Imperial College
London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF MATHEMATICS

Optimal Transport

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 problem, particularly focusing on the L^2-Wasserstein distance. We begin with a theoretical foundation, defining optimal transport maps and their properties. We then discuss various numerical algorithms for computing these maps, including dynamic energy minimization techniques, entropic regularization for Sinkhorn distances, and Schrödinger bridges and stochastic optimal transport methods. The dissertation also covers the implementation of these algorithms, highlighting their computational efficiency and convergence properties. We conclude with a discussion on the applications of optimal transport in various fields, such as machine learning, computer vision, and data analysis, demonstrating its versatility and importance in modern computational mathematics.

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. I also wish to thank my colleagues and friends at Imperial College London for their stimulating discussions and camaraderie. Finally, I am grateful to my family for their unwavering support and belief in my abilities.

Contents

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 mass-transport problem starts with *Monge* (1781), who sought the least-cost movement of soil. A modern optimisation perspective was introduced by Kantorovich (1948), yielding the Monge-Kantorovich formulation

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

where $\Pi(\mu,\nu)$ denotes all couplings between μ and ν and $c(x,y)$ is the transport cost.Choosing $c(x,y) = \|x-y\|^p$ gives the p-Wasserstein distance, a genuine metric on the space of probability measures (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:

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

where C is the 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:

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

The 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, Papadakis et al. (2014) proposed proximal splitting methods to solve the dynamic optimal transport problem, which allows for the 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:

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

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

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

Then using Hamiltonian-Jacobi-Bellman (HJB) equations, the problem can be transformed into forward-backward heat equations, providing a powerful framework for analyzing the dynamics of the system. The existence and uniqueness of solutions to these equations are proved in Beurling (1960), Fortet (1940), and Jamison (1975). Recent work extends SB to nonlinear drifts and high dimensions, i.e., the stochastic differential equation (SDE) is given by

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

thus reducing the problem into a system of forward-backward Kolmogorov partial differential equations. In particular, Caluya and Halder (2019) devise *Wasserstein-proximal-recursor* method that alternate a proximal step in the 2-Wasserstein metric with pointwise updates enforcing end-point marginals to solve the Kolmogorov partial differential equation; Caluya and Halder (2021) then employ this method for forward-backward Kolmogorov system, they also proved the scheme converges under a Hilbert-metric contraction and evolves weighted scatter-point clouds without spatial grids, enabling SB solvers for nonlinear stochastic systems and robotic-swarm density control.

1.2 Structures of the thesis

This dissertation is structured as follows:

- Chapter 1 provides an introduction to the problem of optimal transport and its relevance in various fields.
- Chapter 2 reviews the theoretical foundations of optimal transport, including key concepts and existing methods, mainly discussing the static and dynamic formulations of the optimal transport problem, each with its famous solution methods.
- Chapter 3 presents the proposed approach for solving optimal transport problems using stochastic methods, and build bridge between the static and dynamic formulations of the optimal transport problem, and discusses the Schrödinger bridge method and its connection to optimal transport.
- Chapter 4 discusses the implementation details and computational aspects of the proposed method.
- Chapter 5 concludes the dissertation and outlines potential directions for future research.

Chapter 2

Theory for solving Optimal Transport

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*

$$\rho_a(x) \;=\; \rho_b\big(T(x)\big)\,\big|\det\big(\partial T(x)\big)\big| \tag{2.1}$$

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

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

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:*

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

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 Solving the Optimal Transport Problem

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

2.1.1 Transform the problem

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

$$\rho(x,t) \;=\; \rho_0\big(T_t(x)\big)\,\big|\det\big(\partial T_t(x)\big)\big|, \qquad T_t \;=\; (1-t)\,\mathrm{Id}_d \;+\; t\,T,$$

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 gave proposition 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):

$$\begin{aligned} \min_{(v,\rho)\,\in\,\mathcal{C}^0} \quad & \frac{1}{2}\int_{[0,1]^d}\int_0^1 \rho(x,t)\,\big\|v(x,t)\big\|^2\,\mathrm{d}t\,\mathrm{d}x, \\ \mathcal{C}^0 = \Big\{ (v,\rho) \;\Big|\; & \partial_t\rho + \mathrm{div}_x\big(\rho\,v\big) = 0, \quad v(x,0) = v(x,1) = 0, \\ & \rho(x,0) = \rho_0(x), \quad \rho(x,1) = \rho_1(x)\Big\}. \end{aligned} \tag{2.4}$$

We first introduce change of variables to transform the non-convex problem (2.4). 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:

$$\min_{(m,\rho)\,\in\,\mathcal{C}} \; \mathcal{J}(m,\rho) \;=\; \int_{[0,1]^d}\int_0^1 J\big(m(x,t),\,\rho(x,t)\big)\,\mathrm{d}t\,\mathrm{d}x \tag{2.5}$$

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

$$J(m,\rho) = \begin{cases} \dfrac{\|m\|^2}{2\,\rho}, & \rho > 0, \\ 0, & (m,\rho) = (0,0), \\ +\infty, & \text{otherwise}, \end{cases} \tag{2.6}$$

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

$$\mathcal{C} = \left\{ (m, \rho) \;\middle|\; \partial_t \rho + \mathrm{div}_x(m) = 0,\ m(0,t) = m(1,t) = 0,\ \rho(x,0) = \rho_0(x),\ \rho(x,1) = \rho_1(x) \right\}. \tag{2.7}$$

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:

$$\mathcal{I}(m,\rho) = \begin{cases} 0, & \text{if } (m,\rho) \in \mathcal{C}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{2.8}$$

The overall optimization problem can now be expressed as:

$$\min_{(m,\rho)} \mathcal{J}(m,\rho) + \mathcal{I}(m,\rho) \tag{2.9}$$

2.1.2 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

$$\mathcal{G}_x = \left\{ (x_{i+\frac{1}{2}},\, t_j) \;\middle|\; x_{i+\frac{1}{2}} = (i+\tfrac{1}{2})\,\Delta x,\ t_j = j\,\Delta t,\ -1 \le i \le N,\ 0 \le j \le P \right\}, \tag{2.10}$$

$$\mathcal{G}_t = \left\{ (x_i,\, t_{j+\frac{1}{2}}) \;\middle|\; x_i = i\,\Delta x,\ t_{j+\frac{1}{2}} = (j+\tfrac{1}{2})\,\Delta t,\ 0 \le i \le N,\ -1 \le j \le P \right\}. \tag{2.11}$$

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:

$$\partial_t \rho(x_i, t_j) \approx \frac{\rho(x_i, t_{j+1/2}) - \rho(x_i, t_{j-1/2})}{\Delta t}, \tag{2.12}$$

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

$$\mathrm{div}_x(m)(x_i, t_j) \approx \frac{m(x_{i+1/2}, t_j) - m(x_{i-1/2}, t_j)}{\Delta x} \tag{2.13}$$

