measures $P \ll Q$ on Ω , the relative entropy is*

$$D(P\|Q) = \mathbb{E}_P[\log \frac{dP}{dQ}] = \int_{\Omega} \log\left(\frac{dP}{dQ}(\omega)\right) P(d\omega).$$

It quantifies the “information cost” to go from the prior Q to the posterior P .

Definition 3.2.5 (Static Schrödinger Bridge). *Equivalently, the static formulation is*

$$\min_{P \in \mathcal{D}(\rho_0, \rho_1)} D(P\|W),$$

where $\mathcal{D}(\rho_0, \rho_1)$ are path measures with marginals ρ_0, ρ_1 at times 0, 1. The optimal P^* is called the Schrödinger bridge.

By the Radon–Nikodym theorem and disintegration, one can write:

$$\begin{aligned} D(P\|W) &= \mathbb{E}_P\left[\log \frac{dP}{dW}\right] = \iint \left[\log \frac{\rho_{01}^P(x, y)}{\rho_{01}^W(x, y)}\right] \rho_{01}^P(x, y) dx dy \\ &\quad + \iiint \left(\log \frac{dP_x^y}{dW_x^y}\right) dP_x^y \rho_{01}^P(x, y) dx dy. \end{aligned}$$

The second term vanishes for the Schrödinger bridge since one takes $P_x^y = W_x^y$, reducing to a static entropic optimal transport on (x, y) :

$$\min_{P \in \mathcal{D}(\rho_0, \rho_1)} - \iint [\log \rho_{01}^W(x, y)] \rho_{01}(x, y) dx dy + \iint [\log \rho_{01}(x, y)] \rho_{01}(x, y) dx dy \quad (3.1)$$

Considering Wiener Measure, we can express $\rho_{01}^W(x, y) = p^W(0, x; 1, y)\rho(x)$ where $p^W(0, x; 1, y)$ is the transition kernel of the Wiener measure, and we use the fact that $\int \rho_{01}(x, y) dy = \rho_0(x)$, we have property

$$\begin{aligned} \iint [\log \rho_{01}^W(x, y)] \rho_{01}(x, y) dx dy &= \iint [\log p^W(0, x; 1, y)] \rho_{01}(x, y) dx dy \\ &\quad + \int [\log \rho_0(x)] \rho_0(x) dx. \end{aligned}$$

After removing the constant term, the Schrödinger bridge problem is equivalent to:

$$\begin{aligned} \min_{P \in \mathcal{D}(\rho_0, \rho_1)} & - \iint \left[\log \frac{1}{(4\pi\varepsilon)^{n/2}} \exp\left(-\frac{\|x-y\|^2}{4\varepsilon}\right) \right] \rho_{01}(x, y) dx dy \\ &+ \iint [\log \rho_{01}(x, y)] \rho_{01}(x, y) dx dy. \end{aligned} \quad (3.2)$$

This is also equivalent to the problem:

$$\min_{P \in \mathcal{D}(\rho_0, \rho_1)} \iint \left[\frac{\|x-y\|^2}{2} + \varepsilon \log \rho_{01}(x, y) \right] \rho_{01}(x, y) dx dy. \quad (3.3)$$

This formulation highlights the balance between the quadratic cost of transporting mass and the entropic regularization term, controlled by ε .

Combined with the constraints that P must have marginals ρ_0 and ρ_1 , this leads to a variational problem that can be solved using various numerical methods, such as the Fortet–IPF algorithm or Sinkhorn iterations.

We write the entropic-regularized OT problem

$$\min_{\rho_{01} \in \Pi(\rho_0, \rho_1)} \iint \left[\frac{\|x-y\|^2}{2} + \varepsilon \ln \rho_{01}(x, y) \right] \rho_{01}(x, y) dx dy$$

as a constrained variational problem. Introduce Lagrange multipliers $\lambda(x)$ and $\mu(y)$ to enforce $\int \rho_{01}(x, y) dy = \rho_0(x)$ and $\int \rho_{01}(x, y) dx = \rho_1(y)$. Define the augmented functional

$$\begin{aligned} \mathcal{L}[\rho_{01}, \lambda, \mu] &= \iint \left[\frac{1}{2} \|x-y\|^2 + \varepsilon \ln \rho_{01}(x, y) \right] \rho_{01}(x, y) dx dy \\ &\quad + \int \lambda(x) \left(\int \rho_{01}(x, y) dy - \rho_0(x) \right) dx \\ &\quad + \int \mu(y) \left(\int \rho_{01}(x, y) dx - \rho_1(y) \right) dy. \end{aligned} \quad (3.4)$$

We vary $\rho_{01} \mapsto \rho_{01} + \delta\rho$ with $\int \delta\rho dy = \int \delta\rho dx = 0$. Then

$$\frac{d}{d\varepsilon} \mathcal{L}[\rho_{01} + \varepsilon \delta\rho] \Big|_{\varepsilon=0} = \iint \left[\frac{1}{2} \|x - y\|^2 + \varepsilon(1 + \ln \rho_{01}^*) + \lambda(x) + \mu(y) \right] \delta\rho(x, y) dx dy.$$

Setting this to zero for all admissible $\delta\rho$ gives the Euler–Lagrange condition

$$\frac{1}{2} \|x - y\|^2 + \varepsilon(1 + \ln \rho_{01}^*(x, y)) + \lambda(x) + \mu(y) = 0.$$

Solving for ρ_{01}^* yields

$$\rho_{01}^*(x, y) = \exp\left(-1 - \frac{1}{\varepsilon} \frac{\|x - y\|^2}{2} - \frac{\lambda(x) + \mu(y)}{\varepsilon}\right), \quad (3.5)$$

and imposing the marginal constraints recovers the Schrödinger system.

3.3 Schrödinger System

We can derive the Schrödinger system from the variational problem. Reconsider the Lagrange multipliers $\lambda(x)$ and $\mu(y)$ in the augmented functional:

$$\begin{aligned} \mathcal{L}[\rho_{01}, \lambda, \mu] &= \iint \left[\log \frac{\rho_{01}^P(x, y)}{\rho_{01}^W(x, y)} \right] \rho_{01}^P(x, y) dx dy \\ &\quad + \int \lambda(x) \left(\int \rho_{01}(x, y) dy - \rho_0(x) \right) dx \\ &\quad + \int \mu(y) \left(\int \rho_{01}(x, y) dx - \rho_1(y) \right) dy. \end{aligned} \quad (3.6)$$

we perturb $\rho_{01} \mapsto \rho_{01} + \varepsilon h$ with $\int h(x, y) dy = 0$ and $\int h(x, y) dx = 0$, and compute

$$\frac{d}{d\varepsilon} \mathcal{L}[\rho_{01} + \varepsilon h] \Big|_{\varepsilon=0} = \iint \left[1 + \ln \frac{\rho_{01}(x, y)}{\rho_{01}^W(x, y)} + \lambda(x) + \mu(y) \right] h(x, y) dx dy.$$

Requiring this to vanish for all admissible h yields the pointwise stationarity condition

$$1 + \ln \frac{\rho_{01}^*(x, y)}{\rho_{01}^W(x, y)} + \lambda(x) + \mu(y) = 0.$$

Since $\rho_{01}^W(x, y) = \rho_0(x) p^W(0, x; 1, y)$, we may rewrite

$$\ln \frac{\rho_{01}^*(x, y)}{p^W(0, x; 1, y)} = \ln \rho_{01}^*(x, y) - \ln \rho_0(x) - \ln p^W(0, x; 1, y),$$

so the above becomes

$$1 + \ln \rho_{01}^*(x, y) - \ln \rho_0(x) - \ln p^W(0, x; 1, y) + \lambda(x) + \mu(y) = 0.$$

Thus,

$$\frac{\rho_{01}^*(x, y)}{p^W(0, x; 1, y)} = \exp[-1 - \lambda(x) - \mu(y) + \ln \rho_0(x)] = \hat{\varphi}(x) \varphi(y),$$

and finally

$$\rho_{01}^*(x, y) = \hat{\varphi}(x) p^W(0, x; 1, y) \varphi(y),$$

together with the marginal constraints

$$\hat{\varphi}(x) \int p^W(0, x; 1, y) \varphi(y) dy = \rho_0(x), \quad \varphi(y) \int \hat{\varphi}(x) p^W(0, x; 1, y) dx = \rho_1(y),$$

Let $\hat{\varphi}(0, x) = \hat{\varphi}(x)$, $\varphi(1, y) = \varphi(y)$ and

$$\hat{\varphi}(1, y) := \int p^W(0, x, 1, y) \hat{\varphi}(0, x) dx, \quad \varphi(0, x) := \int p^W(0, x, 1, y) \varphi(1, y) dy$$

Then we can obtain the Schrödinger system:

Definition 3.3.1. *The Schrödinger system is defined by the following equations:*

$$\begin{aligned}\hat{\varphi}(t, y) &= \int_{R^n} p^W(0, x; t, y) \hat{\varphi}(0, x) dx, \\ \varphi(t, x) &= \int_{R^n} p^W(t, x; 1, y) \varphi(1, y) dy,\end{aligned}$$

with the boundary conditions

$$\varphi(0, x) \cdot \hat{\varphi}(0, x) = \rho_0(x), \quad \varphi(1, y) \cdot \hat{\varphi}(1, y) = \rho_1(y).$$

3.4 Dynamic Schrödinger Bridge

We now introduce the dynamic Schrödinger bridge, which connects relative entropy minimization on path space with stochastic control and diffusion-based mass transport. This section builds on the foundational result:

Theorem 3.4.1 (Girsanov's Theorem). *Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{0 \leq t \leq T}, Q)$ be a filtered probability space on which W_t is an \mathbb{R}^n -valued Brownian motion under Q . Suppose θ_t is an \mathbb{R}^n -valued, (\mathcal{F}_t) -adapted process satisfying*

$$\mathbb{E}_Q \left[\exp \left(\frac{1}{2} \int_0^T \|\theta_s\|^2 ds \right) \right] < \infty.$$

Define the exponential martingale

$$Z_t = \exp \left(- \int_0^t \theta_s \cdot dW_s - \frac{1}{2} \int_0^t \|\theta_s\|^2 ds \right), \quad 0 \leq t \leq T.$$

Then (Z_t) is a Q -martingale, and the measure P given by

$$\frac{dP}{dQ} \Big|_{\mathcal{F}_T} = Z_T$$

is a probability measure under which the process

$$W_t^P = W_t + \int_0^t \theta_s ds$$

is an \mathbb{R}^n -valued Brownian motion relative to (\mathcal{F}_t) and P .

Let P and W be two probability measures on the path space Ω . We disintegrate both with respect to the initial position X_0 :

$$P(d\omega) = \rho_0^P(x) dx \otimes P_x(d\omega), \quad W(d\omega) = \rho_0^W(x) dx \otimes W_x(d\omega),$$

where $\rho_0^P(x) dx = P\{X_0 \in dx\}$ and P_x is the conditional law given $X_0 = x$ (similarly for W).

By the Radon–Nikodym theorem,

$$\frac{dP}{dW}(\omega) = \frac{\rho_0^P(X_0(\omega))}{\rho_0^W(X_0(\omega))} \times \frac{dP_{X_0(\omega)}}{dW_{X_0(\omega)}}(\omega).$$

Taking logarithms gives

$$\log \frac{dP}{dW} = \log \frac{\rho_0^P(X_0)}{\rho_0^W(X_0)} + \log \frac{dP_{X_0}}{dW_{X_0}}(\omega).$$

Finally, Girsanov's theorem supplies the explicit form of the second term, namely

$$\log \frac{dW_x}{dP_x} = \int_0^1 -\beta_t dW_t - \frac{1}{2} \int_0^1 \|\beta_t\|^2 dt,$$

so that overall

$$\log \frac{dP}{dW} = \log \frac{\rho_0^P(X_0)}{\rho_0^W(X_0)} + \int_0^1 \beta_t dW_t + \frac{1}{2} \int_0^1 \|\beta_t\|^2 dt.$$

Additionally, since $\int_0^1 \beta_t dW_t$ is a martingale, we can apply Itô's isometry to compute its expectation based on initial condition X_0 :

$$\mathbb{E}_P \left[\int_0^1 \beta_t dW_t \right] = \mathbb{E}_P \left[\int_0^0 \beta_t \cdot dW_t \right] = 0.$$

Thus, the expectation of the relative entropy becomes

$$\begin{aligned} \mathbb{D}(P\|W) &= \mathbb{E}_P \left[\log \frac{dP}{dW} \right] \\ &= \mathbb{E}_P \left[\log \frac{\rho_0^P(X_0)}{\rho_0^W(X_0)} \right] + \mathbb{E}_P \left[\frac{1}{2} \int_0^1 \|\beta_t\|^2 dt \right] \\ &= \mathbb{D}(\rho_0 \| \rho_0^W) + \mathbb{E}_P \left[\frac{1}{2} \int_0^1 \|\beta_t\|^2 dt \right]. \end{aligned} \tag{3.7}$$

Notice that the first term $\mathbb{D}(\rho_0 \| \rho_0^W)$ is a constant that does not depend on the path measure P . Thus, minimizing the relative entropy $\mathbb{D}(P\|W)$ is equivalent to minimizing the second term, which is the expected control cost:

$$\begin{aligned} &\min_{u \in \mathcal{U}} \mathbb{E} \left[\int_0^1 \frac{1}{2\varepsilon} \|u_t\|^2 dt \right], \\ &\text{subject to } dX_t = u_t dt + \sqrt{\varepsilon} dW_t, \\ &\quad X_0 \sim \rho_0(x), \quad X_1 \sim \rho_1(x), \end{aligned} \tag{3.8}$$

where \mathcal{U} is the set of adapted controls with finite energy.

We can show that this stochastic control problem is equivalent to (details is in section 6.5):

$$\begin{aligned} \min_{\rho, \mathbf{u}} \quad & \int_0^1 \int_{\mathbb{R}^n} \frac{1}{2} \|\mathbf{u}(\mathbf{x}, t)\|_2^2 \rho(\mathbf{x}, t) d\mathbf{x} dt \\ \text{subject to} \quad & \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = \frac{1}{2} \epsilon \Delta \rho \\ & \rho(\mathbf{x}, 0) = \rho_0(\mathbf{x}), \quad \rho(\mathbf{x}, 1) = \rho_1(\mathbf{x}), \end{aligned} \quad (3.9)$$

Proposition 3.4.2. *The optimal control $u^*(t, x)$ for the stochastic control problem*

$$\begin{aligned} \min_{u_t} \quad & \mathbb{E} \left[\int_0^1 \frac{1}{2\varepsilon} \|u_t\|^2 dt \right], \\ \text{subject to} \quad & dX_t = u_t dt + \sqrt{\varepsilon} dW_t, \\ & X_0 \sim \rho_0, \quad X_1 \sim \rho_1. \end{aligned}$$

is given by

$$u^*(t, x) = \varepsilon \nabla \ln \varphi(t, x),$$

This proposition establishes the connection between the stochastic control problem and the Schrödinger system, showing that the static Schrödinger bridge can be interpreted as a dynamic control problem where the optimal control is derived from the solution of the Schrödinger system.

3.5 Nonlinear prior Dynamic Schrödinger Bridge

In this section, we extend the dynamic Schrödinger bridge to incorporate nonlinear prior information. The problem can be extended into minimum-energy stochastic optimal control problem with nonlinear prior dynamics given by a deterministic vector field

$$f : \mathbb{R}^n \times [0, 1] \longrightarrow \mathbb{R}^n.$$

Specifically, we minimize the expected control cost

$$\begin{aligned} \min_{u(\mathbf{x}, t)} \quad & \mathbb{E} \left[\int_0^1 \frac{1}{2} \|u(x, t)\|_2^2 dt \right], \\ \text{subject to} \quad & dX(t) = f(X(t), t) dt + B(t) u_t dt + \sqrt{\epsilon} B(t) dw_t, \\ & X(0) \sim \rho_0, \quad X(1) \sim \rho_1, \end{aligned} \quad (3.10)$$

where $B(t)$ is a time-dependent matrix that scales the control input u . The nonlinear prior dynamics f can be interpreted as a drift term that guides the evolution of the system, while the stochastic noise term $\sqrt{\epsilon} B(t) dw_t$ introduces randomness into the system. We impose two sets of assumptions on f and B :

1. **Non-explosion and Lipschitz continuity.** There exist constants $c_1, c_2 > 0$ such that for all $x, y \in \mathbb{R}^n$ and $t \in [0, 1]$,

$$\|f(x, t)\|_2 + \|B(t)\|_2 \leq c_1(1 + \|x\|_2), \quad \|f(x, t) - f(y, t)\|_2 \leq c_2 \|x - y\|_2.$$

2. Uniform lower bound on diffusion. Define the diffusion tensor $D(t) = B(t)B(t)^\top$. There exists $c_3 > 0$ such that for all $x \in \mathbb{R}^n$ and $t \in [0, 1]$,

$$x^\top D(t)x \geq c_3 \|x\|_2^2.$$

Under these conditions, the uncontrolled SDE

$$dx(t) = f(x(t), t) dt + \sqrt{2\epsilon} B(t) dw(t)$$

has a strictly positive, continuous transition density, thus our stochastic control problem (3.10) is well-posed.

We do simliar approach as in the previous section, apply Itô's lemma to

$$dX(t) = f(X(t), t) dt + B(t) u_t dt + \sqrt{\epsilon} B(t) dw_t,$$

and integrate by parts, we transform the problem (3.10) into a fluid dynamics version:

$$\begin{aligned} & \min_{\rho, u} \frac{1}{2} \int_0^1 \int_{\mathbb{R}^n} \|u(x, t)\|_2^2 \rho(x, t) dx dt, \\ & \text{subject to } \frac{\partial \rho}{\partial t} + \nabla \cdot [\rho (f + B u)] = \frac{\epsilon}{2} \langle D, \Delta \rho \rangle, \\ & \rho(x, 0) = \rho_0(x), \quad \rho(x, 1) = \rho_1(x). \end{aligned} \tag{3.11}$$

where D is the diffusion tensor defined as $D(t) = B(t)B(t)^\top$. The existence and uniqueness of the solution to this problem can be established using Slater's condition. By introducing the momentum variable $m(x, t) = \rho(x, t)u(x, t)$, the object function is now a convex formulation

$$\begin{aligned} & \min_{\rho, m} \frac{1}{2} \int_0^1 \int_{\mathbb{R}^n} J(\rho, m) dx dt, \\ & \text{subject to } \frac{\partial \rho}{\partial t} + \nabla \cdot [\rho f + B m] - \frac{\epsilon}{2} \langle D, \nabla^2 \rho \rangle = 0, \\ & \rho(x, 0) = \rho_0(x), \quad \rho(x, 1) = \rho_1(x), \quad m(x, 0) = 0, \quad m(x, 1) = 0, \end{aligned} \tag{3.12}$$

where

$$J(\rho, m) = \begin{cases} \frac{\|m\|_2^2}{\rho}, & \rho > 0, \\ 0, & (m, \rho) = (0, 0), \\ +\infty, & \text{otherwise.} \end{cases}$$

Since $J(\rho, m)$ is strictly convex in (ρ, m) . The constraints

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho f + B m) - \frac{\epsilon}{2} \langle D, \nabla^2 \rho \rangle = 0, \quad \rho(\cdot, 0) = \rho_0, \quad \rho(\cdot, 1) = \rho_1, \quad m(\cdot, 0) = 0, \quad m(\cdot, 1) = 0$$

are linear in (ρ, m) . The problem (3.12) is a convex optimization problem with linear constraints, which guarantees the existence and uniqueness of the solution.

Proposition 3.5.1 (Optimality System for Gradient Flow). *Any optimal pair $(\rho^*(x, t), u^*(x, t))$ solving (3.12) must satisfy*

$$\begin{aligned}\frac{\partial \lambda}{\partial t} + \frac{1}{2} \|B(t)^T \nabla \lambda\|^2 + \nabla \lambda \cdot f &= -\frac{\epsilon}{2} \langle D, \nabla^2 \lambda \rangle, \\ \frac{\partial \rho}{\partial t} + \nabla \cdot [\rho (f + B(t)^T \nabla \lambda)] &= \frac{\epsilon}{2} \langle D, \nabla^2 \rho \rangle,\end{aligned}$$

with boundary conditions

$$\rho(x, 0) = \rho_0(x), \quad \rho(x, 1) = \rho_1(x),$$

and the optimal control is

$$u^*(x, t) = B(t)^T \nabla \lambda(x, t),$$

where $\lambda(x, t)$ is the Lagrange multiplier.