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

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

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

$$\mathcal{I} : \mathcal{E}_h \to \mathbb{R}^{\mathcal{G}_c} \times \mathbb{R}^{\mathcal{G}_c} \tag{2.14}$$

$$\forall\, 0 \le i \le N,\ 0 \le j \le P, \quad \begin{cases} m_{i,j} = \frac{1}{2}\big(\bar{m}_{i-1,j} + \bar{m}_{i,j}\big), \\ \rho_{i,j} = \frac{1}{2}\big(\bar{\rho}_{i,j-1} + \bar{\rho}_{i,j}\big). \end{cases} \tag{2.15}$$

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

$$\forall\, 0 \le i \le N,\ 0 \le j \le P, \quad (\mathrm{div}_h U)_{i,j} = \frac{\bar{m}_{i,j} - \bar{m}_{i-1,j}}{\Delta x} + \frac{\bar{\rho}_{i,j} - \bar{\rho}_{i,j-1}}{\Delta t}. \tag{2.16}$$

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

$$b(\mathcal{I}(U)) = \Big(\{m_{0,j}\}_{j=0}^{P},\ \{m_{N,j}\}_{j=0}^{P},\ \{\rho_{i,0}\}_{i=0}^{N},\ \{\rho_{i,P}\}_{i=0}^{N}\Big). \tag{2.17}$$

We then enforce the discrete boundary conditions

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

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:

$$\min_{U \in \mathcal{E}_h}\ \mathcal{J}_h(U) + \mathcal{I}_h(U) \tag{2.18}$$

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

$$\mathcal{J}_h(U) = \sum_{(x_i,t_j)\in\mathcal{G}_c} J\big(\mathcal{I}(U)(x_i,t_j)\big)\,\Delta x\,\Delta t \tag{2.19}$$

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

$$\mathcal{I}_h(U) = \begin{cases} 0, & \text{if } U \in \mathcal{C}_h, \\ +\infty, & \text{otherwise}, \end{cases} \tag{2.20}$$

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

$$\mathcal{C}_h = \big\{\, U \in \mathcal{E}_h \ \big|\ \mathrm{div}_h(U) = 0,\ b(\mathcal{I}(U)) = b_0 \big\}. \tag{2.21}$$

Algorithm 1 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:
$$W^{(k+1/2)} = W^{(0)} + w_1 \operatorname{prox}_{\tau J}(W^{(k)}) + w_2 \operatorname{prox}_{\tau \mathcal{I}_h}(W^{(k)}), \tag{2.22}$$
4:
$$W^{(k+1)} = W^{(0)} + \mu(2W^{(k+1/2)} - W^{(k)} - \operatorname{prox}_{\tau J}(W^{(k)}) - \operatorname{prox}_{\tau \mathcal{I}_h}(W^{(k)})) \tag{2.23}$$
5: Update the variables:
$$U^{(k+1)} = U^{(k)} + \mu(W^{(k+1/2)} - U^{(k)}) \tag{2.24}$$
6: **end for**
7: **Output:** Final solution $U^{(K)} = (m^{(K)}, \rho^{(K)})$

2.1.3 Solving the Discrete Problem with Symmmetric 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:
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 2.1.1. *The proximal operator of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as*

$$\operatorname{prox}_{\tau f}(x) = \arg\min_{y \in \mathbb{R}^n} \left(\tau f(y) + \frac{1}{2}\|y - x\|^2 \right) \tag{2.25}$$

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

Proposition 2.1.2. *The proximal operator for the cost functional J is given by*

$$\operatorname{prox}_{\tau J}(m, \rho) = \begin{cases} \frac{\bar{\rho} m}{\bar{\rho} + 2\tau}, & \text{if } \bar{\rho} > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{2.26}$$

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

$$P(X) = (X - \bar{\rho})(X + 2\tau)^2 - \tau\|m\|^2 = 0. \tag{2.27}$$

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

$$\begin{aligned} \frac{\partial h}{\partial m'} &= \tau \frac{2m'}{\rho'} + (m' - m) = 0, \\ \frac{\partial h}{\partial \rho'} &= -\frac{\tau \|m'\|^2}{\rho'^2} + (\rho' - \rho) = 0. \end{aligned} \tag{2.28}$$

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

$$m' = \frac{\rho' m}{\rho' + 2\tau}. \tag{2.29}$$

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

$$\begin{aligned} & \frac{\tau \|m\|^2}{(\rho' + 2\tau)^2} - (\rho' - \rho) = 0, \\ \Rightarrow \quad & P(\rho') = (\rho' - \rho)(\rho' + 2\tau)^2 - \tau \|m\|^2 = 0. \end{aligned} \tag{2.30}$$

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

$$\operatorname{prox}_{\tau J}(m, \rho) = \begin{cases} \frac{\bar{\rho} m}{\bar{\rho} + 2\tau}, & \text{if } \bar{\rho} > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{2.31}$$

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

□

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 $\mathcal{I}_c$ satisfying divergence form is defined as follows:

Definition 2.1.3. *The indicator function $\mathcal{I}_s$ is defined as*

$$\mathcal{I}_s(U) = \begin{cases} 0, & \textit{if } U \in \mathcal{C}, \\ \infty, & \textit{otherwise}, \end{cases} \tag{2.32}$$

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 2.1.4. *The indicator function $\mathcal{I}_c$ is defined as*

$$\mathcal{I}_c(U) = \begin{cases} 0, & \textit{if } b(\mathcal{I}(U)) = b_0, \\ \infty, & \textit{otherwise}, \end{cases} \tag{2.33}$$

where b_0 is the prescribed boundary condition defined in (2.17).

Proposition 2.1.5. *The proximal operator for the indicator function $\mathcal{I}_s$ is given by*

$$\mathrm{prox}_{\tau\mathcal{I}_s}(U) = \left(\mathrm{Id} - A^*\Delta^{-1}A\right)U \tag{2.34}$$

where A is the divergence operator defined in (2.16) *and Δ is the discrete Laplacian operator.*

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:

$$h(U', \lambda) = \frac{1}{2}\|U' - U\|^2 + \langle\lambda, \mathrm{div}(U')\rangle, \tag{2.35}$$

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

$$\begin{aligned} \frac{\partial h}{\partial U'} &= (U' - U) + A^*\lambda = 0, \\ \frac{\partial h}{\partial \lambda} &= AU' = 0. \end{aligned} \tag{2.36}$$

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

$$U' = U - A^*\lambda. \tag{2.37}$$

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

$$A(U - A^*\lambda) = 0. \tag{2.38}$$

Rearranging this gives us:

$$\lambda = \Delta^{-1}AU, \tag{2.39}$$

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

$$U' = U - A^*\Delta^{-1}AU. \tag{2.40}$$

□

Remark 2.1.6. *We give a detailed explanation why $\Delta = AA^*$ is the discrete Laplacian operator and how $\mathcal{I}_s$ is calculated numerically. We first consider the continuous case of the Lagrange multiplier.*

$$\mathcal{L}[u', \lambda] = \frac{1}{2}\int_\Omega |u' - u|^2 + \int_\Omega \lambda \,\mathrm{div}\, u' dx \tag{2.41}$$

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

$$\delta_{u'}\mathcal{L} = \int_\Omega (u' - u)\cdot\delta u'\, dx + \int_\Omega \lambda\,\mathrm{div}(\delta u')\, dx. \tag{2.42}$$

Integrating the second term by parts gives

$$\int_\Omega \lambda \operatorname{div}(\delta u')\,dx = -\int_\Omega \nabla\lambda\cdot\delta u'\,dx + \int_{\partial\Omega} \lambda\,(\delta u'\cdot n)\,ds. \tag{2.43}$$

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

$$\int_\Omega (u' - u - \nabla\lambda)\cdot\delta u'\,dx = 0 \tag{2.44}$$

for all admissible $\delta u'$ yields

$$u' - u - \nabla\lambda = 0 \quad\Longrightarrow\quad u' = u + \nabla\lambda. \tag{2.45}$$

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

$$\Delta\lambda = -\operatorname{div} u, \tag{2.46}$$

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

$$u' = u + \nabla\lambda, \tag{2.47}$$

which in operator form reads

$$u' = \Big(I - \nabla\,\Delta^{-1}\operatorname{div}\Big)\,u. \tag{2.48}$$

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

$$\Delta\lambda = -\operatorname{div} u \tag{2.49}$$

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

$$u' = u + \nabla\lambda. \tag{2.50}$$

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

Proposition 2.1.7. *The proximal operator for the indicator function $\mathcal{I}_c$ is given by*

$$\operatorname{prox}_{\tau\mathcal{I}_c}(U) = U - \mathcal{I}^* b^* (b\mathcal{I}\mathcal{I}^* b^*)^{-1}(b(\mathcal{I}(U)) - b_0), \tag{2.51}$$

where b^* is the adjoint operator of the boundary operator b defined in (2.17).

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:

$$h(U', \lambda) = \frac{1}{2}\|U' - U\|^2 + \langle\lambda, b(\mathcal{I}(U')) - b_0\rangle, \tag{2.52}$$

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

$$\begin{aligned} \frac{\partial h}{\partial U'} &= (U' - U) + \mathcal{I}^* b^* \lambda = 0, \\ \frac{\partial h}{\partial \lambda} &= b(\mathcal{I}(U')) - b_0 = 0. \end{aligned} \tag{2.53}$$

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

$$U' = U - \mathcal{I}^* b^* \lambda. \tag{2.54}$$

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

$$b(\mathcal{I}(U - \mathcal{I}^* b^* \lambda)) = b_0. \tag{2.55}$$

Rearranging this gives us:

$$\lambda = (b\mathcal{I}\mathcal{I}^* b^*)^{-1}(b(\mathcal{I}(U)) - b_0). \tag{2.56}$$

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

$$U' = U - \mathcal{I}^* b^* (b\mathcal{I}\mathcal{I}^* b^*)^{-1}(b(\mathcal{I}(U)) - b_0). \tag{2.57}$$

□

Remark 2.1.8. *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*

$$A\begin{pmatrix} \bar{m} \\ m \end{pmatrix} = \begin{pmatrix} \mathcal{I} & -I_n \\ 0 & e_1^T \\ 0 & e_n^T \end{pmatrix} \begin{pmatrix} \bar{m} \\ m \end{pmatrix} = \begin{pmatrix} 0_n \\ 0 \\ 0 \end{pmatrix} \tag{2.58}$$

And for density boundary condition, we set

$$A\begin{pmatrix} \bar{\rho} \\ \rho \end{pmatrix} = \begin{pmatrix} \mathcal{I} & -I_n \\ 0 & e_1^T \\ 0 & e_n^T \end{pmatrix} \begin{pmatrix} \bar{\rho} \\ \rho \end{pmatrix} = \begin{pmatrix} 0_n \\ \rho^0 \\ \rho^1 \end{pmatrix} \tag{2.59}$$

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:

$$A\begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} \mathcal{I} & -I_n \\ 0 & e_1^T \\ 0 & e_n^T \end{pmatrix} \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} 0_n \\ a \\ b \end{pmatrix} = g, \tag{2.60}$$

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 $Ax = g$:

$$\mathrm{prox}_{\tau \mathcal{I}_c}(U) = (\mathrm{Id} - A^*(AA^*)^{-1}A)U + A^*(AA^*)^{-1}g, \tag{2.61}$$

2.1.4 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

$$\min_{U\in\mathcal{E}_h} \mathcal{J}(\mathcal{I}(U)) + \mathcal{I}_h(U) \tag{2.62}$$

We assign $G_2 = \mathcal{J}, A = \mathcal{I}$ and $G_1 = \iota_{\mathcal{C}}$. Thus, the iterations compute a sequence $\left(U^{(\ell)}, \Upsilon^{(\ell)}, V^{(\ell)}\right) \in \mathcal{E}_{\mathrm{s}} \times \mathcal{E}_{\mathrm{s}} \times \mathcal{E}_{\mathrm{c}}$ of variables from an initial $\left(\Upsilon^{(0)}, V^{(0)}\right)$ with the following steps:

Algorithm 2 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^*:

$$V^{(\ell+1)} = \mathrm{Prox}_{\sigma G_2^*}\left(V^{(\ell)} + \sigma\mathcal{I}\Upsilon^{(\ell)}\right) \tag{2.63}$$

4: Compute the proximal operator for G_1:

$$U^{(\ell+1)} = \mathrm{Prox}_{\tau G_1}\left(U^{(\ell)} - \tau\mathcal{I}^* V^{(\ell+1)}\right) \tag{2.64}$$

5: Update the variables:

$$\Upsilon^{(\ell+1)} = U^{(\ell+1)} + \theta\left(U^{(\ell+1)} - U^{(\ell)}\right) \tag{2.65}$$

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

$$\forall w \in \mathcal{H}, \quad \mathrm{Prox}_{\gamma F^\star}(w) = w - \gamma\,\mathrm{Prox}_{F/\gamma}(w/\gamma), \tag{2.66}$$

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

Remark 2.1.10. *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 $\mathrm{Prox}_{\frac{1}{\tau}J}\left(\frac{V^{(\ell)} - \tau\mathcal{I}^* V^{(\ell+1)} + b_0/2}{\tau}\right) - b_0/2$ 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:*

$$\mathrm{Prox}_{\tau G_1}(U) = U - \nabla\Delta^{-1} divU \tag{2.67}$$

2.1.5 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:

$$\begin{aligned} &\min_{(v,\rho)\in\mathcal{C}^0} \quad \frac{1}{2}\int_{[0,1]^d}\int_0^1 \rho^\beta(x,t)\left\|v(x,t)\right\|^2 \mathrm{d}t\,\mathrm{d}x, \\ \mathcal{C}^0 = \Big\{ (v,\rho) \;\Big|\; &\partial_t\rho + \mathrm{div}_x\big(\rho^\beta\, v\big) = 0, \quad v(x,0) = v(x,1) = 0, \\ &\rho(x,0) = \rho_0(x), \quad \rho(x,1) = \rho_1(x)\Big\}. \end{aligned} \tag{2.68}$$