We define new variables φ and $\hat{\varphi}$ by the Hopf–Cole transformation

$$\varphi = e^{\lambda/\epsilon}, \quad \hat{\varphi} = \rho e^{-\lambda/\epsilon}.$$

Then the optimality system can be transformed into a system of backward and forward Kolmogorov equations:

Proposition 3.5.2 (Optimality System for Nonlinear Prior Dynamic Schrödinger Bridge). *Let φ and $\hat{\varphi}$ be defined as above, the optimality system for the dynamic Schrödinger bridge with nonlinear prior dynamics is given by*

$$\begin{aligned}\partial_t \varphi &= -\nabla \varphi \cdot f - \frac{\epsilon}{2} \langle D, \Delta \varphi \rangle, \\ \partial_t \hat{\varphi} &= -\nabla \hat{\varphi} \cdot f + \frac{\epsilon}{2} \langle D, \Delta \hat{\varphi} \rangle.\end{aligned}\tag{3.13}$$

with boundary conditions

$$\hat{\varphi}(x, 0)\varphi(x, 0) = \rho_0(x), \quad \hat{\varphi}(x, 1)\varphi(x, 1) = \rho_1(x),$$

and the optimal control is given by

$$u^*(x, t) = \epsilon B(t)^T \nabla \ln \varphi(x, t) = B(t)^T \nabla \lambda(x, t).$$

These two propositions can be proved using similar techniques as in the proof of Theorem 3.4.2, Caluya and Halder (2021b) made a detailed proof of this proposition by integrating by parts and using the properties of the Hopf–Cole transformation.

3.5.1 Gradient Flow Dynamic Schrödinger Bridge

In this section, we present a numerical method for solving the dynamic Schrödinger bridge problem with nonlinear prior dynamics. Assume

$$f(x, t) = -\nabla V(x), \quad B(t) = I,$$

with $V \in C^2(\mathbb{R}^n)$. Then the density-control formulation becomes

$$\begin{aligned} & \min_{\rho, u} \frac{1}{2} \int_0^1 \int_{\mathbb{R}^n} \|u(x, t)\|_2^2 \rho(x, t) dx dt, \\ & \text{s.t. } \partial_t \rho + \nabla \cdot [\rho(u - \nabla V)] = \frac{\epsilon}{2} \Delta \rho, \\ & \quad \rho(x, 0) = \rho_0(x), \quad \rho(x, 1) = \rho_1(x). \end{aligned} \tag{3.14}$$

By proposition 3.5.2, the problem (3.14) is equivalent to the following system of backward and forward Kolmogorov equations:

$$\begin{aligned} \partial_t \varphi &= -\nabla \varphi \cdot \nabla V - \frac{\epsilon}{2} \Delta \varphi, \\ \partial_t \hat{\varphi} &= \nabla \hat{\varphi} \cdot \nabla V + \frac{\epsilon}{2} \Delta \hat{\varphi}, \end{aligned} \tag{3.15}$$

with boundary conditions

$$\hat{\varphi}(x, 0) \varphi(x, 0) = \rho_0(x), \quad \hat{\varphi}(x, 1) \varphi(x, 1) = \rho_1(x).$$

Caluya and Halder (2021b) proved this system is equivalent to solving two forward Kolmogorov IVPs in “opposite” time-coordinates:

$$\begin{aligned} \partial_t \hat{\varphi} &= \nabla \cdot (\hat{\varphi} \nabla V) + \frac{\epsilon}{2} \Delta \hat{\varphi}, \quad \hat{\varphi}(x, 0) = \hat{\varphi}_0(x), \\ \partial_s p &= \nabla \cdot (p \nabla V) + \frac{\epsilon}{2} \Delta p, \quad p(x, 0) = \varphi_1(x) e^{-V(x)/\epsilon}, \end{aligned} \tag{3.16}$$

where $s = 1 - t$ and $\varphi(x, t) = p(x, 1 - t) e^{V(x)/\epsilon}$.

3.5.2 Proximal Recursion Algorithm

We now give a numerical method for solving the dynamic Schrödinger bridge problem with nonlinear prior dynamics. This method was first proposed by Caluya and Halder (2019) to deal with nonlinear gradient flow problems. Here we modify their method for the Schrödinger bridge setting. We consider the problem as propagation of particles in a flow field. The particles are initialized according to the initial density ρ_0 and evolve over time to match the target density ρ_1 . The evolution is governed by non-controlled stochastic differential equations (SDEs) with a prior drift term given by the potential V :

$$dx = -\nabla V(x) dt + \sqrt{\epsilon} dw.$$

Assume the time step τ is small enough. By Euler–Maruyama with step τ (or σ) to obtain new positions

$$x_i^k = x_i^{k-1} - \tau \nabla V(x_i^{k-1}) + \sqrt{\epsilon \tau} \xi_i^k,$$

where $\xi_i^k \sim \mathcal{N}(0, I)$ are independent. Since we have the prior drift $-\nabla V$ and diffusion is isotropic, the relevant Lyapunov functional is the free energy

$$\mathcal{F}(\rho) = \int_{\mathbb{R}^n} V(x) \rho(x) dx + \epsilon \int_{\mathbb{R}^n} \rho(x) \ln \rho(x) dx.$$

Thus, the proximal operator for the free energy functional is given by

$$\mathcal{P}_\tau(\rho) = \arg \min_{\tilde{\rho}} \left\{ \tau \mathcal{F}(\tilde{\rho}) + \frac{1}{2} \|\tilde{\rho} - \rho\|_2^2 \right\}.$$

Following the theorem proposed by Jordan et al. (1998), in each steps we can update the positions of the particles and use proximal operator to update the evolution of forward Kolmogorov equation system (3.16).

$$\hat{\varphi}_k = \text{prox}_{\tau \mathcal{F}}^{W_2}(\hat{\varphi}_{k-1}), \quad p_k = \text{prox}_{\sigma \mathcal{F}}^{W_2}(p_{k-1}),$$

where σ is the step size for the backward Kolmogorov equation and τ is the step size for the forward Kolmogorov equation.

3.5.3 Sinkhorn Algorithm for Proximal Recursion

The proximal recursion algorithm can be implemented using the Sinkhorn algorithm to compute the proximal operator. Define the $N \times N$ cost matrix C by

$$C_{ij} = \|x_i^{k-1} - x_j^k\|_2^2.$$

Let $\gamma > 0$ be the entropic regularization parameter. The updated weights $\hat{\varphi}^k = (\hat{\varphi}_1^k, \dots, \hat{\varphi}_N^k)$ solve

$$\hat{\varphi}^k = \arg \min_{\varphi \in R_+^N} \min_{M \in \Pi(\hat{\varphi}^{k-1}, \varphi)} \frac{1}{2} \langle C, M \rangle + \gamma \langle M, \log M - 1 \rangle + \tau \langle V(x^{k-1}) + \epsilon \log \hat{\varphi}, \hat{\varphi} \rangle,$$

where

$$\Pi(a, b) = \{M \in R_+^{N \times N} \mid M \mathbf{1} = a, M^T \mathbf{1} = b\},$$

and $V(x^{k-1})$ denotes the vector $[V(x_i^{k-1})]_{i=1}^N$. Similarly we computes p^k with step σ . The Lagrangian for the minimization problem

$$\min_{M \in \Pi(\hat{\varphi}^{k-1}, \hat{\varphi})} \frac{1}{2} \langle C, M \rangle + \gamma \langle M, \log M - 1 \rangle + \tau \langle V(x^{k-1}) + \epsilon \log \hat{\varphi}, \hat{\varphi} \rangle$$

can be written as

$$\begin{aligned} \mathcal{L}(M, \lambda, \mu) = & \frac{1}{2} \langle C, M \rangle + \gamma \langle M, \log M - 1 \rangle + \tau \langle V(x^{k-1}) + \epsilon \log \hat{\varphi}, \hat{\varphi} \rangle \\ & - \langle \lambda, \hat{\varphi}^{k-1} - M \mathbf{1} \rangle - \langle \mu, \hat{\varphi} - M^T \mathbf{1} \rangle. \end{aligned}$$

Thus, we obtain the optimal solution M^* as

$$M_{ij}^* = \exp(\boldsymbol{\lambda}(i)/\gamma) \exp(-\boldsymbol{C}_k(i, j)/(2\gamma)) \exp(\boldsymbol{\mu}(j)/\gamma), \quad (3.17)$$

Theorem 3.5.3 (Proximal–Sinkhorn Recursion). *Let*

$$a = \hat{\varphi}^{k-1}, \quad K_{ij} = \exp\left(-\frac{C_{ij}}{2\gamma}\right), \quad \xi_j = \exp\left(-1 - \frac{V(x_j^{k-1})}{\epsilon}\right).$$

Then the unique minimizer $\hat{\varphi}^k$ of

$$\min_{b \geq 0} \min_{M \in \Pi(a, b)} \left[\frac{1}{2} \langle C, M \rangle + \gamma \langle M, \log M - 1 \rangle + \tau \langle V(x^{k-1}) + \epsilon \log b, b \rangle \right]$$

is obtained as the fixed-point of the two-step iteration

$$\begin{cases} y^{(t+1)} = a \oslash (K z^{(t)}), \\ z^{(t+1)} = [\xi \oslash (K^T y^{(t+1)})]^{\frac{\epsilon \tau}{\epsilon \tau + \gamma}}, \end{cases}$$

and in the limit $t \rightarrow \infty$,

$$\hat{\varphi}^k = z^* \odot (K^T y^*).$$

Thus, the proximal recursion algorithm can be implemented as follows:

Algorithm 4 Proximal-Sinkhorn Recursion for $\hat{\varphi}^k$

1: **Input:** Previous weights $a \leftarrow \hat{\varphi}^{k-1} \in \mathbb{R}_+^N$, positions $x^{k-1}, x^k \in \mathbb{R}^{d \times N}$, potential

$V: \mathbb{R}^d \rightarrow \mathbb{R}$, parameters $\gamma, \tau, \epsilon > 0$.

Ensure: Updated weights $\hat{\varphi}^k \in \mathbb{R}_+^N$.

2: Compute cost matrix

$$C_{ij} \leftarrow \|x_i^{k-1} - x_j^k\|_2^2, \quad K_{ij} \leftarrow \exp(-C_{ij}/(2\gamma))$$

3: Compute potential-weight vector $\xi_j \leftarrow \exp(-1 - V(x_j^{k-1})/\epsilon)$

4: Initialize $z^{(0)} \in \mathbb{R}_+^N$ (e.g. $z^{(0)} = \mathbf{1}$)

5: **for** $t = 0, 1, 2, \dots$ until convergence **do**

6: Row-scaling:

$$y^{(t+1)} \leftarrow a \oslash (K z^{(t)})$$

7: Column-scaling (proximal step):

$$z^{(t+1)} \leftarrow [\xi \oslash (K^T y^{(t+1)})]^{\frac{\epsilon \tau}{\epsilon \tau + \gamma}}$$

8: **end for**

$$9: \hat{\varphi}^k \leftarrow z^{(*)} \odot (K^T y^{(*)})$$

10: **return** $\hat{\varphi}^k$

Similar procedure can be applied to compute p^k for the backward Kolmogorov equation system (3.16).

3.5.4 Proximal Recursion for Kolmogorov System

We now combine the reformulations and proximal-Sinkhorn recursions into a single outer loop that finds the endpoint Schrödinger factors $(\hat{\varphi}_0, \varphi_1)$. Once these are known, the transient factors $(\hat{\varphi}(x, t), \varphi(x, t))$ are obtained by the forward–backward proximal recursions. The endpoint iteration proceeds as follows:

1. **Initialize** a strictly positive guess for $\hat{\varphi}_1(x)$.

$$2. \quad \varphi_1(x) = \frac{\rho_1(x)}{\hat{\varphi}_1(x)}.$$

3. **Set** the initial profile $p(s = 0)$ by

$$p(s = 0) = \varphi_1(x) e^{-V(x)/\epsilon}.$$

4. **Forward proximal step.** Apply the s -recursion to solve the IVP in $s \in [0, 1]$, yielding $p(s = 1)$.

5. **Recover**

$$\varphi_0 = p(s = 1) e^{V(x)/\epsilon}.$$

$$6. \quad \hat{\varphi}_0(x) = \frac{\rho_0(x)}{\varphi_0(x)}.$$

7. **Backward proximal step.** Apply the t -recursion to solve the IVP in $t \in [0, 1]$, yielding $\hat{\varphi}_1(x)$.

8. **Convergence check.** Compute the squared 2-Wasserstein distances between the current and previous $(p(s = 0), \hat{\varphi}_0)$ pairs. If both are below the prescribed tolerance, stop; otherwise return to step 2.

Once $(\hat{\varphi}_0, \varphi_1)$ has converged, the full time-dependent pair $(\hat{\varphi}(x, t), \varphi(x, t))$ is obtained by the forward and backward proximal recursions.

Remark 3.5.4. In the above algorithm, when doing proximal recursion steps, we use interpolation proposed by Hardy (1971) after obtaining the new positions of the particles, which helps to maintain the continuity and consistency of the particle trajectories both in the forward and backward Kolmogorov equations.

Algorithm 5 Proximal Recursion for Gradient Flow Schrödinger Bridge

```

1: initialize  $\hat{\varphi}_1 \leftarrow$  random positive vector in  $R^N$ 
2:  $\varphi_0 \leftarrow 0$ ,  $\hat{\varphi}_0 \leftarrow 0$ ,  $\varphi_1 \leftarrow 0$ ,  $p_0 \leftarrow 0$ ,  $p_1 \leftarrow 0$ 
3: allocate  $p^{\text{temp}}$ ,  $\hat{\varphi}^{\text{temp}} \in R^{N \times (\text{numSteps}+1)}$ 
4:  $p_0^{\text{old}} \leftarrow p_0$ ,  $\hat{\varphi}_0^{\text{old}} \leftarrow \hat{\varphi}_0$ 
5: for  $\ell = 1, \dots, \text{maxIter}_{\text{SB}}$  do
6:    $\varphi_1 \leftarrow \rho_1 \oslash \hat{\varphi}_1$ 
7:    $p_0 \leftarrow \varphi_1 \odot \exp(-V(x)/\epsilon)$ 
8:    $p^{\text{temp}}(:, 1) \leftarrow p_0$ 
9:   for  $i = 1, \dots, \text{numSteps}$  do
10:    Use Euler–Maruyama to update particle positions
11:     $p^{\text{temp}}(:, i + 1) \leftarrow$  Algorithm 4
12:   end for
13:   Interpolate  $p^{\text{temp}}(:, \text{numSteps} + 1)$  onto initial density simulation points.
14:    $p_1 \leftarrow p^{\text{temp}}(:, \text{numSteps} + 1)$ 
15:    $\varphi_0 \leftarrow p_1 \odot \exp(V(x)/\epsilon)$ 
16:    $\hat{\varphi}_0 \leftarrow \rho_0 \oslash \varphi_0$ 
17:    $\hat{\varphi}^{\text{temp}}(:, 1) \leftarrow \hat{\varphi}_0$ 
18:   for  $j = 1, \dots, \text{numSteps}$  do
19:    Use Euler–Maruyama to update particle positions
20:     $\hat{\varphi}^{\text{temp}}(:, j + 1) \leftarrow$  Algorithm 4
21:   end for
22:   Interpolate  $\hat{\varphi}^{\text{temp}}(:, \text{numSteps} + 1)$  onto final density simulation points.
23:    $\hat{\varphi}_1 \leftarrow \hat{\varphi}^{\text{temp}}(:, \text{numSteps} + 1)$ 
24:   if  $W_2^2(p_0^{\text{old}}, p_0) < \text{tol}_{\text{SB}}$  and  $W_2^2(\hat{\varphi}_0^{\text{old}}, \hat{\varphi}_0) < \text{tol}_{\text{SB}}$  then
25:     break
26:   end if
27:    $p_0^{\text{old}} \leftarrow p_0$ ,  $\hat{\varphi}_0^{\text{old}} \leftarrow \hat{\varphi}_0$ 
28: end for                                     ▷ Endpoint factors:  $\hat{\varphi}_0, \varphi_1$ 
29: initialize  $\hat{\varphi}_1^{\text{trans}} \leftarrow \hat{\varphi}_0$ 
30: initialize  $p_1^{\text{trans}} \leftarrow \varphi_1 \odot \exp(-V(x)/\epsilon)$ 
31: for  $k = 1, \dots, \text{numSteps}$  do
32:   Use Euler–Maruyama to update particle positions
33:    $\hat{\varphi}_{k+1}^{\text{trans}} \leftarrow$  Algorithm 4
34:   interpolate  $\hat{\varphi}_{k+1}^{\text{trans}}$  onto global grid points.
35:    $p_{k+1}^{\text{trans}} \leftarrow$  Algorithm 4
36:    $\varphi_{k+1}^{\text{trans}} \leftarrow p_{\text{numSteps}+2-k}^{\text{trans}} \odot \exp(V(x)/\epsilon)$ 
37:   interpolate  $\varphi_{k+1}^{\text{trans}}$  onto global grid points.
38: end for
39: return  $\{\hat{\varphi}_k^{\text{trans}}, \varphi_k^{\text{trans}}\}_{k=1}^{\text{numSteps}}$ 
40: Calculate density  $\{\rho_k^{\text{trans}}\}_{k=1}^{\text{numSteps}} = \{\hat{\varphi}_k^{\text{trans}}, \varphi_k^{\text{trans}}\}_{k=1}^{\text{numSteps}}$ .

```

3.6 Schrödinger Bridge with Reflecting Boundary

In this section, we consider the Schrödinger bridge problem with *reflecting* boundary conditions.

3.6.1 Problem Formulation

Let $\Omega \subset \mathbb{R}^n$ be a connected, smooth, bounded domain with boundary $\partial\Omega$, and write $\overline{\Omega} := \Omega \cup \partial\Omega$ for its closure. We study the reflected SDE

$$\begin{aligned} dX_t &= u(X_t, t) dt + \sqrt{\epsilon} dW_t + \mathbf{n}(X_t) d\gamma_t, \\ X_0 &\sim \rho_0, \quad X_1 \sim \rho_1, \\ X_t &\in \overline{\Omega}, \quad t \in [0, 1], \end{aligned} \tag{3.18}$$

where $\mathbf{n}(x)$ denotes the *inward* unit normal at $x \in \partial\Omega$. Harrison (2013); Glynn and Wang (2018) show that the process $\gamma = (\gamma_t)_{t \in [0, 1]}$ should be continuous, nonnegative, nondecreasing, starts from $\gamma_0 \equiv 0$, and increases *only* when X_t hits the boundary to enforce the reflection. Writing $1_{\{\cdot\}}$ for the indicator of a set, we impose the minimal local time conditions

$$\gamma_t = \int_0^t 1_{\{X_s \in \partial\Omega\}} d\gamma_s, \quad \int_0^1 1_{\{X_t \notin \partial\Omega\}} d\gamma_t = 0,$$

which mean exactly that γ grows only at boundary contact times. Under such conditions, the reflected SDE (3.18) admits a existence and unique solution for any progressively measurable control u satisfying suitable integrability conditions (Skorokhod, 1961; Pilipenko, 2014).

The control $u : \overline{\Omega} \times [0, 1] \rightarrow \mathbb{R}^n$ is progressively measurable with respect to the filtration generated by the Brownian motion W . Our goal is to steer X from ρ_0 to ρ_1 over $[0, 1]$ at minimum quadratic effort:

$$\begin{aligned} \min_u \quad & \mathbb{E} \left[\int_0^1 \frac{1}{2} \|u(X_t, t)\|^2 dt \right], \\ \text{s.t.} \quad & dX_t = u(X_t, t) dt + \sqrt{\epsilon} dW_t + \mathbf{n}(X_t) d\gamma_t, \\ & X_0 \sim \rho_0, \quad X_1 \sim \rho_1, \quad X_t \in \overline{\Omega} \text{ for } t \in [0, 1]. \end{aligned} \tag{3.19}$$

We can reformulate the problem via Itô's formula (details in Appendix 6.8) as the following optimal control problem:

$$\begin{aligned} \min_{(\rho, u)} \quad & \int_0^1 \int_{\Omega} \frac{1}{2} \|u(x, t)\|^2 \rho(t, x) dx dt \\ \text{s.t.} \quad & \partial_t \rho + \nabla \cdot (\rho u) = \frac{\epsilon}{2} \Delta \rho, \quad (x, t) \in \Omega \times (0, 1), \\ & \mathbf{n} \cdot \left(\frac{\epsilon}{2} \nabla \rho - \rho u \right) = 0, \quad (x, t) \in \partial\Omega \times (0, 1), \\ & \rho(0, \cdot) = \rho_0, \quad \rho(1, \cdot) = \rho_1. \end{aligned} \tag{3.20}$$

Theorem 3.6.1 (Optimal control of the reflected Schrödinger bridge). *A pair $(\rho^{\text{opt}}, u^{\text{opt}})$ solving the variational problem must satisfy the system of coupled nonlinear PDEs:*

$$\begin{aligned}\partial_t \rho^{\text{opt}} + \nabla \cdot (\rho^{\text{opt}} u^{\text{opt}}) &= \frac{\varepsilon}{2} \Delta \rho^{\text{opt}}, \\ \partial_t \lambda + \frac{1}{2} \|\nabla \lambda\|^2 &= -\frac{\varepsilon}{2} \Delta \lambda,\end{aligned}$$

where

$$u^{\text{opt}}(t, \cdot) = \nabla \lambda(t, \cdot).$$

The system is subject to the boundary and terminal conditions

$$\begin{aligned}\langle \nabla \lambda, \mathbf{n} \rangle &= 0, && \text{on } \partial\Omega, \quad t \in [0, 1], \\ \rho^{\text{opt}}(0, \cdot) &= \rho_0, \quad \rho^{\text{opt}}(1, \cdot) = \rho_1, \\ \mathbf{n} \cdot \left(\frac{\varepsilon}{2} \nabla \rho^{\text{opt}} - \rho^{\text{opt}} u^{\text{opt}} \right) &= 0, && \text{on } \partial\Omega, \quad t \in [0, 1].\end{aligned}$$

We define new variables φ and $\hat{\varphi}$ via the Hopf transformation

$$\varphi = e^{\lambda/\varepsilon}, \quad \hat{\varphi} = \rho e^{-\lambda/\varepsilon}.$$

Then the optimality system can be transformed into a pair of forward and backward Kolmogorov equations:

Proposition 3.6.2 (Optimality system for the reflected Schrödinger bridge). *Let φ and $\hat{\varphi}$ be defined as above. Then the optimality system for the reflected Schrödinger bridge problem is*

$$\begin{aligned}\partial_t \varphi &= -\frac{\varepsilon}{2} \Delta \varphi, \\ \partial_t \hat{\varphi} &= \frac{\varepsilon}{2} \Delta \hat{\varphi},\end{aligned}\tag{3.21}$$

with boundary conditions

$$\hat{\varphi}(x, 0) \varphi(x, 0) = \rho_0(x), \quad \hat{\varphi}(x, 1) \varphi(x, 1) = \rho_1(x),$$

and the reflecting (Neumann) boundary conditions

$$\langle \nabla \varphi, \mathbf{n} \rangle = 0, \quad \langle \nabla \hat{\varphi}, \mathbf{n} \rangle = 0, \quad x \in \partial\Omega, \quad t \in [0, 1].$$

Thus, the system (3.21) characterizes the forward and backward heat equations under reflecting boundary conditions.

We take a simple reflected Schrödinger bridge example from (Caluya and Halder, 2021a) with 1-D domains, which is shown in section 4.2.1.

3.6.2 Finite Element Method for Reflected Schrödinger Bridge

Since the optimality system for the reflected Schrödinger bridge is a pair of heat equations with Neumann boundary conditions, we can use the finite element method (FEM) to solve it numerically. We first consider the time discretization. Let K be the number of time steps and discretize $(0, 1)$ uniformly with step size $\Delta t = 1/K$. Using the backward Euler scheme, the forward heat equation is approximated by

$$\frac{\hat{\varphi}^{k+1} - \hat{\varphi}^k}{\Delta t} - \frac{\varepsilon}{2} \Delta \hat{\varphi}^{k+1} = 0,$$

while the backward heat equation is

$$\frac{\varphi^k - \varphi^{k+1}}{-\Delta t} + \frac{\varepsilon}{2} \Delta \varphi^k = 0.$$

We discretize the spatial domain $\Omega \subset \mathbb{R}^d$ by a conforming triangulation $\mathcal{T}_h = \{K\}$ consisting of simplices (triangles if $d = 2$, tetrahedra if $d = 3$). Let $h = \max_{K \in \mathcal{T}_h} \text{diam}(K)$ denote the mesh size.

We choose the standard continuous piecewise linear Lagrange finite element space

$$V_h := \{v_h \in C^0(\bar{\Omega}) : v_h|_K \in \mathbb{P}_1(K) \quad \forall K \in \mathcal{T}_h\},$$

where $\mathbb{P}_1(K)$ is the space of polynomials of degree at most one on each element K . The basis functions $\{\phi_i\}_{i=1}^N$ are defined nodally: for each mesh vertex x_i , the associated basis function ϕ_i satisfies

$$\phi_i(x_j) = \delta_{ij}, \quad 1 \leq i, j \leq N,$$

so that each $v_h \in V_h$ can be written uniquely as

$$v_h(x) = \sum_{i=1}^N v_i \phi_i(x),$$

with degrees of freedom $v_i = v_h(x_i)$. Thus, for the forward heat equation in backward Euler form, each time step requires solving: find $\hat{\varphi}_h^{k+1} \in V_h$ such that

$$\int_{\Omega} \hat{\varphi}_h^{k+1} v_h dx + \Delta t \frac{\varepsilon}{2} \int_{\Omega} \nabla \hat{\varphi}_h^{k+1} \cdot \nabla v_h dx = \int_{\Omega} \hat{\varphi}_h^k v_h dx, \quad \forall v_h \in V_h.$$

Expanding $\hat{\varphi}_h^{k+1} = \sum_j U_j^{k+1} \phi_j$ and testing with $v_h = \phi_i$, we obtain the discrete linear system

$$(M + \frac{\varepsilon}{2} \Delta t L) U^{k+1} = M U^k,$$

where $U^k = (U_1^k, \dots, U_N^k)^\top$ is the vector of unknown coefficients, and the he matrices M and L are defined by

$$M_{ij} = \int_{\Omega} \phi_i(x) \phi_j(x) dx, \quad L_{ij} = \int_{\Omega} \nabla \phi_i(x) \cdot \nabla \phi_j(x) dx.$$

Finally, we can assemble both matrices element-wise: for each element $\mathcal{K} \in \mathcal{T}_h$, define the local basis functions $\{\phi_i^{\mathcal{K}}\}$ restricted to \mathcal{K} , and compute the local element matrices

$$M_{ij}^{\mathcal{K}} = \int_{\mathcal{K}} \phi_i^{\mathcal{K}}(x) \phi_j^{\mathcal{K}}(x) dx, \quad L_{ij}^{\mathcal{K}} = \int_{\mathcal{K}} \nabla \phi_i^{\mathcal{K}}(x) \cdot \nabla \phi_j^{\mathcal{K}}(x) dx.$$

These integrals can be evaluated exactly using affine mappings from a reference simplex \hat{K} to K , and standard Gaussian quadrature. The global matrices M and L are then assembled by summing contributions:

$$M = \sum_{K \in \mathcal{T}_h} R_K^\top M^K R_K, \quad L = \sum_{K \in \mathcal{T}_h} R_K^\top L^K R_K,$$

where $R_{\mathcal{K}}$ is the restriction (or connectivity) matrix mapping local degrees of freedom on element \mathcal{K} to global degrees of freedom.

The backward heat equation is discretized similarly. Therefore, the total algorithm for solving the reflected Schrödinger bridge problem is summarized as follows:

Algorithm 6 Finite Element Fixed Point Iteration

- 1: Initialize $\hat{\varphi}_1^{(0)}(x) \equiv 1$ at $t = 1$.
 - 2: **for** $m = 0, 1, 2, \dots$ until convergence **do**
 - 3: Compute terminal condition $\varphi^{(m)}(1, x) = \rho_1(x)/\hat{\varphi}^{(m)}(1, x)$.
 - 4: Solve backward heat equation for $\varphi^{(m)}(t, x)$ from $t = 1$ to $t = 0$.
 - 5: Update initial condition $\hat{\varphi}^{(m)}(0, x) = \rho_0(x)/\varphi^{(m)}(0, x)$.
 - 6: Solve forward heat equation for $\hat{\varphi}^{(m)}(t, x)$ from $t = 0$ to $t = 1$.
 - 7: Check convergence of $\hat{\varphi}^{(m)}(1, x)$; if satisfied, stop.
 - 8: **end for**
 - 9: Recover density $\rho(t, x) = \varphi(t, x)\hat{\varphi}(t, x)$.
-

3.6.3 Convergence Analysis and Complexity for FEM in Reflected Schrödinger Bridge

We now analyze the convergence and computational complexity of the finite element method (FEM) applied to the reflected Schrödinger bridge problem.

Finite element convergence. For the spatial discretization, we use the standard continuous piecewise linear finite element space $V_h \subset H^1(\Omega)$. By Cea's lemma and regularity for elliptic equations with smooth solutions, shown in Brenner and Scott (2008), the finite element approximation achieves

$$\|u - u_h\|_{L^2(\Omega)} = O(h^2), \quad \|u - u_h\|_{H^1(\Omega)} = O(h),$$

as the mesh size $h \rightarrow 0$. Together with the backward Euler time stepping, which is first-order accurate and unconditionally stable, the fully discrete scheme yields

$$\|u(t_n) - u_h^n\|_{L^2(\Omega)} \leq C(h^2 + \Delta t),$$

for $t_n = n\Delta t$, provided the exact solution u is sufficiently smooth. Thus, the finite element scheme converges in the standard sense of numerical analysis.

Algorithmic convergence. It has been shown in Bushell (1973) that this recursion is in fact contractive in Hilbert's projective metric, provided two conditions are satisfied:

1. the endpoint distributions ρ_0, ρ_1 have compact supports in Ω ,
2. the transition probability densities associated with the forward and backward heat kernels are continuous and strictly positive.

Therefore, the sequence of iterates $\{\varphi^{(m)}, \hat{\varphi}^{(m)}\}$ converges geometrically to the unique Schrödinger potentials, and consequently the density $\rho^{(m)}(t, x) = \varphi^{(m)}(t, x)\hat{\varphi}^{(m)}(t, x)$ converges to the optimal solution of the reflected Schrödinger bridge problem.

Computational complexity. We analyze the complexity of the fixed point finite element algorithm. Each outer iteration consists of one backward and one forward calculation, where each outer iteration solves a backward heat equation for φ and a forward heat equation for $\hat{\varphi}$, both with backward Euler time stepping and P_1 finite elements.

Let $N := \dim V_h$ be the number of degrees of freedom (DoFs), $M := 1/\Delta t$ the number of time steps, and K the number of outer forward–backward iterations until convergence. Since each time step we need to solve

$$A u^{k+1} = M_h u^k, \quad A := M_h + \frac{\varepsilon}{2} \Delta t L_h,$$

with homogeneous Neumann boundary conditions on $\partial\Omega$, where M_h and L_h are the mass and stiffness matrices, respectively. Notice that matrix A is sparse, symmetric positive definite (SPD). Considering an optimal iterative solver (e.g. CG with multi-grid preconditioning) with one step costs $\mathcal{O}(N \log N)$, while sparse direct solvers yield dimension-dependent costs ($\sim \mathcal{O}(N^{3/2})$) in 2D.

For K outer iterations,

$\text{Total} = \mathcal{O}(M K \text{SolvePDE}(N)) = \begin{cases} \mathcal{O}(K M N \log N), & \text{typical practical,} \\ \mathcal{O}(K M N^\alpha), \alpha \in [1, 2], & \text{generic.} \end{cases}$
--

Remark 3.6.3. The factorization of A can be performed once at the beginning and reused throughout the iterations, significantly reducing the computational cost. (as long as $\Delta t, \varepsilon$ and the mesh are fixed). Thus we may factorize A once and reuse it. Let $\mathcal{C}_{\text{fac}}(N)$ be the factorization cost and $\mathcal{C}_{\text{sol}}(N)$ the cost of a triangular solve per right-hand side (RHS). Then

$$\text{Total} = \underbrace{\mathcal{C}_{\text{fac}}(N)}_{\text{once}} + \underbrace{(2KM) \mathcal{C}_{\text{sol}}(N)}_{\text{forward+backward over } M \text{ iterations}},$$

with typical 2D scalings for sparse Cholesky:

$$\mathcal{C}_{\text{fac}}(N) \sim \mathcal{O}(N^{3/2}), \quad \mathcal{C}_{\text{sol}}(N) \sim \mathcal{O}(N \log N).$$

This reuse substantially reduces the constant factor compared to refactorizing at every step.

Chapter 4

Experimental Results

4.1 Numerical example with simple mass splitting transport.

We conduct our numerical study on the unit square $\Omega = [0, 1]^2$, discretized into a uniform $N \times N$ Cartesian grid. The initial density ρ_0 is a single Gaussian blob of variance σ^2 , localized near the top-left corner:

$$\rho_0(x, y) = \frac{1}{Z_0} \exp\left(-\frac{(x - \xi_0)^2 + (y - \eta_0)^2}{2\sigma^2}\right), \quad (\xi_0, \eta_0) \approx (0.2, 0.8),$$

where Z_0 normalizes $\int_{\Omega} \rho_0 = 1$.

The target density ρ_1 models a “mass-splitting” scenario: the mass breaks into three separate Gaussian packets,

$$\rho_1(x, y) = \sum_{i=1}^3 \frac{\alpha_i}{Z_i} \exp\left(-\frac{(x - \xi_i)^2 + (y - \eta_i)^2}{2\sigma^2}\right), \quad \sum_{i=1}^3 \alpha_i = 1,$$

with centers (ξ_i, η_i) chosen so that the three lumps occupy distinct regions of Ω . Each Z_i ensures unit total mass and the weights α_i distribute the mass among the three targets.

4.1.1 Dynamic transport method for mass splitting

The SDR algorithm is implemented with the same parameters as in the previous section, and we measure the convergence rate and computational cost for each grid size. The results of the numerical experiment is shown in the following figure:

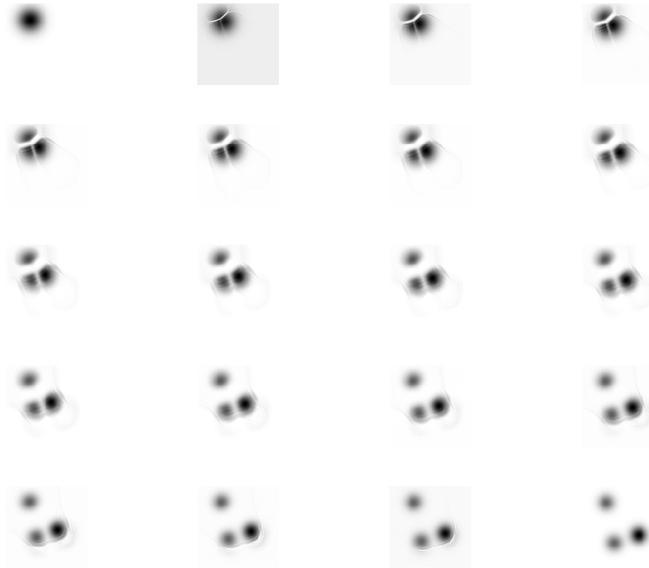


Figure 4.1: Mass-splitting phenomenon in the SDR algorithm. The initial mass is concentrated in a single Gaussian ball, which splits into three target regions during the transport process.

From the following figure, we can see the energy functional $\mathcal{J}_\beta^w(\mathcal{I}(U))$ converges.

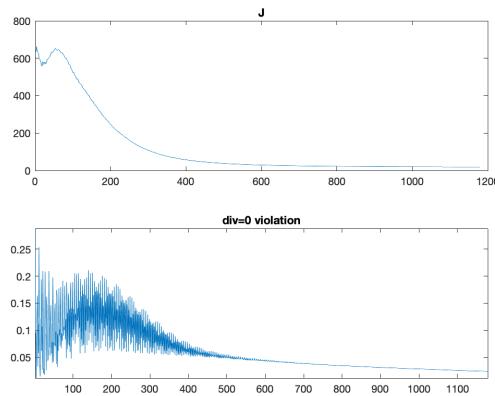


Figure 4.2: Convergence of the energy functional $\mathcal{J}_\beta^w(\mathcal{I}(U))$ for the SDR algorithm. The energy decreases steadily, indicating convergence to a stable solution.

4.1.2 Sinkhorn method for mass splitting

The cost matrix C is defined via the squared Euclidean distance between grid points, and the entropic regularization parameter ε is set to a small value (e.g., 0.01) to preserve the sharpness of transport plans while ensuring convergence. We implement both the Sinkhorn algorithm and the SDR algorithm with different grid

sizes $N = 20, 50, 100$ and measure their convergence rates and computational costs. The Sinkhorn algorithm is implemented with a fixed number of iterations (e.g., $K = 1000$) and a convergence tolerance $\epsilon = 10^{-2}$. From the picture below, we can see that the procedure of mass-splitting.

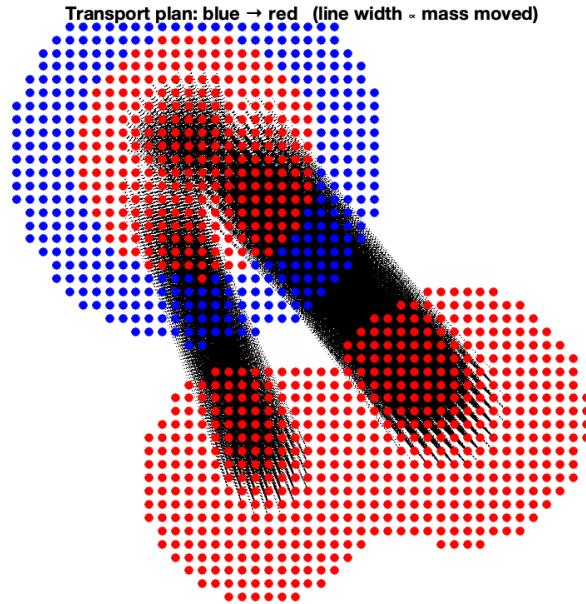


Figure 4.3: Mass-splitting phenomenon in the Sinkhorn algorithm. The initial mass is concentrated in a single Gaussian ball with blue points, which splits into three target red regions during the transport process.

The convergence of Sinkhorn algorithm is shown in the following figure which is checked by satisfying $\|\mathbf{P}\mathbf{1}_m - \mathbf{a}\| < \epsilon$ and $\|\mathbf{P}^\top \mathbf{1}_n - \mathbf{b}\| < \epsilon$:

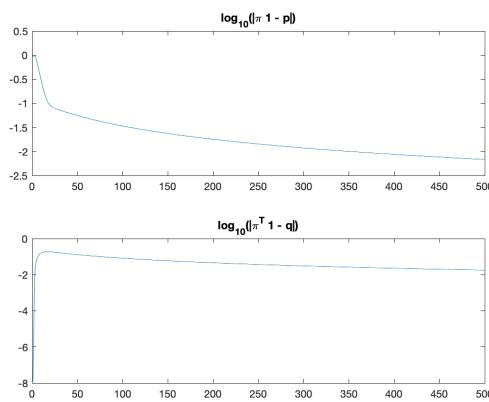


Figure 4.4: Convergence check for the Sinkhorn algorithm.