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:

$$\mathcal{J}_\beta^w(\mathcal{I}(U)) = \sum_{k\in\mathcal{G}_\mathrm{c}} w_k J_\beta\left(m_k, f_k\right) \tag{2.69}$$

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

$$J_\beta(m,\rho) = \begin{cases} \frac{\|m\|^2}{2\rho^\beta} & \text{if } f > 0 \\ 0 & \text{if } (m,\rho) = (0,0) \\ +\infty & \text{otherwise} \end{cases} \tag{2.70}$$

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 2.1.11. *(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:*

$$\mathrm{prox}_{\tau J_\beta}(m,\rho) = \begin{cases} \frac{\bar\rho^\beta m}{\bar\rho^\beta + 2\tau}, & \textit{if } \bar\rho > 0, \\ 0, & \textit{otherwise}, \end{cases} \tag{2.71}$$

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

$$P(X) = X^{1-\beta}(X-\bar\rho)(X^\beta + 2\tau)^2 - \tau\beta\|m\|^2 = 0. \tag{2.72}$$

2.1.6 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: $\mathrm{prox}_{\tau J_\beta}(m,\rho)$.
- Proximal operator for divergence form: $\mathrm{prox}_{\tau\mathcal{I}_s}(U)$.

- 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

$$P(X) = X^{1-\beta}(X - \bar{\rho})(X^\beta + 2\tau)^2 - \tau\beta\|m\|^2 = 0. \tag{2.73}$$

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

  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): $\mathcal{O}(N^3P)$,
  - Vertical direction (y): $\mathcal{O}(N^3P)$,

– Temporal direction (t): $\mathcal{O}(P^2N^2)$.

Summing over all three directions, the total complexity is

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

Factoring this, we obtain

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

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:

$$\mathcal{O}(T(Kn + n\log n + n^{4/3})), \tag{2.74}$$

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 Kantorovich's optimal transport problem

We now consider the discrete setting for Kantorovich's optimal transport problem.

Definition 2.2.1. *(Kantorovich's optimal transport problem) The Kantorovich's optimal transport problem is defined as the following optimization problem:*

$$\begin{aligned} \min_{\mathbf{P}\in\mathbb{R}_+^{n\times m}} & \langle \mathbf{C}, \mathbf{P}\rangle \\ \textit{s.t.} \quad & \mathbf{P}\mathbf{1}_m = \mathbf{a} \\ & \mathbf{P}^\top \mathbf{1}_n = \mathbf{b} \end{aligned} \tag{2.75}$$

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.

This problem 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.

Definition 2.2.2. *(Sinkhorn distance) The Sinkhorn distance is defined as the following optimization problem:*

$$\begin{aligned} \min_{\mathbf{P}\in\mathbb{R}_+^{n\times m}} & \langle \mathbf{C}, \mathbf{P}\rangle + \epsilon H(\mathbf{P}) \\ \textit{s.t.} \quad & \mathbf{P}\mathbf{1}_m = \mathbf{a} \\ & \mathbf{P}^\top \mathbf{1}_n = \mathbf{b} \end{aligned} \tag{2.76}$$

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

Proposition 2.2.3. *(Convergence with ε). The unique solution $\mathbf{P}_\varepsilon$ converges to the optimal transport plan as $\varepsilon \to 0$. Additionally, if $\varepsilon \to \infty$, the solution $\mathbf{P}_\varepsilon$ converges to $ab^T = (a_i b_j)_{i,j}$.*

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:

$$\mathcal{L}(\mathbf{P}, \mathbf{u}, \mathbf{v}) = \langle \mathbf{C}, \mathbf{P}\rangle + \epsilon H(\mathbf{P}) - \langle \mathbf{u}, \mathbf{P}\mathbf{1}_m - \mathbf{a}\rangle - \langle \mathbf{v}, \mathbf{P}^\top \mathbf{1}_n - \mathbf{b}\rangle \tag{2.77}$$

Consider the first order optimality conditions, we have:

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

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

$$\mathbf{P}_{i,j} = \mathbf{u}_i \mathbf{v}_j \mathbf{K}_{i,j} \tag{2.78}$$

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:

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

$$\Rightarrow \mathbf{u} = \frac{\mathbf{a}}{\mathbf{K}\mathbf{v}}$$

Similarly, for the second marginal constraint, we have:

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

$$\Rightarrow \mathbf{v} = \frac{\mathbf{b}}{\mathbf{K}^\top \mathbf{u}}$$

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

Algorithm 3 Sinkhorn Algorithm

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

2.2.1 Convergence 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 $K = \exp(-C/\varepsilon)$ 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.

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 $\Omega = C\big([0,1];\mathbb{R}^n\big)$ 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*

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

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*

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

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*

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

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:

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

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):

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

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

$$\iint \left[\log \rho_{01}^W(x,y)\right] \rho_{01}(x,y)dxdy = \iint \left[\log p^W(0,x;1,y)\right] \rho_{01}(x,y)dxdy + \int \left[\log \rho_0(x)\right] \rho_0(x)dx.$$

Thus, the Schrödinger bridge problem is equivalent to:

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

This is also equivalent to the problem:

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

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

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

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

$$\mathcal{L}[\rho_{01},\lambda,\mu] = \iint \Big[\tfrac{1}{2}\|x-y\|^2 + \varepsilon \ln \rho_{01}\Big]\rho_{01}\,dx\,dy + \int \lambda(x)\Big(\int \rho_{01}\,dy - \rho_0\Big)dx + \int \mu(y)\Big(\int \rho_{01}\,dx - \rho_1\Big)dy.$$

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

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

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

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

Solving for ρ_{01}^* yields

$$\rho_{01}^*(x,y) = \exp\Big(-1 - \tfrac{1}{\varepsilon}\tfrac{\|x-y\|^2}{2} - \tfrac{\lambda(x)+\mu(y)}{\varepsilon}\Big), \quad (3.4)$$

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:

$$\mathcal{L}[\rho_{01},\lambda,\mu] = \iint \Big[\log \tfrac{\rho_{01}^P(x,y)}{\rho_{01}^W(x,y)}\Big]\,\rho_{01}^P(x,y)\,dx\,dy + \int \lambda(x)\Big(\int \rho_{01}\,dy - \rho_0\Big)dx + \int \mu(y)\Big(\int \rho_{01}\,dx - \rho_1\Big)dy.$$

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

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

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

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

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,

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

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:*

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

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 $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{0\le t\le T}, Q)$ 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*

$$\mathbb{E}_Q\Big[\exp\Big(\tfrac{1}{2}\int_0^T \|\theta_s\|^2\,ds\Big)\Big] < \infty.$$

Define the exponential martingale

$$Z_t \;=\; \exp\Big(-\int_0^t \theta_s\cdot dW_s \;-\; \tfrac{1}{2}\int_0^t \|\theta_s\|^2\,ds\Big), \quad 0\le t\le T.$$

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:

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

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,

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

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

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

so that overall

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

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:

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

Thus, the expectation of the relative entropy becomes

$$\mathbb{D}(P\|W) = \mathbb{E}_P\Big[\log \frac{dP}{dW}\Big] = \mathbb{E}_P\Big[\log \frac{\rho_0^P(X_0)}{\rho_0^W(X_0)}\Big] + \mathbb{E}_P\Big[\tfrac{1}{2} \int_0^1 \|\beta_t\|^2 \, dt\Big] = \mathbb{D}(\rho_0\|\rho_0^W) + \mathbb{E}_P\Big[\tfrac{1}{2} \int_0^1 \|\beta_t\|^2 \, dt\Big].$$

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:

$$\begin{aligned} &\min_{u\in\mathcal{U}} \ \mathbb{E}\Big[\int_0^1 \tfrac{1}{2\varepsilon} \|u_t\|^2 \, dt\Big], \\ &\text{subject to} \quad dX_t = u_t \, dt + \sqrt{\varepsilon} \, dW_t, \\ &\qquad\qquad X_0 \sim \rho_0(x), \quad X_1 \sim \rho_1(x), \end{aligned} \tag{3.5}$$

where $\mathcal{U}$ is the set of adapted controls with finite energy, let $\rho(t,x)$ be the probability density of X_t. Fix a test function $f \in C_c^\infty(\mathbb{R}^n)$. Applying Itô's lemma to $f(X_t)$ gives

$$df\big(X_t\big) = \Big(u\cdot \nabla f + \tfrac{\varepsilon}{2}\Delta f\Big)(t, X_t) \, dt + \sqrt{\varepsilon} \, \nabla f(t, X_t) \cdot dW_t.$$

Integrate from 0 to t:

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

Under P, the stochastic integral has zero mean, so

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

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

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

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

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

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

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

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

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

is given by

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

Proof. Consider the unconstrained minimization of the lagrangian functional

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

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

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

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:

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

Consider the Hopf-Cole transformation

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

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 $\hat{\varphi}$:

$$
\begin{aligned}
\partial_t\varphi + \tfrac{\varepsilon}{2}\Delta\varphi &= 0,\\
\partial_t\hat{\varphi} - \tfrac{\varepsilon}{2}\Delta\hat{\varphi} &= 0.
\end{aligned}
\tag{3.6}
$$

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 (3.6) leads to the Schrödinger system:

$$
\begin{aligned}
\varphi(t,x) &= \int_{R^n} p(t,x;1,y)\,\varphi(1,y)\,dy,\\
\hat{\varphi}(t,y) &= \int_{R^n} p(0,x;t,y)\,\hat{\varphi}(0,x)\,dx,
\end{aligned}
\tag{3.7}
$$

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

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

□

This proof 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

$$\begin{aligned} &\min_{u(\mathbf{x},t)} \ \mathbb{E}\Big[\int_0^1 \frac{1}{2}\,\|u(x,t)\|_2^2\,dt\Big], \\ \text{subject to} \quad &dX(t) = f\big(X(t),t\big)\,dt + B(t)\,u_t\,dt + \sqrt{\epsilon}\,B(t)\,dw_t, \\ &X(0) \sim \rho_0, \quad X(1) \sim \rho_1, \end{aligned} \tag{3.8}$$

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 $t \in [0,1]$,

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

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

$$x^{\mathsf{T}}\,D(t)\,x \ \ge\ c_3\,\|x\|_2^2.$$

Under these conditions, the uncontrolled SDE

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

has a strictly positive, continuous transition density, thus our stochastic control problem (3.8) is well-posed.
We do simliar approach as in the previous section, apply Itô's lemma to

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

and integrate by parts, we transform the problem (3.8) into a fluid dynamics version:

$$\begin{aligned} &\min_{\rho,u} \ \frac{1}{2}\int_0^1 \int_{\mathbb{R}^n} \|u(x,t)\|_2^2\,\rho(x,t)\,dx\,dt, \\ \text{subject to} \quad &\frac{\partial \rho}{\partial t} + \nabla\cdot\big[\rho\,\big(f + B\,u\big)\big] = \frac{\epsilon}{2}\,\langle D,\,\Delta\rho\rangle, \\ &\rho(x,0) = \rho_0(x), \quad \rho(x,1) = \rho_1(x). \end{aligned} \tag{3.9}$$

where D is the diffusion tensor defined as $D(t) = B(t)\,B(t)^{\mathsf{T}}$. 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

$$\begin{aligned} &\min_{\rho,m}\ \frac{1}{2}\int_0^1\int_{\mathbb{R}^n} J\big(\rho,m\big)\,dx\,dt,\\ \text{subject to}\quad &\frac{\partial\rho}{\partial t}+\nabla\cdot\big[\rho\,f+B\,m\big]-\frac{\epsilon}{2}\,\langle D,\,\nabla^2\rho\rangle=0,\\ &\rho(x,0)=\rho_0(x),\quad \rho(x,1)=\rho_1(x),\quad m(x,0)=0,\quad m(x,1)=0, \end{aligned} \tag{3.10}$$

where

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

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

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

are linear in (ρ,m). The problem (3.10) 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* $\big(\rho^*(x,t),\,u^*(x,t)\big)$ *solving* (3.10) *must satisfy*

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

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*

$$\begin{aligned} \partial_t \varphi &= -\nabla\varphi \cdot f \; - \; \frac{\epsilon}{2}\langle D,\, \Delta\varphi\rangle, \\ \partial_t \hat{\varphi} &= -\nabla\hat{\varphi} \cdot f \; + \; \frac{\epsilon}{2}\langle D,\, \Delta\hat{\varphi}\rangle. \end{aligned} \tag{3.11}$$

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 (2021) made a detailed proof of this proposition by integrating by parts and using the properties of the Hopf-Cole transformation.