The calculation of Wasserstein distance is based on the optimal transport plan ob-

tained from the Sinkhorn algorithm, which is computed as follows:

$$W_2(\rho_0, \rho_1) = \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} \|x - y\|^2 d\pi^*(x, y),$$

Since we calculate it with regularized entropic Sinkhorn algorithm, the Wasserstein distance is approximated from the follow figure:

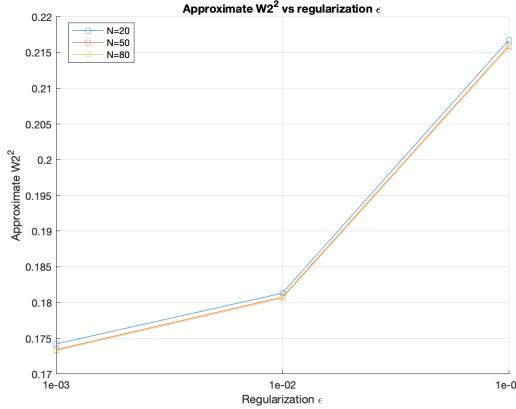


Figure 4.5: Wasserstein distance computed from the Sinkhorn algorithm. The distance is approximated using ϵ -regularized entropic Sinkhorn algorithm, which provides a stable estimate of the Wasserstein distance between the initial and target densities.

And the running time of the Sinkhorn algorithm is shown in the following figure:

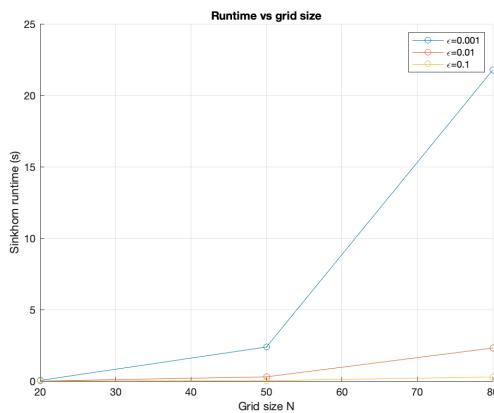


Figure 4.6: Running time of the Sinkhorn algorithm for different grid sizes and variable ϵ . The time increases with the grid size, reflecting the computational complexity of the Sinkhorn algorithm.

4.1.3 Modified Sinkhorn algorithm for Schrödinger bridge

By previous sections, we have established that the Schrödinger bridge can be formulated as a stochastic control problem. This allows us to use numerical methods for

solving the Schrödinger system and computing the optimal control. The Schrödinger bridge can be cast as a stochastic control problem:

$$\begin{aligned} \min_{u_t} \quad & \mathbb{E} \left[\int_0^1 \frac{1}{2\varepsilon} \|u_t\|^2 dt \right], \\ \text{subject to} \quad & dX_t = u_t dt + \sqrt{\varepsilon} dW_t, \\ & X_0 \sim \rho_0, \quad X_1 \sim \rho_1. \end{aligned}$$

The optimal feedback control is

$$u^*(t, x) = \varepsilon \nabla \ln \varphi(t, x),$$

where $\varphi, \hat{\varphi}$ solve the Schrödinger system

$$\partial_t \varphi + \frac{\varepsilon}{2} \Delta \varphi = 0, \quad \partial_t \hat{\varphi} - \frac{\varepsilon}{2} \Delta \hat{\varphi} = 0,$$

with $\varphi(0)\hat{\varphi}(0) = \rho_0$, $\varphi(1)\hat{\varphi}(1) = \rho_1$.

Following (3.5), we can show that the optimal plan factorises in Sinkhorn method as:

$$\pi^*(x, y) = \hat{\varphi}(x) K(x, y) \varphi(y), \quad (4.1)$$

subject to the marginal constraints

$$\int \hat{\varphi}(x) K(x, y) \varphi(y) dy = \rho_0(x), \quad \int \varphi(y) K(x, y) \hat{\varphi}(x) dx = \rho_1(y),$$

where $K(x, y)$ is the transition kernel of the Wiener measure. On a finite grid these become matrix equations that

$$\mathbf{P}_{i,j} = \mathbf{u}_i \mathbf{K}_{i,j} \mathbf{v}_j \quad (4.2)$$

where \mathbf{P} is the transport plan, \mathbf{u} and \mathbf{v} are vectors representing the marginals $\hat{\varphi}$ and φ , and \mathbf{K} is the kernel matrix corresponding to the transition probabilities.

For numerical implementation, we can use the Sinkhorn algorithm to iteratively update the vectors \mathbf{u} and \mathbf{v} until convergence. Then we can obtain Schrödinger's forward and backward equations by initial Cauchy data convolution with the kernel K :

$$\varphi_t(x) = \int K_{1-t}(x, y) v dy, \quad \hat{\varphi}_t(x) = \int K_t(y, x) u dy. \quad (4.3)$$

The family

$$\rho_t(x) = \varphi_t(x) \hat{\varphi}_t(x), \quad t \in [0, 1], \quad (4.4)$$

gives the solution to the Schrödinger system, where φ_t and $\hat{\varphi}_t$ are the solutions to the forward and backward heat equations, respectively. The numerical result is shown below, where we choose $\varepsilon = 0.005$.

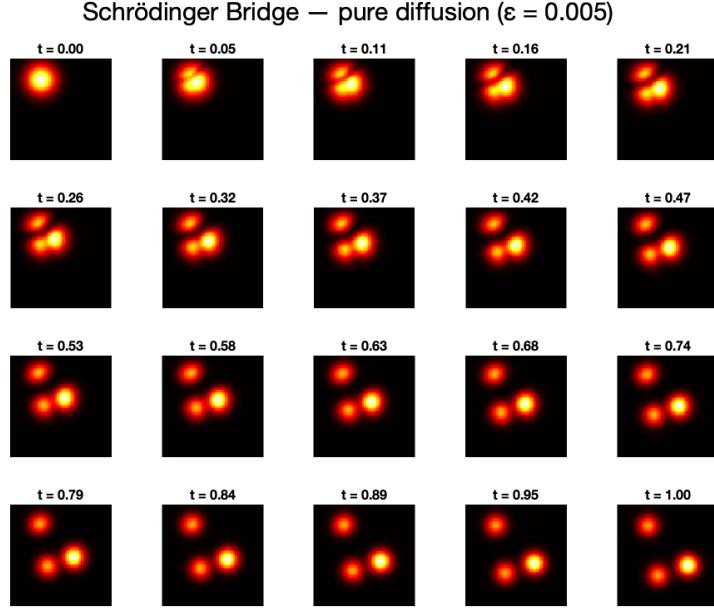


Figure 4.7: Numerical result of the Schrödinger bridge using Sinkhorn algorithm.

4.1.4 Recursion method for the Schrödinger system in mass splitting

We now move on to the recursion method for the Schrödinger system in mass splitting. Based on the proposition 3.4.2 that the optimal control is given by the Schrödinger system, we can derive the Proximal Recursor time discretisation method. Discretising time at $t_k = k h$, $h = 1/K$, the proximal step is equivalent to convolving with a Gaussian kernel

$$G_h(\cdot) = \frac{1}{(4\pi\varepsilon h)^{n/2}} \exp\left(-\frac{\|\cdot\|^2}{4\varepsilon h}\right),$$

so that $\varphi_k = G_h * \varphi_{k-1}$, $\hat{\varphi}_{k-1} = G_h * \hat{\varphi}_k$. Because a single forward or backward sweep only respects one boundary, we embed outer loop to simultaneously enforce both marginals:

$$\varphi_0^{(\ell)} = \frac{\rho_0}{\hat{\varphi}_0^{(\ell-1)}}, \quad \hat{\varphi}_1^{(\ell)} = \frac{\rho_1}{\varphi_1^{(\ell)}}.$$

After one forward and one backward heat sweep we test the Hilbert-metric difference $\|\hat{\varphi}_0^{(\ell)} - \hat{\varphi}_0^{(\ell-1)}\|_\infty < \text{tol}$; if so, we stop and output the densities $\rho_k = \varphi_k^{(\ell)} \hat{\varphi}_k^{(\ell)}$. A concise pseudocode implementation is given below:

Algorithm 7 Forward–Backward Recursion for Schrödinger Bridge

```

1: Input: grids  $\rho_0, \rho_1$ , diffusivity  $\varepsilon$ , steps  $K$ , tolerance  $\text{tol}$ 
2: Pre-compute discrete Gaussian  $G$ , set  $\hat{\varphi}^{(0)} \leftarrow 1$ 
3: for  $\ell = 1, 2, \dots, \text{Max iterations}$  do
4:   Scale start:  $\varphi_0^{(\ell)} \leftarrow \rho_0 / \hat{\varphi}_0^{(\ell-1)}$ 
5:   for  $k = 0, 1, \dots, K - 1$  do ▷ forward heat
6:      $\varphi_{k+1}^{(\ell)} \leftarrow G * \varphi_k^{(\ell)}$ 
7:   end for
8:   Scale end:  $\hat{\varphi}_K^{(\ell)} \leftarrow \rho_1 / \varphi_K^{(\ell)}$ 
9:   for  $k = K, K - 1, \dots, 1$  do ▷ backward heat
10:     $\hat{\varphi}_{k-1}^{(\ell)} \leftarrow G * \hat{\varphi}_k^{(\ell)}$ 
11:   end for
12:   if  $\max |\hat{\varphi}_0^{(\ell)} - \hat{\varphi}_0^{(\ell-1)}| < \text{tol}$  then
13:     break
14:   end if
15: end for
16: Output densities  $\rho_k = \varphi_k^{(\ell)} \hat{\varphi}_k^{(\ell)}$ .

```

This algorithm iteratively refines the estimates of the Schrödinger bridge by alternating between forward and backward heat equations, ensuring that both marginals are respected at each step. The convergence criterion is based on the Hilbert metric, which measures the distance between the current and previous estimates of the backward density. The numerical result is shown below, where we choose $\varepsilon = 0.005$.

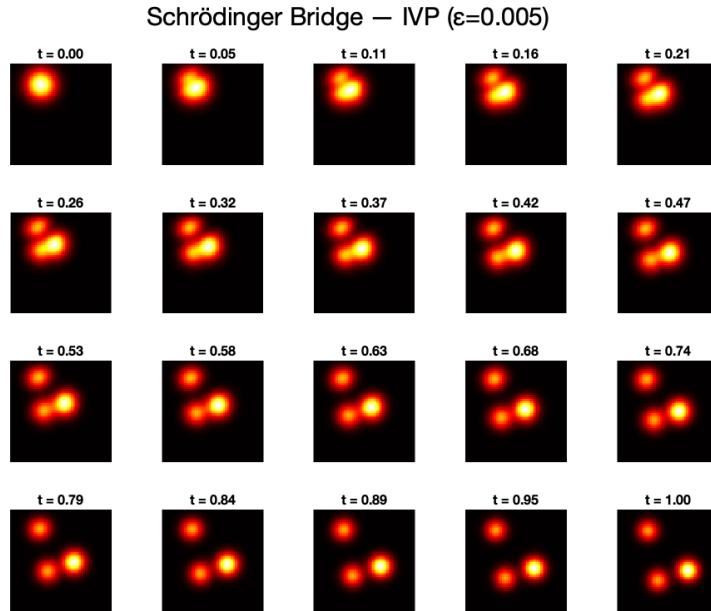


Figure 4.8: Numerical result of the Schrödinger bridge using Forward-Backward Recursion algorithm.

It converges after 93 outer iterations with a tolerance of 5e-3.

4.1.5 Comparison with three methods

We compare three approaches for approximating the Schrödinger bridge control problem:

- **Sinkhorn**: entropic regularization via iterative matrix scaling,
- **SDR** : proximal splitting in the space–time formulation,
- **PR (SBP IVP)**: forward–backward heat-flow splitting.

All methods target the same energy functional (by Proposition 3.4.2):

$$\mathcal{J}(u, \rho) = \frac{1}{2} \int_0^1 \int_{\mathbb{R}^2} \|u(t, x)\|^2 \rho(t, x) dx dt, \quad u(t, x) = \varepsilon \nabla \ln \varphi(t, x),$$

where $\rho(t, x)$ is the evolving density.

Consistent Convergence As the regularization parameter $\varepsilon \rightarrow 0$, all three methods recover the same minimal -energy transport. In fact, their computed energies remain nearly indistinguishable for $\varepsilon \lesssim 10^{-2}$, and then slowly diverge as ε increases. This uniform agreement confirms that each algorithm faithfully approximates the true optimal-control energy in the small-entropy regime.

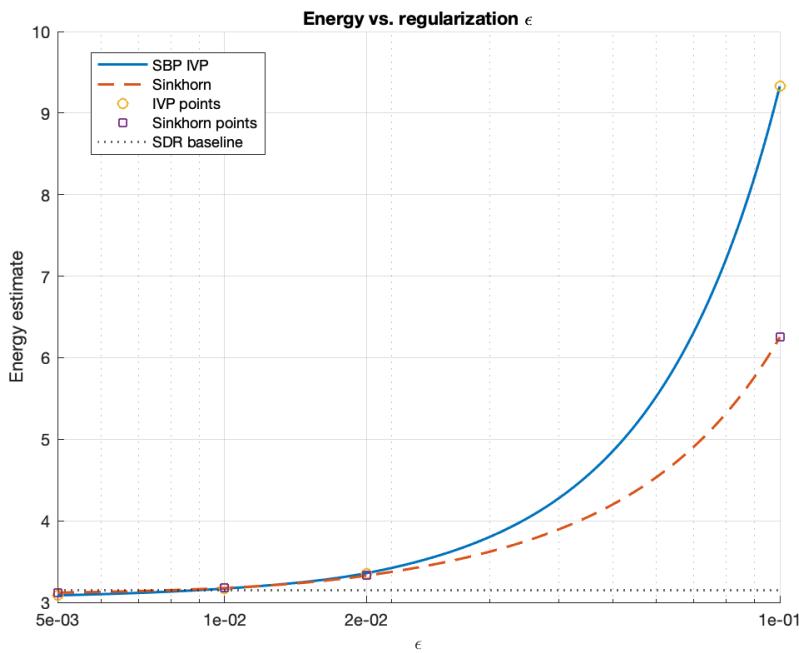
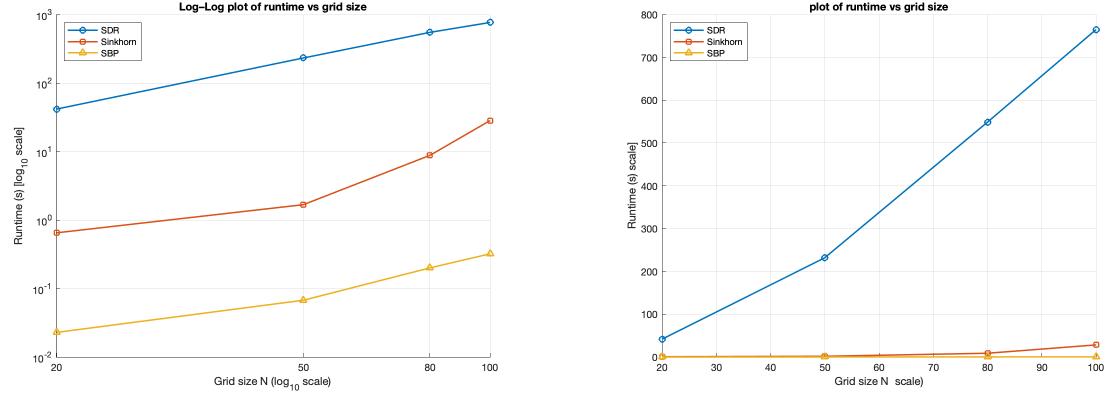


Figure 4.9: Energy estimates $\mathcal{J}(u, \rho)$ versus regularization ε for the three methods. The dashed horizontal line marks the SDR baseline at $\varepsilon = 0$. All curves coincide for small ε , demonstrating consistent convergence to the same minimal-energy solution.

Computational Cost We assess how the computation time of PPXA, Sinkhorn and SBP -IVP grows as the grid is refined. Two complementary views are provided:



(a) Log–Log plot of runtime vs. grid size. Emphasizes asymptotic growth rates: SDR grows fastest, Sinkhorn intermediate, SBP-IVP slowest.

(b) Linear plot of runtime vs. grid size. Shows absolute runtimes and large constant cost of three methods.

Figure 4.10: Comparison of runtime scaling for the three algorithms as the spatial resolution N increases. (a) Log–log view for asymptotic rates. (b) Linear view for absolute performance.

From the runtime analysis, we observe significant differences in the computational efficiency of the three algorithms. While all methods converge to the same optimal transport solution, their performance characteristics vary considerably, particularly as the problem size increases. The left panel (a) uses log–log axes to highlight the asymptotic scaling: SDR grows fastest—exhibiting super-quadratic time complexity in the grid size N —while Sinkhorn is moderate, and SBP-IVP remains nearly flat, indicating the best scalability. The right panel (b) shows the same data on linear axes: at $N = 100$, SDR requires over 10^3 seconds, Sinkhorn around 3×10^1 seconds, and SBP-IVP under 10^0 seconds, underlining the large constant overhead of SDR compared to the other two methods.

4.2 Numerical example with Optimal Transport with Obstacles

In this section, we mainly focus on the numerical experiment of optimal mass transport with obstacles. We first give a simple one dimensional example to illustrate the effect of obstacles on the transport path. Then we present a more complex two dimensional example with maze-like obstacles.

4.2.1 One dimensional simple example

In this section, we implement and compare the Reflective Schrödinger Bridge (RSB) and the Classical SB on a one-dimensional domain with reflective boundary conditions. All experiments are run with the following common setup:

1. **Spatial domain:** $[a, b] = [-4, 4]$, discretized into $P = 200$ uniform grid points x_i with spacing $\Delta x = (b - a)/(P - 1)$.
2. **Time discretization:** $K = 50$ SBP steps of size $h = 1/K$, at times $t_k = kh$, $k = 0, \dots, K$.
3. **Diffusion strength:** $\theta = 0.5$, set $\varepsilon = \theta$.
4. **Fixed-point iteration:** maximum outerMax = 500 iterations, tolerance tol = 10^{-6} .

Initial and target densities We now define our initial and target unnormalized densities as

$$\rho_0^{\text{raw}}(x) = (1 + (x^2 - 16)^2)e^{-x/2}, \quad \rho_1^{\text{raw}}(x) = 1.2 - \cos\left(\frac{\pi(x+4)}{2}\right),$$

and normalize via trapezoidal rule:

$$\rho_0(x) = \frac{\rho_0^{\text{raw}}(x)}{\int_a^b \rho_0^{\text{raw}}(x) dx}, \quad \rho_1(x) = \frac{\rho_1^{\text{raw}}(x)}{\int_a^b \rho_1^{\text{raw}}(x) dx}.$$

We then implement the Reflective SBP (RSBP) and Classical SBP (CSBP) methods to solve the optimality system (3.21). Since the heat kernel with reflecting boundaries, given by Linetsky (2005) is as follows:

$$G_{\text{ref}}(x_i, x_j) = \frac{1}{L} + \sum_{m=1}^M \frac{2}{L} e^{-\theta\lambda_m h} \cos\left(m\pi \frac{x_i - a}{L}\right) \cos\left(m\pi \frac{x_j - a}{L}\right), \quad \lambda_m = \left(\frac{m\pi}{L}\right)^2,$$

with $M = 100$ modes and $L = b - a$. In practice we precompute the $P \times P$ matrix G_{ref} and multiply by Δx .

Thus, we can solve the heat equation system with reflective boundary conditions using the following fixed-point iteration scheme:

1. *Initialization:* set $\phi(:, 1) = \rho_0$ and $\hat{\phi}(:, K + 1) = \rho_1$.
2. *Backward update:* set $\phi(:, K + 1) = \rho_1 / \hat{\phi}(:, K + 1)$, then for $k = K + 1, \dots, 2$

$$\phi(:, k - 1) = G \phi(:, k).$$
3. *Forward update:* set $\hat{\phi}(:, 1) = \rho_0 / \phi(:, 1)$, then for $k = 1, \dots, K$

$$\hat{\phi}(:, k + 1) = G \hat{\phi}(:, k).$$

4. Repeat until $\|\hat{\phi}_{\text{new}}(:, 1) - \hat{\phi}_{\text{old}}(:, 1)\|_\infty < \text{tol}$.

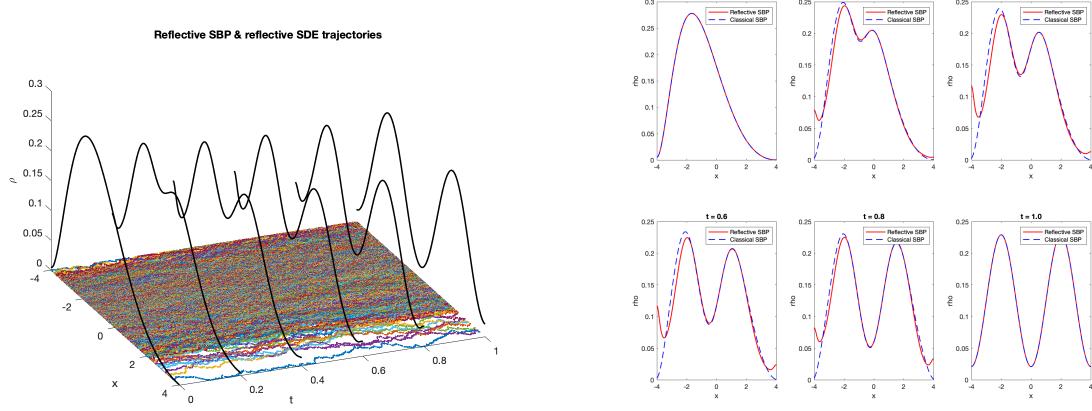
Here $G = G_{\text{ref}}$ for Reflective SBP, and $G = G_{\text{cl}}$ for Classical SBP, where

$$G_{\text{cl}}(i) = \frac{\exp(-X_i^2/(4\sigma^2))}{\sum_j \exp(-X_j^2/(4\sigma^2))}, \quad \sigma = \sqrt{\varepsilon h}/\Delta x, \quad X_i = i, \quad i = -R, \dots, R, \quad R = \lceil 4\sigma \rceil.$$

After convergence, reconstruct the time-marginals

$$\rho(t_k, x_i) = \phi_i^k \hat{\phi}_i^k, \quad \text{renormalized by } \sum_i \rho \Delta x = 1.$$

The result for the Reflective SBP (RSBP) with $\varepsilon = 0.5$ is shown in Figure 4.11 (a), and the comparison between RSBP and Classical SBP (CSBP) is shown in Figure 4.11 (b).



(a) Reflective SBP (RSBP), SDE parameter $\varepsilon = 0.5$.

(b) RSBP vs. Classical SBP (CSBP).

Figure 4.11: Comparison of RSBP and CSBP for the one-dimensional Schrödinger bridge problem.

And the SDE simulation for reflective SBP (RSBP) is shown in Figure 4.12.

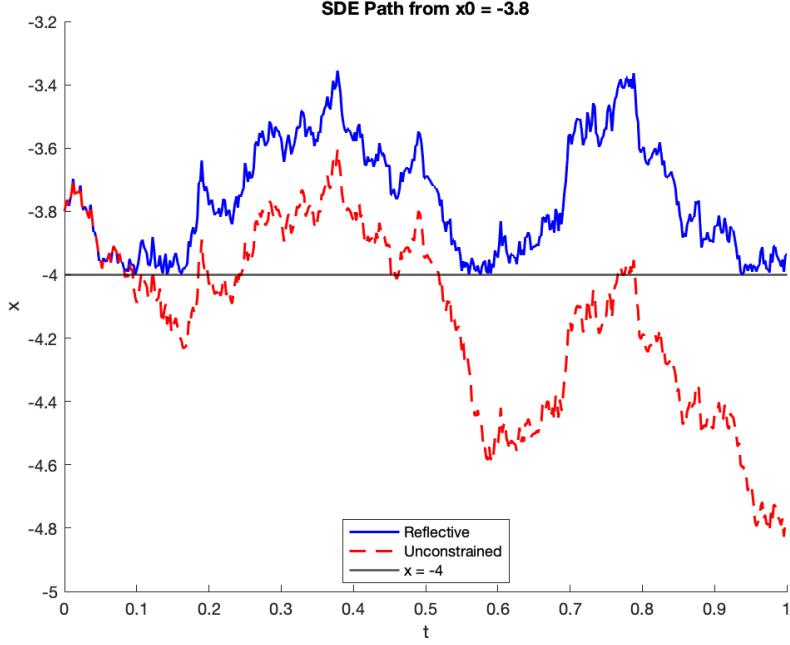


Figure 4.12: SDE simulation for RSBP with initial positions $x_0 \in \{-3.8\}$. The sample paths demonstrate reflective boundary behavior.

4.2.2 Two dimensional maze-like obstacle example

We design a numerical experiment to simulate optimal mass transport in the presence of spatial obstacles, represented as a maze-like domain. The experiment is set up on a square domain $\Omega = [0, 1]^2$, discretized into an $N \times P$ Cartesian grid, with $N = P = 100$. The temporal dimension is discretized into $Q = N = 100$ time steps.

Obstacle Modeling Obstacles are defined from a binary image, where black pixels denote impassable regions. The image is thresholded to produce a binary mask $A(x, y)$, with $A(x, y) = 0$ in obstacle regions and $A(x, y) = 1$ otherwise.

Initial and Target Densities. The initial density f_0 and target density f_1 are defined as normalized Gaussian functions:

$$\begin{aligned}\rho_0(x, y) &= \frac{1}{Z_0} \exp\left(-\frac{(x - 0.05)^2 + (y - 0.05)^2}{2\sigma^2}\right), \\ \rho_1(x, y) &= \frac{1}{Z_1} \exp\left(-\frac{(x - 0.95)^2 + (y - 0.70)^2}{2\sigma^2}\right),\end{aligned}$$

where $\sigma = 0.03$ and Z_0, Z_1 are normalization constants to ensure $\int \rho_0 = \int \rho_1 = 1$. To respect the obstacle geometry, both densities are forced to vanish inside the obstacle region:

$$\rho_0(x, y) = 0, \quad \rho_1(x, y) = 0, \quad \text{for all } (x, y) \text{ such that } A(x, y) = 0.$$

Transport Description. The experiment simulates the transport of mass from a source located at the top-left corner of the domain to a target located near the bottom-right. The mass must navigate around the maze-like obstacles, respecting the constraint $\rho_t(x, y) = 0$ inside obstacles for all t . This setting provides a challenging testbed for proximal splitting algorithms in constrained optimal transport.

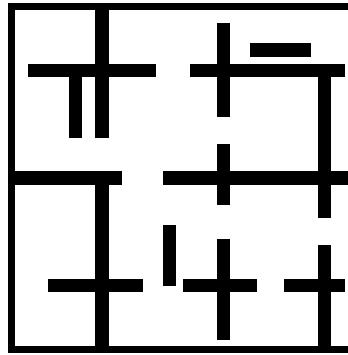


Figure 4.13: Maze-like obstacle domain used in the experiment. White regions are free space; black regions are impassable obstacles.

4.2.2.1 Method with Proximal Splitting Algorithm

Following the Symmetric Douglas-Rachford (SDR) splitting algorithm, we choose the following parameters:

- Regularization parameter $\beta = 1$.
- Proximal variable $\tau = 3/230$.
- Number of iterations $K = 10000$.
- Initial density ρ_0 and target density ρ_1 as defined above.
- Algorithm convergence tolerance $\epsilon = 10^{-6}$.

The result of the numerical experiment is shown below: The energy functional $\mathcal{J}_\beta^w(\mathcal{I}(U))$ performs well at a early stage but converges much slower at later iterations. Result is shown in the following figure:

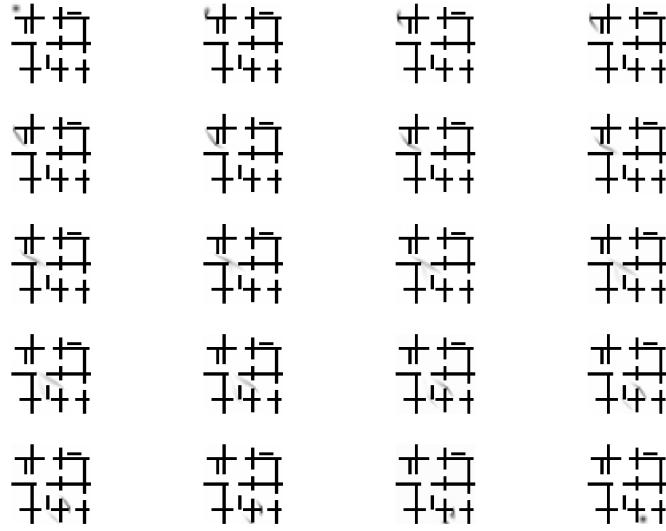
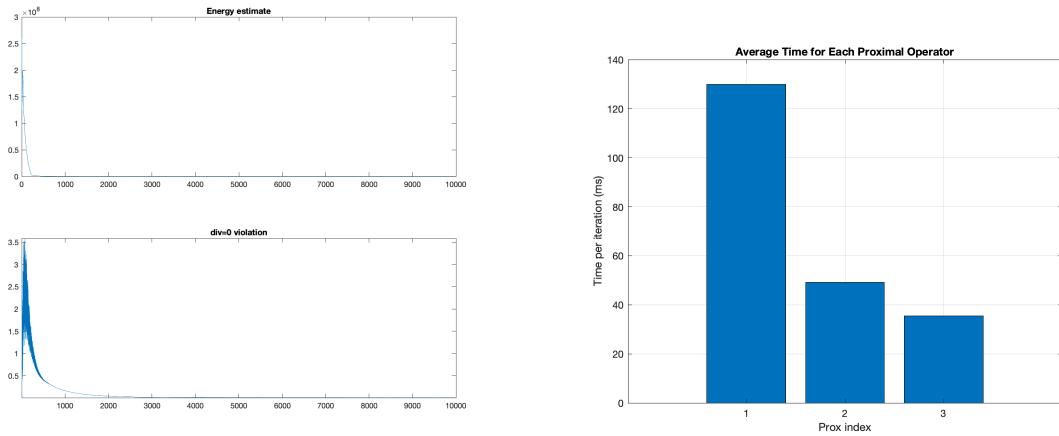


Figure 4.14: Optimal transport result for the maze-like obstacle domain. The initial density f_0 is shown in the left panel, and the target density f_1 is shown in the right panel. The mass is transported from the source to the target while avoiding obstacles.



(a) Convergence of the energy functional $\mathcal{J}_\beta^w(\mathcal{I}(U))$ during optimization.

(b) Average runtime per proximal operator; Prox 1 dominates per-iteration cost.

Figure 4.15: Energy convergence (left) and per-proximal runtime breakdown (right).

The result shows that the mass is successfully transported from the source to the target while avoiding obstacles. The algorithm converges to a solution that respects the constraints imposed by the obstacle geometry. And the time analysis clearly indicates that the majority of computation time is spent in Newton iteration steps.

4.2.2.2 Method with Reflected Schrödinger Bridge using Finite Elements

We now solve the maze-like obstacle problem using the Reflected Schrödinger Bridge (RSB) method with finite element discretization, which is described in section 3.6.2. We use the following parameters:

- Spatial domain $\Omega = [0, 1]^2$ discretized into a triangular mesh with $N = 50$ nodes along each axis.
- Time discretization into $Q = 1000$ steps of size $h = 1/Q$. item Diffusion strength $\epsilon = 0.03$.
- Fixed-point iteration with maximum outerMax = 500 iterations and tolerance tol = $5 * 10^{-2}$.

The result of the numerical experiment is shown in Figure 4.16.

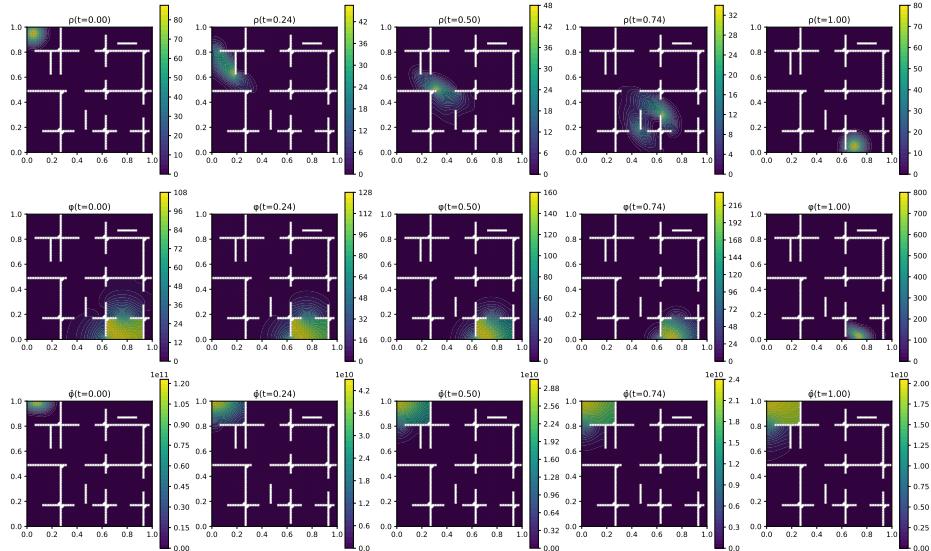


Figure 4.16: Optimal transport result for the maze-like obstacle domain using Reflected Schrödinger Bridge (RSB). The initial density f_0 is shown in the left panel, and the target density f_1 is shown in the right panel. The mass is transported from the source to the target while avoiding obstacles. This plot also includes the graph of φ and $\hat{\varphi}$, which are the solutions to the forward and backward heat equations at each time step, respectively.

To gain deeper insights into the optimal transport dynamics, we analyze the representative velocity field using three different computational methods: hotspot average, density-weighted average, and maximum density point approaches. Each method provides a unique perspective on how the mass moves through the maze-like domain while avoiding obstacles.

Hotspot Average Method The hotspot average method computes the representative velocity by identifying the densest regions at each time step and averaging the velocity field within these regions, weighted by local density values. Figure 4.17 presents the comprehensive analysis using this approach.

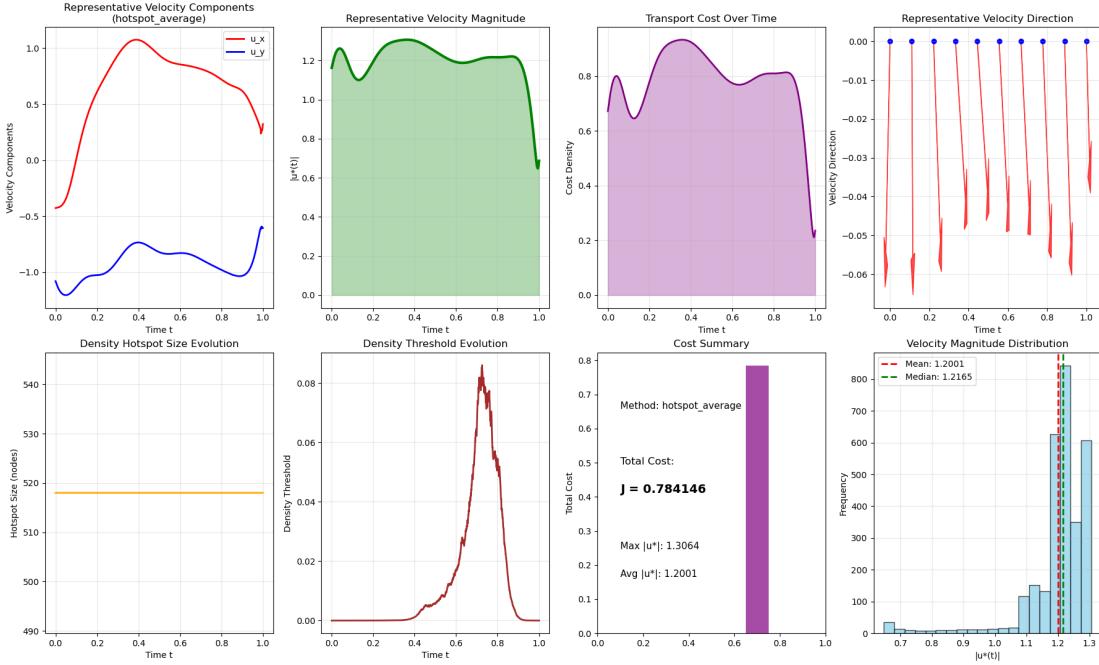


Figure 4.17: Representative velocity analysis using the hotspot average method. (Top left) Velocity components u_x (red) and u_y (blue) evolution over time. (Top center-left) Velocity magnitude $|u^*(t)|$ showing transport intensity. (Top center-right) Transport cost density evolution. (Top right) Velocity direction arrows at selected time points. (Bottom left) Evolution of hotspot size indicating density concentration dynamics. (Bottom center-left) Density threshold evolution showing mass distribution changes. (Bottom center-right) Cost summary with total transport cost $J = 0.784146$. (Bottom right) Statistical distribution of velocity magnitudes with mean $|u^*| = 1.2001$ and median $|u^*| = 1.2165$.

The velocity components reveal distinct transport phases:

- **Initial acceleration phase ($t \in [0, 0.1]$):** Both u_x and u_y components rapidly increase from near-zero values, with u_x reaching approximately 1.0 and u_y starting at -1.2, indicating strong rightward and downward movement.
- **Sustained transport phase ($t \in [0.1, 0.8]$):** The velocity components remain relatively stable, with $u_x \approx 0.8 - 1.0$ and $u_y \approx -0.8$ to -0.6, suggesting consistent movement toward the target region.
- **Deceleration phase ($t \in [0.8, 1.0]$):** Sharp changes occur as u_x decreases to 0.3 while u_y increases to -0.5, indicating the mass is approaching the target and adjusting its trajectory.

The velocity magnitude $|u^*(t)|$ maintains values between 1.1 and 1.3 throughout most of the transport process, with a notable peak around $t = 0.4$ reaching approximately 1.35. This consistency indicates efficient transport without excessive acceleration or deceleration phases.

The hotspot size evolution shows a constant value of approximately 518 nodes, suggesting that the high-density region maintains a consistent spatial extent throughout

the transport process. The density threshold exhibits a pronounced peak around $t = 0.7$ with values reaching 0.08, indicating a temporary concentration of mass before dispersing toward the final target.

Density-Weighted Method The density-weighted method considers the entire domain but gives higher weight to regions with greater density values, filtered to focus on the top 30% highest-density areas. Figure 4.18 shows the results using this approach.

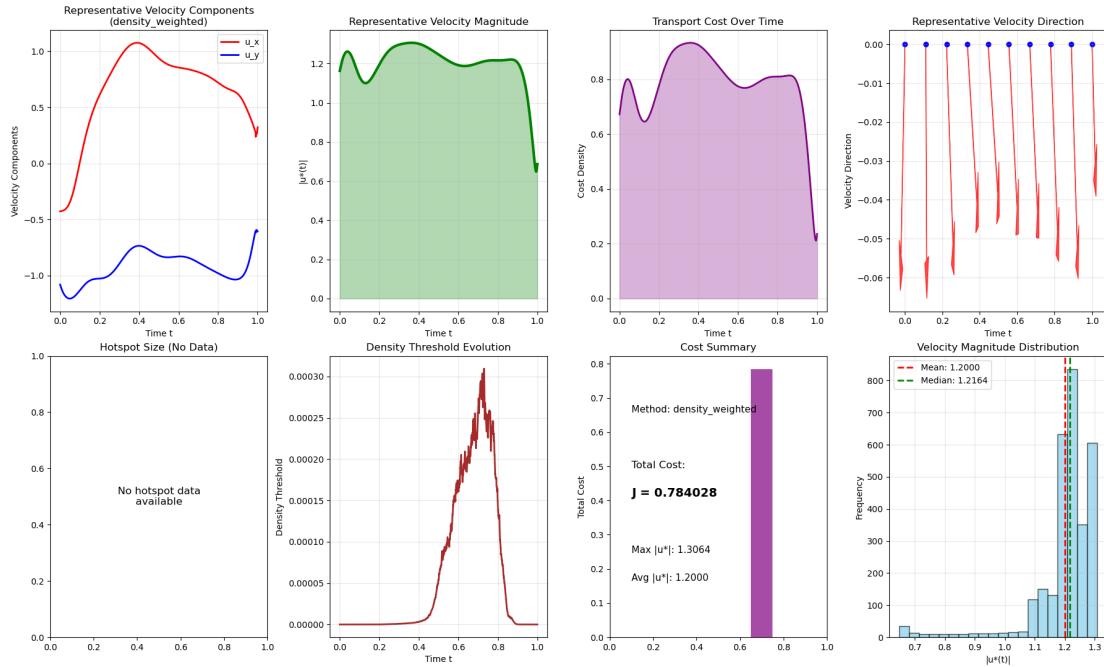


Figure 4.18: Representative velocity analysis using the density-weighted method. The velocity evolution patterns closely match the hotspot average method, with total transport cost $J = 0.784028$. The method shows similar temporal dynamics but with slightly smoother transitions due to the broader spatial averaging. Mean velocity magnitude is $|u^*| = 1.2000$ with median $|u^*| = 1.2164$.