3.6 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

$$\begin{aligned} \min_{\rho,u} \; & \frac{1}{2}\int_0^1 \int_{\mathbb{R}^n} \|u(x,t)\|_2^2 \,\rho(x,t)\,dx\,dt, \\ \text{s.t.} \quad & \partial_t\rho + \nabla\cdot\big[\rho\,(u - \nabla V)\big] = \frac{\epsilon}{2}\Delta\rho, \\ & \rho(x,0) = \rho_0(x), \quad \rho(x,1) = \rho_1(x). \end{aligned} \tag{3.12}$$

By proposition 3.5.2, the problem (3.12) is equivalent to the following system of backward and forward Kolmogorov equations:

$$\begin{aligned} \partial_t \varphi &= -\nabla\varphi\cdot\nabla V \; - \; \frac{\epsilon}{2}\Delta\varphi, \\ \partial_t \hat{\varphi} &= \nabla\hat{\varphi}\cdot\nabla V \; + \; \frac{\epsilon}{2}\Delta\hat{\varphi}, \end{aligned} \tag{3.13}$$

with boundary conditions

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

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

$$\begin{aligned} \partial_t\hat{\varphi} &= \nabla\cdot(\hat{\varphi}\,\nabla V) + \frac{\epsilon}{2}\Delta\hat{\varphi}, \quad \hat{\varphi}(x,0) = \hat{\varphi}_0(x), \\ \partial_s p &= \nabla\cdot(p\,\nabla V) + \frac{\epsilon}{2}\Delta p, \quad p(x,0) = \varphi_1(x)\,e^{-V(x)/\epsilon}, \end{aligned} \tag{3.14}$$

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

3.7 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

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

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

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

Thus, in each steps we update the positions of the particles and use proximal operator to update the evolution of forward Kolmogorov equation system (3.14).

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

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

3.7.1 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

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

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

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

where

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

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

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

can be written as

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

Thus, we obtain the optimal solution M^* as

$$M^*_{ij} = \exp\left(\boldsymbol{\lambda}(i)/\gamma\right)\exp\left(-\boldsymbol{C}_k(i,j)/(2\gamma)\right)\exp\left(\boldsymbol{\mu}(j)/\gamma\right), \tag{3.15}$$

Theorem 3.7.1 (Proximal–Sinkhorn Recursion)**.** *Let*

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

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

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

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

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

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

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

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 $M^* = \operatorname{diag}(u)\,K\,\operatorname{diag}(v)$, 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

$$b_j = \exp\Big(-1 - \tfrac{V_j}{\epsilon} - \tfrac{\mu_j^*}{\tau\epsilon}\Big).$$

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

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

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 $\hat\varphi^k$. □

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

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

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 $V : \mathbb{R}^d \to \mathbb{R}$, parameters $\gamma, \tau, \epsilon > 0$.
Ensure: Updated weights $\hat{\varphi}^k \in \mathbb{R}_+^N$.
Compute cost matrix

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

Compute potential-weight vector $\xi_j \leftarrow \exp\big(-1 - V(x_j^{k-1})/\epsilon\big)$
Initialize $z^{(0)} \in \mathbb{R}_+^N$ (e.g. $z^{(0)} = \mathbf{1}$)
for $t = 0, 1, 2, \ldots$ until convergence **do**
    Row-scaling:

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

    Column-scaling (proximal step):

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

end for
$\hat{\varphi}^k \leftarrow z^{(*)} \odot \big(K^T\, y^{(*)}\big)$
return $\hat{\varphi}^k$

Similar procedure can be applied to compute p^k for the backward Kolmogorov equation system (3.14).

3.7.2 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 $\big(\hat{\varphi}_0, \varphi_1\big)$. Once these are known, the transient factors $\big(\hat{\varphi}(x,t), \varphi(x,t)\big)$ 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. $\varphi_1(x) \;=\; \dfrac{\rho_1(x)}{\hat{\varphi}_1(x)}$.
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. $\hat{\varphi}_0(x) = \dfrac{\rho_0(x)}{\varphi_0(x)}$.

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 $\big(p(s=0), \hat{\varphi}_0\big)$ pairs. If both are below the prescribed tolerance, stop; otherwise return to step 2.

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

Remark 3.7.2. *In the above algorithm, when doing proximal recursion steps, we use interpolation 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 Recursor for Gradient Flow Schrödinger Bridge

initialize $\hat{\varphi}_1 \leftarrow$ random positive vector in R^N
$\varphi_0 \leftarrow 0,\ \hat{\varphi}_0 \leftarrow 0,\ \varphi_1 \leftarrow 0,\ p_0 \leftarrow 0,\ p_1 \leftarrow 0$
allocate $p^{\text{temp}},\ \hat{\varphi}^{\text{temp}} \in R^{N\times(\text{numSteps}+1)}$
$p_0^{\text{old}} \leftarrow p_0,\ \hat{\varphi}_0^{\text{old}} \leftarrow \hat{\varphi}_0$
for $\ell = 1, \ldots, \text{maxIter}_{\text{SB}}$ **do**
    $\varphi_1 \leftarrow \rho_1 \oslash \hat{\varphi}_1$
    $p_0 \leftarrow \varphi_1 \odot \exp\big(-V(x)/\epsilon\big)$
    $p^{\text{temp}}(:, 1) \leftarrow p_0$
    for $i = 1, \ldots, \text{numSteps}$ **do**
        Use Euler–Maruyama to update particle positions
        $p^{\text{temp}}(:, i+1) \leftarrow$ Algorithm 4
    end for
    Interpolate $p^{\text{temp}}(:, \text{numSteps}+1)$ onto initial density simulation points.
    $p_1 \leftarrow p^{\text{temp}}(:, \text{numSteps}+1)$
    $\varphi_0 \leftarrow p_1 \odot \exp\big(V(x)/\epsilon\big)$
    $\hat{\varphi}_0 \leftarrow \rho_0 \oslash \varphi_0$
    $\hat{\varphi}^{\text{temp}}(:, 1) \leftarrow \hat{\varphi}_0$
    for $j = 1, \ldots, \text{numSteps}$ **do**
        Use Euler–Maruyama to update particle positions
        $\hat{\varphi}^{\text{temp}}(:, j+1) \leftarrow$ Algorithm 4
    end for
    Interpolate $\hat{\varphi}^{\text{temp}}(:, \text{numSteps}+1)$ onto final density simulation points.
    $\hat{\varphi}_1 \leftarrow \hat{\varphi}^{\text{temp}}(:, \text{numSteps}+1)$
    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**
        break
    end if
    $p_0^{\text{old}} \leftarrow p_0,\ \hat{\varphi}_0^{\text{old}} \leftarrow \hat{\varphi}_0$
end for ▷ Endpoint factors: $\hat{\varphi}_0, \varphi_1$
initialize $\hat{\varphi}_1^{\text{trans}} \leftarrow \hat{\varphi}_0$
initialize $p_1^{\text{trans}} \leftarrow \varphi_1 \odot \exp\big(-V(x)/\epsilon\big)$
for $k = 1, \ldots, \text{numSteps}$ **do**
    Use Euler–Maruyama to update particle positions
    $\hat{\varphi}_{k+1}^{\text{trans}} \leftarrow$ Algorithm 4
    interpolate $\hat{\varphi}_{k+1}^{\text{trans}}$ onto global grid points.
    $p_{k+1}^{\text{trans}} \leftarrow$ Algorithm 4
    $\varphi_{k+1}^{\text{trans}} \leftarrow p_{\text{numSteps}+2-k}^{\text{trans}} \odot \exp\big(V(x)/\epsilon\big)$
    interpolate $\varphi_{k+1}^{\text{trans}}$ onto global grid points.
end for
return $\{\hat{\varphi}_k^{\text{trans}}, \varphi_k^{\text{trans}}\}_{k=1}^{\text{numSteps}}$
Calculate density $\{\rho_k^{\text{trans}}\}_{k=1}^{\text{numSteps}} = \{\hat{\varphi}_k^{\text{trans}}\varphi_k^{\text{trans}}\}_{k=1}^{\text{numSteps}}$