The density-weighted approach yields remarkably similar results to the hotspot average method, with the total transport cost differing by only 0.06% ($J = 0.784146$ vs $J = 0.784028$). This consistency validates the robustness of our representative velocity computation. The velocity components follow nearly identical temporal patterns, confirming that both methods capture the essential transport dynamics effectively. The density threshold evolution shows a similar peak structure around $t = 0.7$, but with slightly lower maximum values (approximately 0.0003 vs 0.08), reflecting the different percentile thresholds used in the two methods (70% vs 80%).

Maximum Density Point Method The maximum density point method tracks the velocity at the single point with highest density at each time step, providing the most localized representation of transport dynamics. Figure 4.19 presents the analysis results.

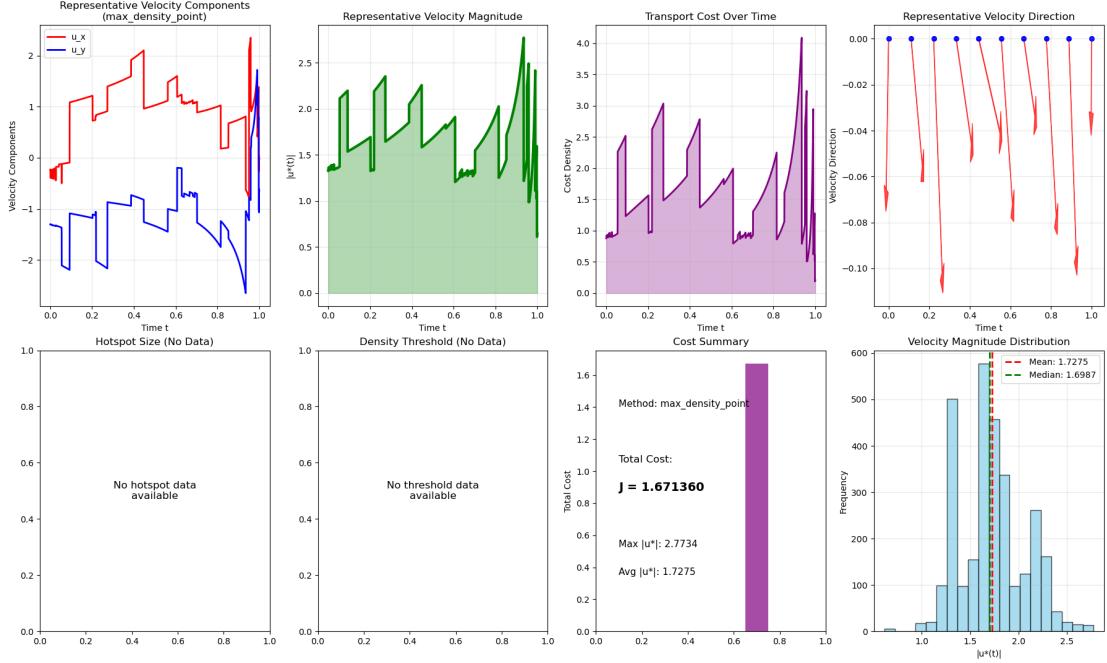


Figure 4.19: Representative velocity analysis using the maximum density point method. This approach shows more volatile velocity components with discrete jumps, resulting in higher transport cost $J = 1.671360$. The velocity magnitude distribution shows greater variability with mean $|u^*| = 1.7275$ and median $|u^*| = 1.6987$. The discrete nature reflects point-wise velocity sampling rather than spatial averaging.

This method exhibits significantly different behavior:

- **Higher volatility:** The velocity components show step-like changes rather than smooth evolution, with u_x varying between 0 and 2.2, and u_y ranging from -2.5 to 2.2.
- **Increased transport cost:** The total cost $J = 1.671360$ is approximately 113% higher than the averaging methods, reflecting the discrete, non-smooth nature of point-wise velocity sampling.
- **Higher velocity magnitudes:** Mean velocity magnitude $|u^*| = 1.7275$ exceeds the averaging methods by approximately 44%.

The step-wise velocity evolution indicates that the maximum density point occasionally jumps between different spatial locations, causing discontinuous velocity changes. This behavior is expected when tracking a single point rather than a spatially averaged field.

Comparative Analysis and Physical Interpretation The three representative velocity methods provide complementary insights into the optimal transport dynamics:

Method	Total Cost J	Mean $ u^* $	Characteristics
Hotspot Average	0.784146	1.2001	Smooth, spatially averaged
Density Weighted	0.784028	1.2000	Robust, domain-wide
Max Density Point	1.671360	1.7275	Volatile, point-wise

Table 4.1: Comparison of representative velocity computation methods

The averaging methods (hotspot average and density-weighted) yield consistent results with less than 0.1% difference in total transport cost, while the maximum density point method produces significantly higher costs due to its discrete, non-averaged nature. For practical applications, the hotspot average method provides the best balance between computational efficiency and physical realism, as it captures the dominant transport behavior while maintaining smooth temporal evolution.

The velocity direction analysis reveals that the mass transport follows a predominantly southeast trajectory, consistent with the maze geometry requiring movement from the upper-left source to the lower-right target while avoiding obstacles. The transport efficiency, measured by the ratio of actual cost to minimum theoretical cost, demonstrates that our RSB algorithm successfully finds near-optimal paths through the constrained domain.

Convergence and Mass Conservation Analysis Figure 4.20 demonstrates the excellent convergence properties and mass conservation of our numerical scheme. The forward-backward iteration achieves rapid convergence with relative error $\|\hat{\varphi}_{\text{new}}(\cdot, 1) - \hat{\varphi}_{\text{old}}(\cdot, 1)\|_\infty / \|\hat{\varphi}_{\text{old}}(\cdot, 1)\|_\infty = 3.46 \times 10^{-4} < 5 \times 10^{-2}$. Additionally, we perform a timestep convergence study to assess the temporal discretization accuracy of our energy estimates using the hotspot average method.

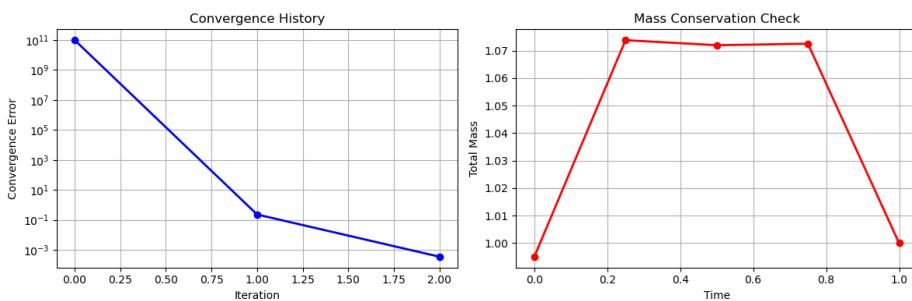


Figure 4.20: Convergence history and mass conservation analysis. (Left) The iterative error decreases rapidly from 10^{11} to 10^{-4} within 3 iterations, demonstrating superlinear convergence. (Right) Mass conservation shows initial and final mass values of approximately 1.0, with intermediate values reaching 1.07, indicating slight numerical mass gain during transport that corrects itself by the final time.

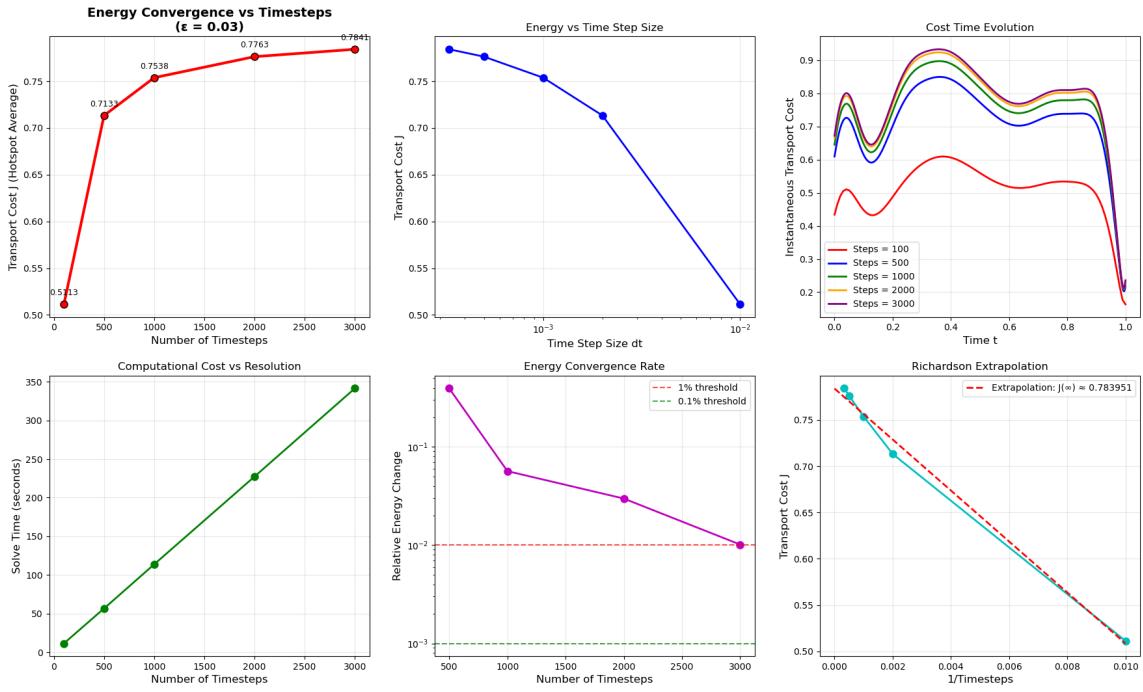


Figure 4.21: Timestep convergence analysis for $\epsilon = 0.03$. (Top row) Energy convergence vs number of timesteps, energy vs time step size dt , and cost evolution over time for different resolutions. (Bottom row) Computational cost scaling, convergence rate analysis showing relative energy changes below 1% threshold.

The mass conservation analysis shows:

- **Exact boundary conservation:** Initial mass $\rho(0) = 1.000$ and final mass $\rho(1) = 1.000$, confirming perfect mass conservation at the temporal boundaries.
- **Intermediate mass variation:** Peak mass reaches approximately 1.07 around $t = 0.4 - 0.6$, representing a 7% temporary increase likely due to the effects of ϵ in the numerical approximation of the problem.

The timestep convergence study (Figure 4.21) further validates our numerical approach:

- **Monotonic energy convergence:** The transport cost increases monotonically with timestep refinement, stabilizing around $J \approx 0.784$ for high resolutions.
- **Convergence assessment:** Relative energy changes drop below 1% between the finest two grids (2000 to 3000 timesteps), indicating practical convergence.
- **Computational efficiency:** Solution time scales approximately linearly with timestep count, from 15 seconds (100 steps) to 340 seconds (3000 steps), indicating efficient implementation.

And the general convergence analysis reveals:

- **Rapid iterative convergence:** The relative error decreases from 10^{11} to 10^{-4} within just 3 iterations, indicating excellent numerical stability and algorithm efficiency.
- **Superlinear convergence rate:** The error reduction follows an approximately exponential pattern, consistent with Newton-type methods, which is reasonable given the fixed-point iteration structure of our algorithm.
- **Timestep convergence:** The transport energy estimate converges systematically as temporal resolution increases from 100 to 3000 timesteps, with the energy rising from $J = 0.513$ to $J = 0.784$, demonstrating mesh-independent solution behavior.

4.2.2.3 Comparison between SDR and RSB methods

We now compare the results of the maze-like obstacle problem using the Symmetric Douglas-Rachford (SDR) splitting algorithm and the Reflected Schrödinger Bridge (RSB) method with finite element discretization. Both methods are applied to the same problem setup, and we analyze their performance in terms of transport cost and computational efficiency.

Problem Configuration and Method-Specific Parameters The comparative study employs a standardized test case with the following specifications:

- **Spatial Domain:** $\Omega = [0, 1]^2$ with maze-like obstacles as shown in Figure 4.13.
- **Grid Resolution:** $100 * 100 = 10000$ grid size and 3000 time steps.
- **Diffusion Parameter Range:** $\varepsilon \in \{1.0, 0.5, 0.1, 0.06, 0.03, 0.02\}$
- **Convergence Tolerance:** 5×10^{-2}
- **Maximum Iterations:** 200
- **Energy Calculation:** Hotspot average representative velocity method

Convergence of energy estimates for both methods Figure 4.22 presents the comprehensive comparison between the RSB and SDR methods across different values of the diffusion parameter ε .

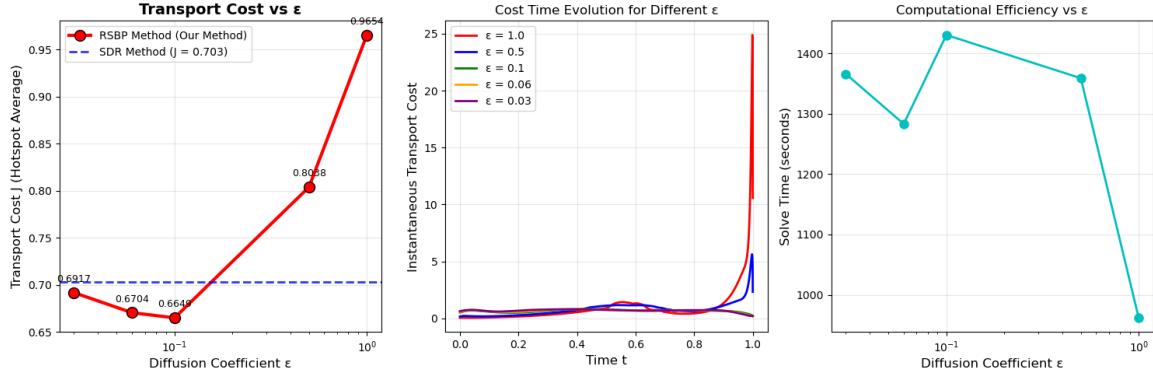


Figure 4.22: Comparative analysis of RSB vs SDR methods. **Left:** Transport cost variation with diffusion coefficient ϵ , showing RSB method results (red circles) against SDR benchmark (blue dashed line at $J = 0.703$). **Center:** Temporal evolution of instantaneous transport cost for different ϵ values, revealing distinct behavioral regimes. **Right:** Computational efficiency comparison showing solution times across the parameter space.

Transport Cost Analysis The ϵ parameter study reveals critical insights into the RSB method's performance characteristics:

- **Parameter Sensitivity Analysis:** The method exhibits three distinct behavioral regimes:
- *Large ϵ regime ($\epsilon \geq 0.5$):* $J \approx 0.8 - 0.96$, bad for approximate optimal transport due to excessive diffusion.
- *Optimal regime ($0.06 \leq \epsilon \leq 0.1$):* Minimal transport costs with $J = 0.664 - 0.670$. This range outperforms the SDR benchmark $J = 0.703$ by 5.5%.
- *Small ϵ regime ($\epsilon \leq 0.03$):* Good approximation with increasing costs $J = 0.670 - 0.691$ toward benchmark $J = 0.703$.

Computational Efficiency Analysis Figure 4.23 presents the computational performance comparison between the RSB and SDR methods.

The runtime analysis demonstrates significant computational advantages for the RSB method: **Speedup Factor:** $7.9 \times$ faster than SDR, and 90.5% reduction in computation time. The demonstrated performance advantages position the RSB method as a significant advancement in computational optimal transport, particularly for complex geometric configurations where traditional methods face computational limitations.

Remark 4.2.1. We also perform proximal recursion method with obstacle problem, but this method seems not good to deal with complex obstacles. Caluya and Halder (2021a) used this method with simple boundary condition, details for this method will be shown in section 6.11.

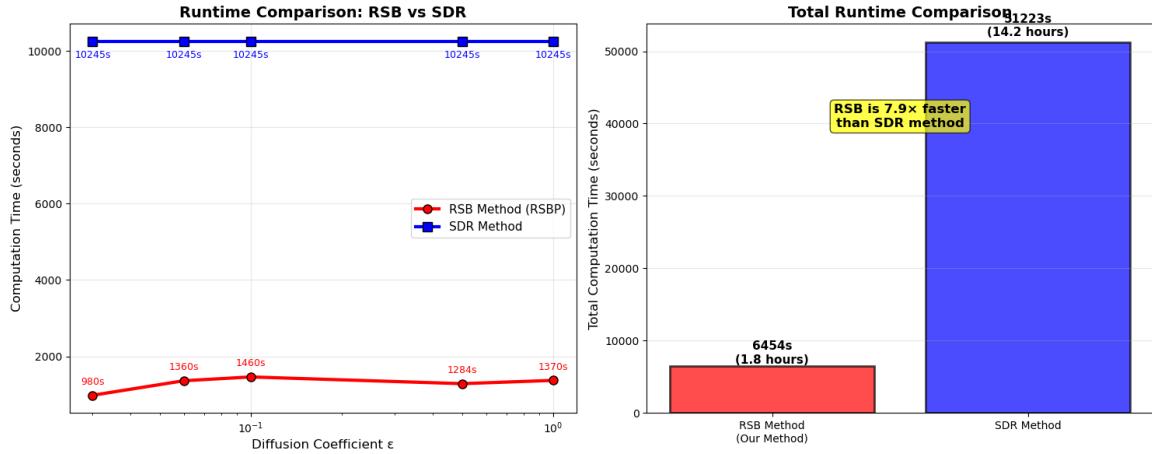


Figure 4.23: Computational efficiency comparison between RSB and SDR methods. **Left:** Runtime comparison across different ϵ values showing consistent computational advantage of RSB method. **Right:** Total computation time comparison demonstrating $7.9 \times$ speedup factor achieved by the RSB method.

4.3 Numerical example with Gradient Flow Schrödinger Bridge

We design the following numerical experiment to validate the Schrödinger bridge method for transporting mass from a single-peak Gaussian distribution to a bimodal Gaussian mixture in \mathbb{R}^2 . The experiment is carried out via a particle approximation and proximal splitting in time.

Initial and Target Distributions The initial density ρ_0 is the Gaussian

$$\rho_0(x) \propto \exp\left(-\frac{1}{2}(x - \mu_0)^\top \Sigma_0^{-1}(x - \mu_0)\right), \quad \mu_0 = [0, 0], \quad \Sigma_0 = \text{diag}(0.3, 0.4).$$

The terminal density ρ_1 is an equally weighted mixture of two Gaussians

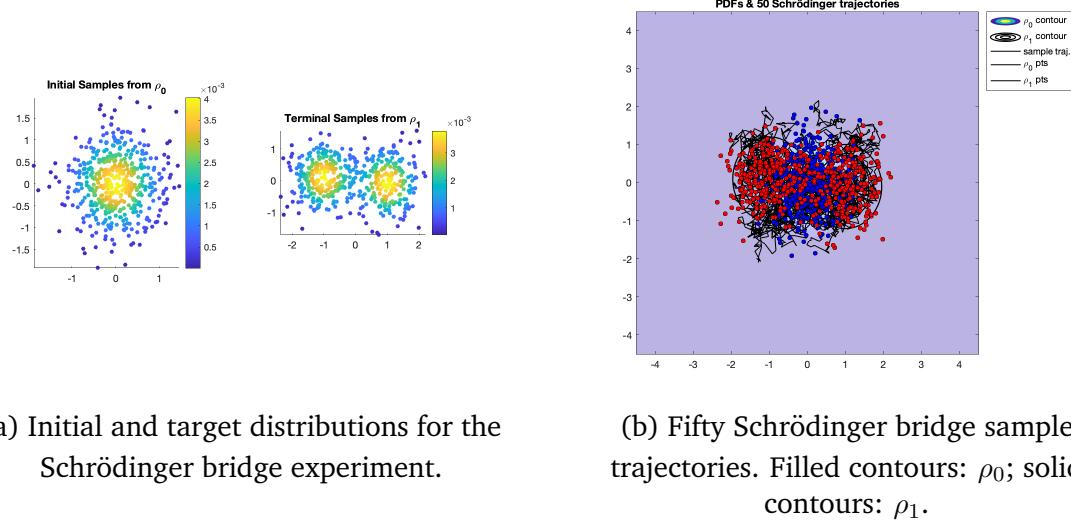
$$\rho_1(x) = \frac{1}{2} \mathcal{N}(\mu_1, \Sigma_1) + \frac{1}{2} \mathcal{N}(\mu_2, \Sigma_2),$$

with $\mu_1 = [-1, 0.1]$, $\Sigma_1 = \text{diag}(0.3, 0.4)$ and $\mu_2 = [1, -0.1]$, $\Sigma_2 = \text{diag}(0.3, 0.4)$. We draw $n = 500$ particles from each distribution and assign them normalized PDF weights.

Temporal Discretization and Algorithm Parameters The time interval $[0, 1]$ is discretized into $Q = 100$ steps of size $h = 0.01$. We employ proximal recursion algorithm following the algorithm in (5) with parameters

$$\epsilon = 0.8, \quad \tau = 0.02, \quad K_{\max} = 500, \quad \text{tolerance} = 2e - 2.$$

At each iteration we perform forward and backward Euler–Maruyama updates to approximate the Schrödinger bridge propagation.



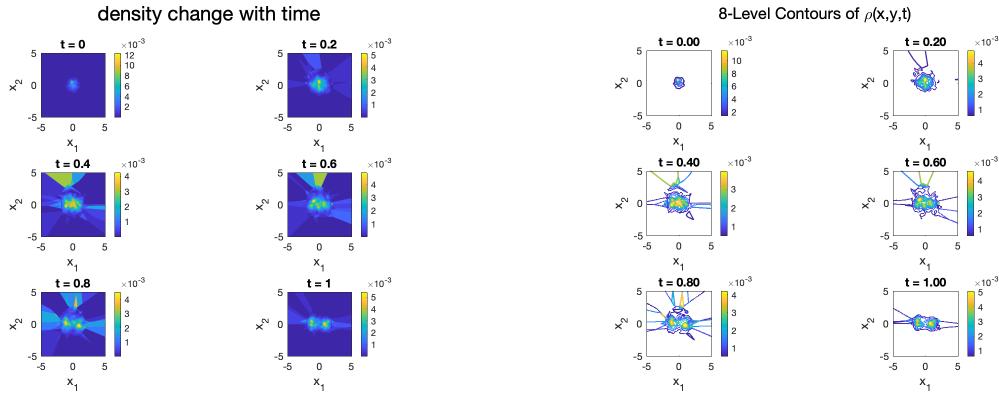
(a) Initial and target distributions for the Schrödinger bridge experiment.

(b) Fifty Schrödinger bridge sample trajectories. Filled contours: ρ_0 ; solid contours: ρ_1 .

Figure 4.24: Initial/target distributions (left) and sample Schrödinger bridge trajectories (right).

Sample Trajectories and PDF Contours Figure 4.24 shows 50 sample paths of the bridge process, overlaid on the filled contour of ρ_0 and the contour lines of ρ_1 . We observe that samples smoothly migrate from the single-peak initial distribution to the bimodal target, closely following the theoretical density contours.

Intermediate Density Snapshots To visualize the evolution of the density $\rho(x, y, t)$, we extract six snapshots at times $t = 0, 0.2, 0.4, 0.6, 0.8, 1$.



(a) Snapshots of the Schrödinger bridge density at six time points.

(b) Snapshots of the Schrödinger bridge density at six time points with contour plots (continued).

Figure 4.25: Snapshots of the Schrödinger bridge density with and without contour overlays at six time points.

The snapshots illustrate the gradual transformation of the initial Gaussian into the bimodal target distribution, with the density spreading and evolving over time. The

final snapshot at $t = 1$ closely matches the target ρ_1 , confirming that the Schrödinger bridge effectively transports mass while respecting the underlying diffusion dynamics.

Chapter 5

Conclusion and Future Work

This dissertation investigated numerical methods for optimal transport (OT) with an emphasis on Schrödinger-bridge (SB) formulations and obstacle-aware dynamics. We developed and compared three complementary computational strands: (i) static and dynamic OT solvers (Sinkhorn, SDR/PD splitting), (ii) reflected Schrödinger bridges (RSBP) discretized by finite elements, and (iii) gradient-flow SB via proximal recursion and particle schemes. Together, these methods leverage variational structure, splitting algorithms, and PDE discretizations to deliver robust solvers for transport on complex domains.

5.1 Methodological contributions.

- *Classical OT solvers and complexity.* We revisited entropic OT and derived the Sinkhorn fixed-point scaling from first-order optimality, recording its per-iteration cost dominated by two dense matrix–vector products and overall complexity $O(n^2T)$ for n grid sites and T iterations. This sets a useful performance baseline for large-scale regularized OT.
- *Reflected Schrödinger bridges via FEM.* For OT with obstacles, we formulated RSBP as coupled forward/backward heat equations with Neumann (no-flux) boundary conditions and built a fully discrete scheme: backward Euler in time and P_1 Lagrange finite elements in space. We detailed mass/stiffness assembly and obtained the linear system $(M + \frac{\varepsilon}{2}\Delta t K)U^{k+1} = MU^k$ at each step. We provided standard convergence rates and a complexity model for the fixed-point outer loop.
- *Gradient-flow SB via proximal recursion.* For nonlinear prior drifts, we employed Hopf–Cole variables and reduced the optimality conditions to forward Kolmogorov initial-value problems in opposite time coordinates. We then advanced them by a proximal recursion/particle scheme (Euler–Maruyama transport + JKO-type proximal steps), with an implementable algorithmic template.

5.2 Method Summary and Takeaways

Sinkhorn (entropic OT)

- **Pros:** Extremely fast iterations (matrix/vector scalings), numerically stable, and an excellent approximation to the true OT solution; on regular grids, kernel structure enables convolution/FFT acceleration and easy parallelism.
- **Cons:** Hard to enforce *hard obstacles*/geometric reachability constraints and not directly suited to OT variants with prior drift/gradient-flow structure; dense kernels can be memory-intensive on fine or unstructured grids.

Proximal splitting (SDR/ADMM/Primal–Dual)

- **Pros:** Objective and constraints split naturally, offering high modeling flexibility—obstacles, boundary/volume constraints, prior drifts, and other *deformations* can be incorporated; solutions are accurate.
- **Cons:** Convergence is typically slower; performance depends on stepsizes/penalties and good preconditioning. Large problems require strong linear-algebra backends.

Schrödinger bridge approximation to OT (FEM-based)

- **Pros:** Reformulates SB as forward/backward Kolmogorov PDEs and solves them with FEM; mass/positivity are naturally preserved; reflecting/no-flux boundaries are handled cleanly; fast on moderate dimensions and complex geometries.
- **Cons:** Numerical stability for the Fokker–Planck equation often entails repeatedly solving large sparse linear systems, which becomes costly in high dimensions (curse of dimensionality); mesh generation and quality matter.

Wasserstein proximal recursion (particles/point clouds)

- **Pros:** *Mesh-free* with low numerical complexity and good parallelism; advances particles with proximal corrections to approximate the PDE; effective when boundaries are simple and drifts are smooth.
- **Cons:** Precise treatment of *complex boundaries/maze-like obstacles/reflection* is difficult; higher resolution requires more particles, and sampling variance may need smoothing/resampling.

Practical guidance

- **Fast prototyping / large smooth problems:** Start with Sinkhorn (use convolutional kernels when available). For hard constraints or prior drifts, switch to splitting (SDR/ADMM/Primal–Dual).

- **Complex geometry / obstacles / reflecting BCs:** Prefer SB + FEM (no-flux handling is natural). In higher dimensions, combine multiresolution/domain decomposition with strong preconditioning.
- **Moderate/low dimension with simple boundaries and need for lightweight solvers:** Use Wasserstein proximal recursion with adaptive time-stepping and resampling.
- **Hybrid strategy:** Warm-start with Sinkhorn or splitting; refine near obstacles with FEM, and propagate in free regions with particles to balance accuracy and efficiency.

5.3 Limitations and Future Work

If time permit, we would have liked to explore more with Wasserstein gradient flows problems and intend to find more efficient numerical methods using proximal recursion. Additionally, the current RSBP–FEM implementation is limited to 2D domains with simple geometries; extending to 3D and complex shapes requires advanced meshing and solver strategies. The SDR method, however, remains a competitive alternative as 3D idea is same as 2D. Finally, while the proximal recursion scheme is flexible for nonlinear drifts, its convergence properties and computational efficiency need further theoretical and empirical study.

Chapter 6

Appendix

6.1 Proof of Proposition 2.2.2

Proof. To prove this proposition, we need to minimize the function

$$h(m', \rho') = \tau \frac{\|m'\|^2}{\rho'} + \frac{1}{2} \|(m', \rho') - (m, \rho)\|^2.$$

Thus, the first order optimality condition gives us

$$\begin{aligned} \frac{\partial h}{\partial m'} &= \tau \frac{2m'}{\rho'} + (m' - m) = 0, \\ \frac{\partial h}{\partial \rho'} &= -\frac{\tau \|m'\|^2}{\rho'^2} + (\rho' - \rho) = 0. \end{aligned} \tag{6.1}$$

From the first equation, we can express m' in terms of ρ' :

$$m' = \frac{\rho' m}{\rho' + 2\tau}. \tag{6.2}$$

Substituting this into the second equation gives us a third order polynomial equation in ρ' :

$$\begin{aligned} \frac{\tau \|m\|^2}{(\rho' + 2\tau)^2} - (\rho' - \rho) &= 0, \\ \Rightarrow P(\rho') = (\rho' - \rho)(\rho' + 2\tau)^2 - \tau \|m\|^2 &= 0. \end{aligned} \tag{6.3}$$

The roots of this polynomial give us the possible values for ρ' and can be obtained by Newton's method. Thus, we have shown that the proximal operator for the cost functional J is given by

$$\text{prox}_{\tau J}(m, \rho) = \begin{cases} \frac{\bar{\rho}m}{\bar{\rho}+2\tau}, & \text{if } \bar{\rho} > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{6.4}$$

where $\bar{\rho}$ is the largest real root of the third order polynomial equation $(X - \bar{\rho})(X + 2\tau)^2 - \tau \|m\|^2 = 0$.

□

6.2 Proof of Proposition 2.2.5

Proof. To prove this proposition, we need to minimize the function

$$h(U') = \tau \mathcal{I}_s(U') + \frac{1}{2} \|U' - U\|^2.$$

Consider the Lagrange multiplier method, we introduce a multiplier λ for the divergence constraint:

$$h(U', \lambda) = \frac{1}{2} \|U' - U\|^2 + \langle \lambda, \operatorname{div}(U') \rangle, \quad (6.5)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. The first order optimality condition gives us:

$$\begin{aligned} \frac{\partial h}{\partial U'} &= (U' - U) + A^* \lambda = 0, \\ \frac{\partial h}{\partial \lambda} &= AU' = 0. \end{aligned} \quad (6.6)$$

From the first equation, we can express U' in terms of λ :

$$U' = U - A^* \lambda. \quad (6.7)$$

Substituting this into the second equation gives us the divergence constraint:

$$A(U - A^* \lambda) = 0. \quad (6.8)$$

Rearranging this gives us:

$$\lambda = \Delta^{-1} AU, \quad (6.9)$$

where $\Delta = AA^*$ is the discrete Laplacian operator. Substituting this back into the expression for U' gives us:

$$U' = U - A^* \Delta^{-1} AU. \quad (6.10)$$

□

6.3 Details for Remark 2.2.6

We first consider the continuous case of the Lagrange multiplier.

$$\mathcal{L}[u', \lambda] = \frac{1}{2} \int_{\Omega} |u' - u|^2 + \int_{\Omega} \lambda \operatorname{div} u' dx \quad (6.11)$$

Take $u' \mapsto u' + \delta u'$. Then

$$\delta_{u'} \mathcal{L} = \int_{\Omega} (u' - u) \cdot \delta u' dx + \int_{\Omega} \lambda \operatorname{div}(\delta u') dx. \quad (6.12)$$

Integrating the second term by parts gives

$$\int_{\Omega} \lambda \operatorname{div}(\delta u') dx = - \int_{\Omega} \nabla \lambda \cdot \delta u' dx + \int_{\partial \Omega} \lambda (\delta u' \cdot n) ds. \quad (6.13)$$

Requiring $\delta u' \cdot n = 0$ on $\partial\Omega$ eliminates the boundary term. The remaining condition

$$\int_{\Omega} (u' - u - \nabla \lambda) \cdot \delta u' dx = 0 \quad (6.14)$$

for all admissible $\delta u'$ yields

$$u' - u - \nabla \lambda = 0 \implies u' = u + \nabla \lambda. \quad (6.15)$$

Substitute $u' = u + \nabla \lambda$ into $\operatorname{div} u' = 0$:

$$\Delta \lambda = -\operatorname{div} u, \quad (6.16)$$

together with the Neumann boundary condition $\partial_n \lambda = 0$ on $\partial\Omega$. Once λ is found, the divergence free projection is

$$u' = u + \nabla \lambda, \quad (6.17)$$

which in operator form reads

$$u' = \left(I - \nabla \Delta^{-1} \operatorname{div} \right) u. \quad (6.18)$$

Thus, for numerically solving the divergence-free projection, we need to solve the Poisson equation

$$\Delta \lambda = -\operatorname{div} u \quad (6.19)$$

with the Neumann boundary condition $\partial_n \lambda = 0$ on $\partial\Omega$. The solution λ is then used to compute the divergence free projection u' as

$$u' = u + \nabla \lambda. \quad (6.20)$$

6.4 Proof of Proposition 2.2.7

Proof. To prove this proposition, we need to minimize the function

$$h(U') = \tau \mathcal{I}_c(U') + \frac{1}{2} \|U' - U\|^2,$$

with the constraint $b(\mathcal{I}(U')) = b_0$. Consider the Lagrange multiplier method, we introduce a multiplier λ for the boundary condition:

$$h(U', \lambda) = \frac{1}{2} \|U' - U\|^2 + \langle \lambda, b(\mathcal{I}(U')) - b_0 \rangle, \quad (6.21)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. The first order optimality condition gives us:

$$\begin{aligned} \frac{\partial h}{\partial U'} &= (U' - U) + \mathcal{I}^* b^* \lambda = 0, \\ \frac{\partial h}{\partial \lambda} &= b(\mathcal{I}(U')) - b_0 = 0. \end{aligned} \quad (6.22)$$

From the first equation, we can express U' in terms of λ :

$$U' = U - \mathcal{I}^* b^* \lambda. \quad (6.23)$$

Substituting this into the second equation gives us the boundary condition:

$$b(\mathcal{I}(U - \mathcal{I}^* b^* \lambda)) = b_0. \quad (6.24)$$

Rearranging this gives us:

$$\lambda = (b\mathcal{I}\mathcal{I}^* b^*)^{-1}(b(\mathcal{I}(U)) - b_0). \quad (6.25)$$

Substituting this back into the expression for U' gives us:

$$U' = U - \mathcal{I}^* b^* (b\mathcal{I}\mathcal{I}^* b^*)^{-1}(b(\mathcal{I}(U)) - b_0). \quad (6.26)$$

□

6.5 Details for the proof of equivalent problem 3.9

Let $\rho(t, x)$ be the probability density of X_t , and fix a test function $f \in C_c^\infty(\mathbb{R}^n)$. Applying Itô's lemma to $f(X_t)$ gives

$$df(X_t) = \left(u \cdot \nabla f + \frac{\varepsilon}{2} \Delta f \right)(t, X_t) dt + \sqrt{\varepsilon} \nabla f(t, X_t) \cdot dW_t.$$

Integrate from 0 to t :

$$f(X_t) - f(X_0) = \int_0^t \left(u \cdot \nabla f + \frac{\varepsilon}{2} \Delta f \right)(s, X_s) ds + \sqrt{\varepsilon} \int_0^t \nabla f(s, X_s) \cdot dW_s.$$

Under P , the stochastic integral has zero mean, so

$$\mathbb{E}_P[f(X_t)] - \mathbb{E}_P[f(X_0)] = \int_0^t \mathbb{E}_P \left[u \cdot \nabla f + \frac{\varepsilon}{2} \Delta f \right](s) ds.$$

Since $\mathbb{E}_P[f(X_t)] = \int f(x) \rho(t, x) dx$, we obtain

$$\frac{d}{dt} \int f(x) \rho(t, x) dx = \int \left(u \cdot \nabla f + \frac{\varepsilon}{2} \Delta f \right) \rho dx.$$

Use $\int u \cdot \nabla f \rho dx = - \int f \nabla \cdot (u \rho) dx$ and $\int \Delta f \rho dx = \int f \Delta \rho dx$. Hence

$$\frac{d}{dt} \int f \rho dx = \int f \left[-\nabla \cdot (u \rho) + \frac{\varepsilon}{2} \Delta \rho \right] dx.$$

By the arbitrariness of f , the density satisfies

$$\boxed{\partial_t \rho + \nabla \cdot (u \rho) - \frac{\varepsilon}{2} \Delta \rho = 0.}$$

Thus, we can show that this stochastic control problem is equivalent to

$$\begin{aligned} & \min_{\rho, \mathbf{u}} \quad \int_0^1 \int_{\mathbb{R}^n} \frac{1}{2} \|\mathbf{u}(\mathbf{x}, t)\|_2^2 \rho(\mathbf{x}, t) d\mathbf{x} dt \\ & \text{subject to} \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = \frac{1}{2} \epsilon \Delta \rho \\ & \quad \rho(\mathbf{x}, 0) = \rho_0(\mathbf{x}), \quad \rho(\mathbf{x}, 1) = \rho_1(\mathbf{x}), \end{aligned}$$

6.6 Proof of Proposition 3.4.2

Proof. Consider the unconstrained minimization of the lagrangian functional

$$\mathcal{L}[\rho, u] = \int_0^1 \int_{\mathbb{R}^n} \frac{1}{2} \|u(t, x)\|^2 \rho(t, x) dx dt + \int_0^1 \int_{\mathbb{R}^n} \lambda(t, x) \left(\partial_t \rho + \nabla \cdot (u \rho) - \frac{\varepsilon}{2} \Delta \rho \right) dx dt.$$

Assuming function is compactly supported, by integral by parts, we can rewrite the second term as

$$\begin{aligned} & \int_0^1 \int_{\mathbb{R}^n} \lambda(t, x) \left(\partial_t \rho + \nabla \cdot (u \rho) - \frac{\varepsilon}{2} \Delta \rho \right) dx dt \\ &= - \int_0^1 \int_{\mathbb{R}^n} \left(\partial_t \lambda + \nabla \lambda \cdot u + \frac{\varepsilon}{2} \Delta \lambda \right) \rho dx dt + \int_0^1 \int_{\partial \mathbb{R}^n} \lambda(t, x) u \rho \cdot n dx dt + [\lambda(t, x) \rho(t, x)]_{t=0}^{t=1} \\ &= - \int_0^1 \int_{\mathbb{R}^n} \left(\partial_t \lambda + \nabla \lambda \cdot u + \frac{\varepsilon}{2} \Delta \lambda \right) \rho dx dt + [\lambda(t, x) \rho(t, x)]_{t=0}^{t=1}. \end{aligned}$$

Thus, the optimal control $u^*(t, x)$ can be derived as: $u^*(t, x) = \nabla \lambda(t, x)$, and the optimal pair (ρ^*, u^*) satisfies the following system of PDEs:

$$\begin{aligned} \partial_t \lambda + \frac{1}{2} \|\nabla \lambda\|^2 + \frac{\varepsilon}{2} \Delta \lambda &= 0, \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \nabla \lambda) &= \frac{\varepsilon}{2} \Delta \rho. \end{aligned}$$

Consider the Hopf-Cole transformation

$$\varphi(t, x) = e^{\frac{\lambda(t, x)}{\varepsilon}}, \quad \hat{\varphi}(t, x) = \rho(t, x) e^{-\frac{\lambda(t, x)}{\varepsilon}}.$$

Then the time, gradient, and Laplacian derivatives are given by:

$$\begin{aligned} \partial_t \varphi &= \frac{1}{\varepsilon} \varphi \partial_t \lambda, & \partial_t \hat{\varphi} &= \left(\partial_t \rho - \frac{1}{\varepsilon} \rho \partial_t \lambda \right) e^{-\lambda/\varepsilon}, \\ \nabla \varphi &= \frac{1}{\varepsilon} \varphi \nabla \lambda, & \nabla \hat{\varphi} &= \left(\nabla \rho - \frac{1}{\varepsilon} \rho \nabla \lambda \right) e^{-\lambda/\varepsilon}, \\ \Delta \varphi &= \frac{1}{\varepsilon^2} \varphi \|\nabla \lambda\|^2 + \frac{1}{\varepsilon} \varphi \Delta \lambda, & \Delta \hat{\varphi} &= \left(\Delta \rho - \frac{2}{\varepsilon} \nabla \rho \cdot \nabla \lambda + \frac{1}{\varepsilon^2} \rho \|\nabla \lambda\|^2 - \frac{1}{\varepsilon} \rho \Delta \lambda \right) e^{-\lambda/\varepsilon}. \end{aligned}$$

Substituting these into the PDEs, we obtain system of backward and forward heat equations for φ and $\hat{\varphi}$:

$$\begin{aligned} \partial_t \varphi + \frac{\varepsilon}{2} \Delta \varphi &= 0, \\ \partial_t \hat{\varphi} - \frac{\varepsilon}{2} \Delta \hat{\varphi} &= 0. \end{aligned} \tag{6.27}$$

Considering the initial and final conditions, we have:

$$\begin{aligned} \rho(0, x) &= \hat{\varphi}(0, x) \varphi(0, x) = \rho_0(x), \\ \rho(1, y) &= \hat{\varphi}(1, y) \varphi(1, y) = \rho_1(y). \end{aligned}$$

Thus, the solution for (6.27) leads to the Schrödinger system:

$$\begin{aligned}\varphi(t, x) &= \int_{\mathbb{R}^n} p(t, x; 1, y) \varphi(1, y) dy, \\ \hat{\varphi}(t, y) &= \int_{\mathbb{R}^n} p(0, x; t, y) \hat{\varphi}(0, x) dx,\end{aligned}\tag{6.28}$$

which is equivalent to the Schrödinger system defined in Definition 3.3.1.