Chapter 4

Experimental Results

4.1 Numerical example with 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:

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

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,

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

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:

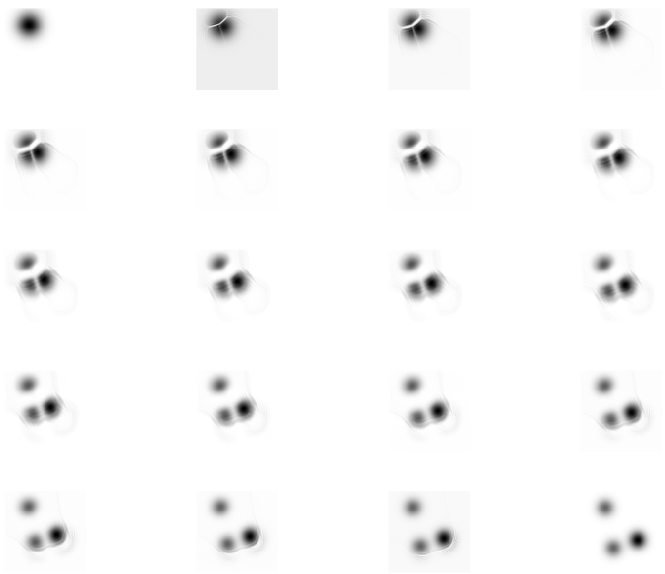


Figure 4.1: Mass-splitting phenomenon in the SDR algorithm. The initial mass is concentrated in a single Gaussian ball, which splits into three target regions during the transport process.

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

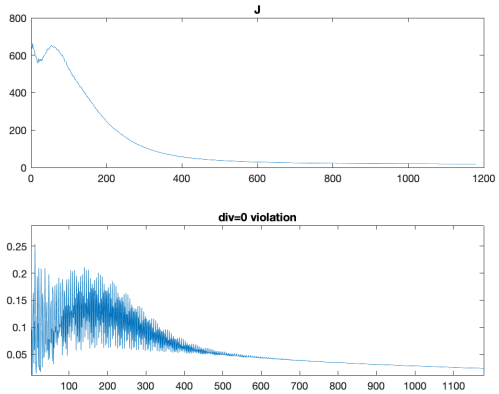


Figure 4.2: Convergence of the energy functional $\mathcal{J}_\beta^w(\mathcal{I}(U))$ for the SDR algorithm. The energy decreases steadily, indicating convergence to a stable solution.

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.

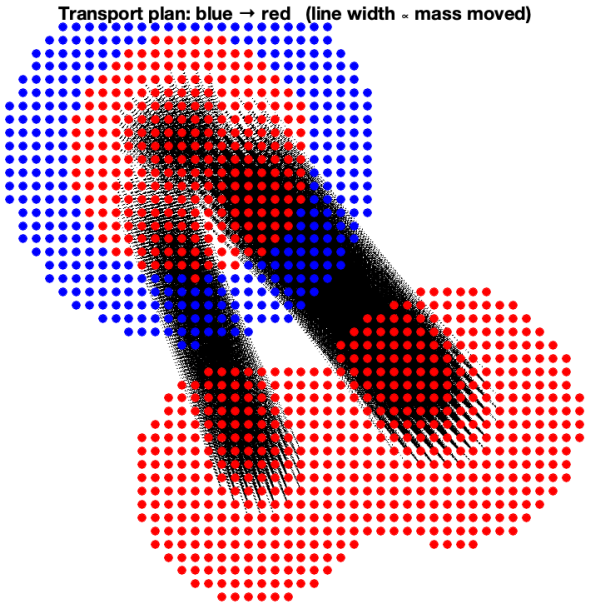


Figure 4.3: Mass-splitting phenomenon in the Sinkhorn algorithm. The initial mass is concentrated in a single Gaussian ball with blue points, which splits into three target red regions during the transport process.

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 $\|\mathbf{P}^\top \mathbf{1}_n - \mathbf{b}\| < \epsilon$.:

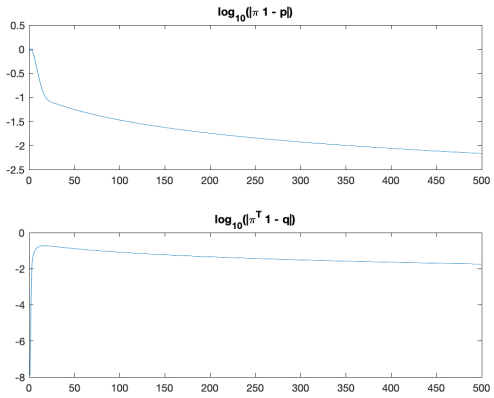


Figure 4.4: Convergence check for the Sinkhorn algorithm.

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:

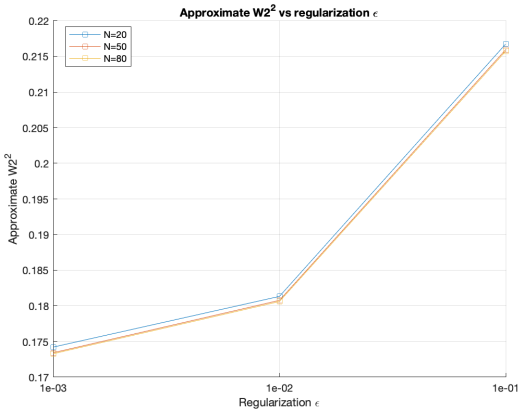


Figure 4.5: Wasserstein distance computed from the Sinkhorn algorithm. The distance is approximated using ϵ-regularized entropic Sinkhorn algorithm, which provides a stable estimate of the Wasserstein distance between the initial and target densities.

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

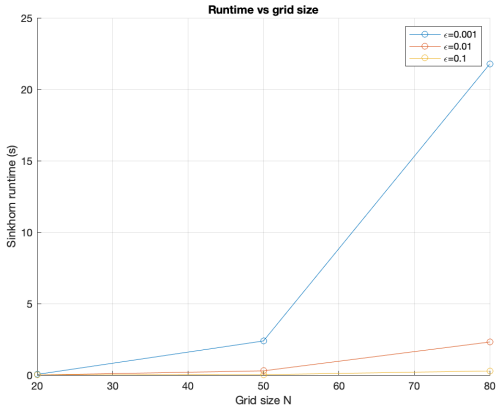


Figure 4.6: Running time of the Sinkhorn algorithm for different grid sizes and variable ϵ. The time increases with the grid size, reflecting the computational complexity of the Sinkhorn algorithm.