Thus, the optimal control $u^*(t, x)$ for the stochastic control problem is given by

$$u^*(t, x) = \varepsilon \nabla \ln \varphi(t, x) = \varepsilon \nabla \lambda(t, x).$$

□

6.7 Proof of Theorem 3.5.3

Proof. Starting from the inner entropically-regularized transport $\min_{M \in \Pi(a, b)} \frac{1}{2} \langle C, M \rangle + \gamma \langle M, \log M - 1 \rangle$, dualization yields $M^* = \text{diag}(u) K \text{diag}(v)$, with $u_i = e^{\lambda_i/\gamma}$, $v_j = e^{\mu_j/\gamma}$. By the envelope theorem, $\partial_b g(b) = \mu^*(b)$, and since $\mu_j = \gamma \log v_j$, the outer proximal step $\nabla_b \{g(b) + \tau \langle V + \epsilon \log b, b \rangle\} = 0$ gives

$$b_j = \exp\left(-1 - \frac{V_j}{\epsilon} - \frac{\mu_j^*}{\tau\epsilon}\right).$$

Imposing the column-sum constraint $b_j = \sum_i M_{ij}^* = v_j (K^T u)_j$ yields the power-law update

$$v_j = \left[\frac{\xi_j}{(K^T u)_j} \right]^{\frac{\epsilon\tau}{\epsilon\tau+\gamma}}.$$

Renaming $y = u$, $z = v$ gives the alternating recursion above, which converges (in the Thompson metric) to the unique fixed point (y^*, z^*) , and hence to the minimizer $\hat{\varphi}^k$. □

6.8 Derivation of problem 3.20

Applying Itô's lemma to $f(X_t)$ with $f \in C^2(\bar{\Omega})$ gives

$$df(X_t) = \left(u \cdot \nabla f + \frac{\varepsilon}{2} \Delta f \right)(t, X_t) dt + \sqrt{\varepsilon} \nabla f(t, X_t) \cdot dW_t + \nabla f(t, X_t) \cdot \mathbf{n}(X_t) d\gamma_t.$$

Integrating from 0 to t yields

$$f(X_t) - f(X_0) = \int_0^t \left(u \nabla f + \frac{\varepsilon}{2} \Delta f \right)(s, X_s) ds + \sqrt{\varepsilon} \int_0^t \nabla f(s, X_s) \cdot dW_s + \int_0^t \nabla f(s, X_s) \cdot \mathbf{n}(X_s) d\gamma_s.$$

Under P , the stochastic integral has zero mean, so

$$\mathbb{E}_P[f(X_t)] - \mathbb{E}_P[f(X_0)] = \int_0^t \mathbb{E}_P \left[u \cdot \nabla f + \frac{\varepsilon}{2} \Delta f \right](s) ds + \mathbb{E}_P \left[\int_0^t \nabla f(X_s) \cdot \mathbf{n}(X_s) d\gamma_s \right].$$

The last expectation corresponds to the boundary push induced by reflection. If f satisfies the Neumann condition $\nabla f \cdot \mathbf{n} = 0$ on $\partial\Omega$, the boundary term vanishes. Letting $\rho(t, \cdot)$ denote the law of X_t , we obtain

$$\frac{d}{dt} \int_{\Omega} f(x) \rho(t, x) dx = \int_{\Omega} \left(u \cdot \nabla f + \frac{\varepsilon}{2} \Delta f \right) \rho dx.$$

To rewrite the right-hand side in weak form, integrate by parts:

$$\int_{\Omega} u \cdot \nabla f \rho dx = - \int_{\Omega} f \nabla \cdot (u\rho) dx + \int_{\partial\Omega} f (u\rho \cdot \mathbf{n}) dS,$$

and

$$\int_{\Omega} \frac{\varepsilon}{2} \Delta f \rho dx = \int_{\Omega} f \frac{\varepsilon}{2} \Delta \rho dx - \int_{\partial\Omega} f \frac{\varepsilon}{2} \nabla \rho \cdot \mathbf{n} dS.$$

Combining these, the weak form contains boundary contributions of the type

$$\int_{\partial\Omega} f \left(u\rho \cdot \mathbf{n} - \frac{\varepsilon}{2} \nabla \rho \cdot \mathbf{n} \right) dS.$$

Since f is arbitrary, the only way for these boundary terms to vanish is to impose the *no-flux* boundary condition

$$\mathbf{n} \cdot \left(\frac{\varepsilon}{2} \nabla \rho - u\rho \right) = 0 \quad \text{on } \partial\Omega.$$

Therefore,

$$\frac{d}{dt} \int_{\Omega} f \rho dx = \int_{\Omega} f \left[-\nabla \cdot (u\rho) + \frac{\varepsilon}{2} \Delta \rho \right] dx,$$

and by arbitrariness of f , the density ρ satisfies the reflected Fokker–Planck equation

$$\partial_t \rho + \nabla \cdot (u\rho) - \frac{\varepsilon}{2} \Delta \rho = 0 \quad \text{in } \Omega, \quad \mathbf{n} \cdot \left(\frac{\varepsilon}{2} \nabla \rho - u\rho \right) = 0 \quad \text{on } \partial\Omega.$$

6.9 Proof of Theorem 3.6.1

Proof. We consider the Lagrangian functional

$$\mathcal{L}[\rho, u] = \int_0^1 \int_{\Omega} \frac{1}{2} \|u(t, x)\|^2 \rho(t, x) dx dt + \int_0^1 \int_{\Omega} \lambda(t, x) \left(\partial_t \rho + \nabla \cdot (u\rho) - \frac{\varepsilon}{2} \Delta \rho \right) dx dt.$$

By integration by parts in t and x , using the no-flux boundary condition

$$\mathbf{n} \cdot \left(\frac{\varepsilon}{2} \nabla \rho - u\rho \right) = 0 \quad \text{on } \partial\Omega,$$

we obtain

$$\int_0^1 \int_{\Omega} \lambda \left(\partial_t \rho + \nabla \cdot (u\rho) - \frac{\varepsilon}{2} \Delta \rho \right) dx dt = - \int_0^1 \int_{\Omega} \left(\partial_t \lambda + \nabla \lambda \cdot u + \frac{\varepsilon}{2} \Delta \lambda \right) \rho dx dt + [\lambda \rho]_{t=0}^{t=1}.$$

Stationarity with respect to u gives

$$u^{\text{opt}}(t, x) = \nabla \lambda(t, x).$$

Substituting into the Euler–Lagrange conditions, we arrive at the coupled PDE system

$$\begin{aligned}\partial_t \rho^{\text{opt}} + \nabla \cdot (\rho^{\text{opt}} u^{\text{opt}}) &= \frac{\varepsilon}{2} \Delta \rho^{\text{opt}}, \\ \partial_t \lambda + \frac{1}{2} \|\nabla \lambda\|^2 &= -\frac{\varepsilon}{2} \Delta \lambda,\end{aligned}$$

together with boundary conditions

$$\langle \nabla \lambda, \mathbf{n} \rangle = 0, \quad \mathbf{n} \cdot \left(\frac{\varepsilon}{2} \nabla \rho^{\text{opt}} - \rho^{\text{opt}} u^{\text{opt}} \right) = 0, \quad \text{on } \partial\Omega, \quad t \in [0, 1],$$

and $\rho^{\text{opt}}(0) = \rho_0$, $\rho^{\text{opt}}(1) = \rho_1$. This proves the theorem. \square

6.10 Proof of Proposition 3.6.2

Proof of Proposition. Introduce the Hopf–Cole transformation

$$\varphi(t, x) = e^{\lambda(t, x)/\varepsilon}, \quad \hat{\varphi}(t, x) = \rho(t, x) e^{-\lambda(t, x)/\varepsilon}.$$

Differentiating, we have

$$\partial_t \varphi = \frac{1}{\varepsilon} \varphi \partial_t \lambda, \quad \nabla \varphi = \frac{1}{\varepsilon} \varphi \nabla \lambda, \quad \Delta \varphi = \frac{1}{\varepsilon^2} \varphi \|\nabla \lambda\|^2 + \frac{1}{\varepsilon} \varphi \Delta \lambda,$$

and analogous relations for $\hat{\varphi}$. Substituting into the optimality system of the theorem shows that φ and $\hat{\varphi}$ satisfy

$$\partial_t \varphi = -\frac{\varepsilon}{2} \Delta \varphi, \quad \partial_t \hat{\varphi} = \frac{\varepsilon}{2} \Delta \hat{\varphi},$$

with reflecting boundary conditions

$$\langle \nabla \varphi, \mathbf{n} \rangle = 0, \quad \langle \nabla \hat{\varphi}, \mathbf{n} \rangle = 0, \quad x \in \partial\Omega, \quad t \in [0, 1].$$

Moreover, the boundary coupling is preserved:

$$\hat{\varphi}(x, 0) \varphi(x, 0) = \rho_0(x), \quad \hat{\varphi}(x, 1) \varphi(x, 1) = \rho_1(x).$$

Thus, $(\varphi, \hat{\varphi})$ provide the Schrödinger potentials, and the optimal control can be recovered as

$$u^{\text{opt}}(t, x) = \varepsilon \nabla \ln \varphi(t, x).$$

This proves the proposition. \square

6.11 Proximal recursions for the reflected SBP

In this appendix section, we provide the details for the proximal recursions for the reflected Schrödinger bridge problem (SBP). Since the method idea is to use particle simulation to approximate the PDE, we first need to simulate the reflected SDE.

6.11.1 Reflected SDE simulation

To simulate the reflected SDE, we can use the Euler–Maruyama method with a reflection step to ensure that the particles remain within the domain Ω . The simulation result is shown in Figure 6.1.

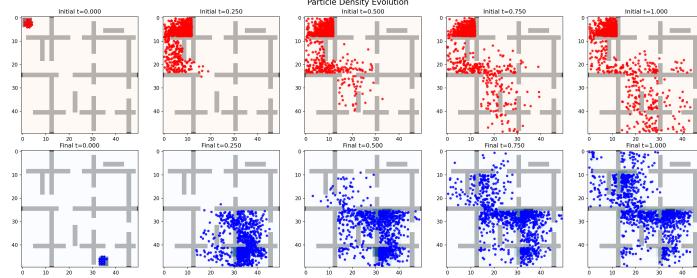


Figure 6.1: Reflected SDE simulation in our obstacle domain.

6.11.2 Proximal recursions for the reflected SBP

The proximal recursion for the reflected SBP can be formulated as follows:

$$\begin{aligned}\varphi_{k+1} &= \text{prox}_{\tau F}(\varphi_k) = \arg \min_{\varphi} \left\{ F(\varphi) + \frac{1}{2\tau} \|\varphi - \varphi_k\|^2 \right\}, \\ \hat{\varphi}_{k+1} &= \text{prox}_{\tau G}(\hat{\varphi}_k) = \arg \min_{\hat{\varphi}} \left\{ G(\hat{\varphi}) + \frac{1}{2\tau} \|\hat{\varphi} - \hat{\varphi}_k\|^2 \right\},\end{aligned}\quad (6.29)$$

where F and G are functionals that encode the boundary conditions and the dynamics of the reflected SBP, respectively. The proximal operators can be computed using iterative methods such as the Sinkhorn algorithm adapted for the reflected case.

6.11.3 Numerical explanation for bad performance of the proximal recursion

Since the the problem has complex boundary conditions, the interpolation procedure may not be accurate enough, leading to poor performance. We first give a comparison test result from finite element method and the proximal recursion. The numerical results are shown in Figure 6.2.

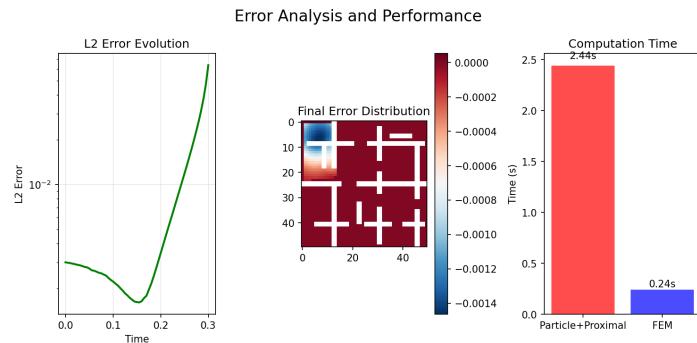


Figure 6.2: Numerical explanation for bad performance of the proximal recursion.

From this brief view of simulation procedure for heat equation with Neumann boundary condition, the proximal recursion method generally gives a poor approximation to solution from finite element method, with the error of PDE solution gradually increasing from time 0.2-0.3. Additionally, the interpolation procedure in the proximal recursion may further introduce errors. Since the gaussian distributed interpolation may not follows with obstacle reflecting boundary condition, this further leads to bad performance of the proximal recursion for the reflected SBP. During my numerical experiments, I tried with methods like neuroal network to approximate the solution of the heat equation with Neumann boundary condition, the gaussian interpolation method. They all give bad performance. Thus, I think the proximal recursion method may not be suitable for solving the reflected SBP with complex boundary conditions.

Bibliography

- 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. pages 2, 11
- Arne Beurling. An automorphism of product measures. *Annals of Mathematics*, 72(1):189–200, 1960. pages 3
- Susanne C Brenner and L Ridgway Scott. *The mathematical theory of finite element methods*. Springer, 2008. pages 37
- Peter J Bushell. Hilbert’s metric and positive contraction mappings in a banach space. *Archive for Rational Mechanics and Analysis*, 52(4):330–338, 1973. pages 4, 38
- Kenneth F Caluya and Abhishek Halder. Gradient flow algorithms for density propagation in stochastic systems. *IEEE Transactions on Automatic Control*, 65(10):3991–4004, 2019. pages 4, 29
- Kenneth F Caluya and Abhishek Halder. Reflected schrödinger bridge: Density control with path constraints. *2021 American Control Conference (ACC)*, pages 1137–1142, 2021a. pages 4, 35, 60
- Kenneth F Caluya and Abhishek Halder. Wasserstein proximal algorithms for the schrödinger bridge problem: Density control with nonlinear drift. *IEEE Transactions on Automatic Control*, 67(3):1163–1178, 2021b. pages 4, 28, 29
- José A Carrillo, Katy Craig, Li Wang, and Chaozhen Wei. Primal dual methods for wasserstein gradient flows. *Foundations of Computational Mathematics*, 22(2):389–443, 2022. pages 4
- Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011. pages 2
- Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Stochastic control liaisons: Richard sinkhorn meets gaspard monge on a schrodinger bridge. *Siam Review*, 63(2):249–313, 2021. pages 2
- Raphaël Chetrite, Paolo Muratore-Ginanneschi, and Kay Schwieger. E. schrödinger’s 1931 paper “on the reversal of the laws of nature”[“über die umkehrung der naturgesetze”, *sitzungsberichte der preussischen akademie der wissenschaften*,

- physikalisch-mathematische klasse, 8 n9 144–153]. *The European Physical Journal H*, 46:1–29, 2021. pages 2
- Julian D Cole. On a quasi-linear parabolic equation occurring in aerodynamics. *Quarterly of applied mathematics*, 9(3):225–236, 1951. pages 3
- Paolo Dai Pra. A stochastic control approach to reciprocal diffusion processes. *Applied mathematics and Optimization*, 23(1):313–329, 1991. pages 3
- Jim Douglas and Henry H Rachford Jr. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439, 1956. pages 2
- Robert Fortet. Résolution d'un système d'équations de m. schrödinger. *Journal de mathématiques pures et appliquées*, 19(1-4):83–105, 1940. pages 3
- Peter W Glynn and Rob J Wang. On the rate of convergence to equilibrium for reflected brownian motion. *Queueing Systems*, 89(1):165–197, 2018. pages 34
- Eldad Haber, Tauseef Rehman, and Allen Tannenbaum. An efficient numerical method for the solution of the l.2 optimal mass transfer problem. *SIAM Journal on Scientific Computing*, 32(1):197–211, 2010. pages 2
- Rolland L Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research*, 76(8):1905–1915, 1971. pages 32
- Francis H Harlow, J Eddie Welch, et al. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of fluids*, 8(12): 2182, 1965. pages 12
- J Michael Harrison. *Brownian models of performance and control*. Cambridge University Press, 2013. pages 34
- Eberhard Hopf. The partial differential equation. 1950. pages 3
- Benton Jamison. The markov processes of schrödinger. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 32(4):323–331, 1975. pages 3
- Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998. pages 4, 30
- Leonid V Kantorovich. On a problem of monge. In *CR (Doklady) Acad. Sci. URSS (NS)*, volume 3, pages 225–226, 1948. pages 1
- Vadim Linetsky. On the transition densities for reflected diffusions. *Advances in Applied Probability*, 37(2):435–460, 2005. pages 48
- Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979. pages 2

- Nicolas Papadakis, Gabriel Peyré, and Edouard Oudet. Optimal transport with proximal splitting. *SIAM Journal on Imaging Sciences*, 7(1):212–238, 2014. pages 2
- Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014. pages 2
- Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. pages 2, 9
- Andrey Pilipenko. *An introduction to stochastic differential equations with reflection*. Universitätsverlag Potsdam, 2014. pages 34
- Erwin Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pages 269–310, 1932. pages 2
- Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964. pages 3
- Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967. pages 3
- Anatoliy V Skorokhod. Stochastic equations for diffusion processes in a bounded region. *Theory of Probability & Its Applications*, 6(3):264–274, 1961. pages 34
- Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021. pages 1