4.2 Numerical method with Stochastic Control Formulation

4.2.1 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:

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

The optimal feedback control is

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

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

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

with $\varphi(0)\hat{\varphi}(0) = \rho_0$, $\varphi(1)\hat{\varphi}(1) = \rho_1$.
Following (3.4), we can show that the optimal plan factorises in Sinkhorn method as:

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

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

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

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:

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

The family

$$\rho_t(x) = \phi_t(x)\,\hat{\phi}_t(x), \qquad t \in [0,1], \tag{4.4}$$

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

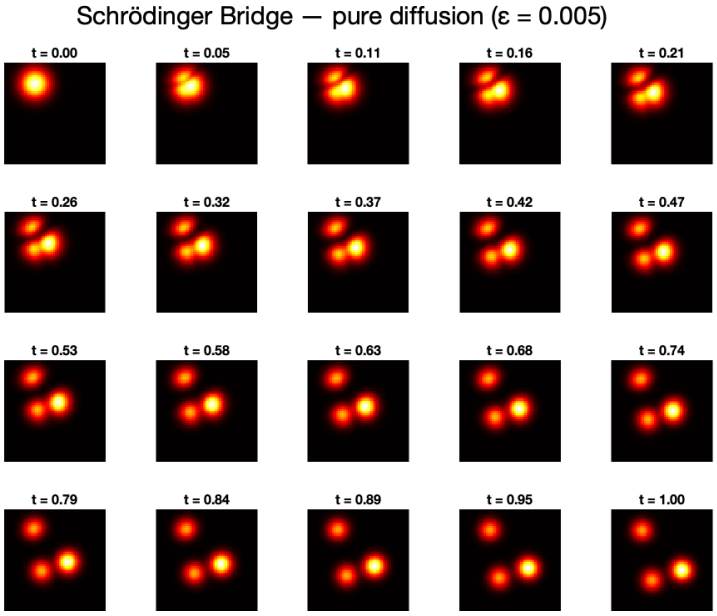


Figure 4.7: Numerical result of the Schrödinger bridge using Sinkhorn algorithm.

4.2.2 Proximal Recursor time discretisation method for the Schrödinger system

Proximal Recursor method is a powerful approach for solving the Schrödinger system. 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

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

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:

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

After one forward and one backward heat sweep we test the Hilbert-metric difference $\|\hat{\varphi}_0^{(\ell)} - \hat{\varphi}_0^{(\ell-1)}\|_\infty < \texttt{tol}$; 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 6 Forward–Backward Sinkhorn

Input: grids ρ_0, ρ_1, diffusivity ε, steps K, tolerance `tol`
Pre-compute discrete Gaussian G, set $\hat{\varphi}^{(0)} \leftarrow \mathbf{1}$
for $\ell = 1, 2, \ldots,$ `Max iterations` **do**
  Scale start: $\varphi_0^{(\ell)} \leftarrow \rho_0 / \hat{\varphi}_0^{(\ell-1)}$
  for $k = 0, 1, \ldots, K-1$ **do** ▷ forward heat
    $\varphi_{k+1}^{(\ell)} \leftarrow G * \varphi_k^{(\ell)}$
  end for
  Scale end: $\hat{\varphi}_K^{(\ell)} \leftarrow \rho_1 / \varphi_K^{(\ell)}$
  for $k = K, K-1, \ldots, 1$ **do** ▷ backward heat
    $\hat{\varphi}_{k-1}^{(\ell)} \leftarrow G * \hat{\varphi}_k^{(\ell)}$
  end for
  if $\max |\hat{\varphi}_0^{(\ell)} - \hat{\varphi}_0^{(\ell-1)}| <$ `tol` **then**
    break
  end if
end for
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 $\varepsilon = 0.005$.

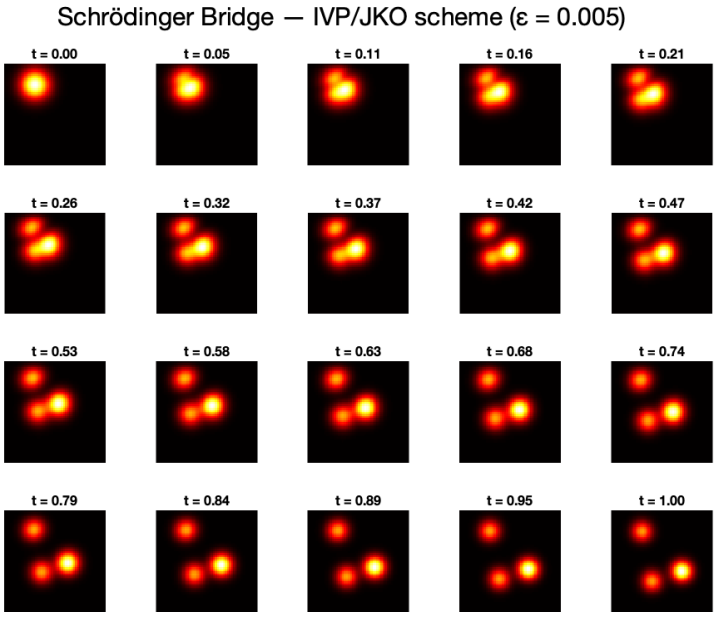


Figure 4.8: Numerical result of the Schrödinger bridge using JKO algorithm.

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

4.3 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 $\varepsilon \to 0$, 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.

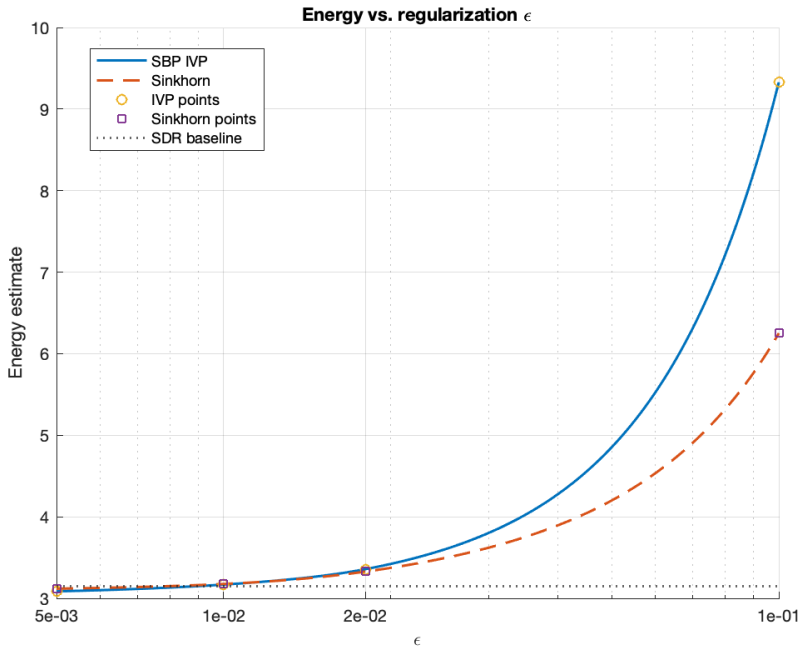
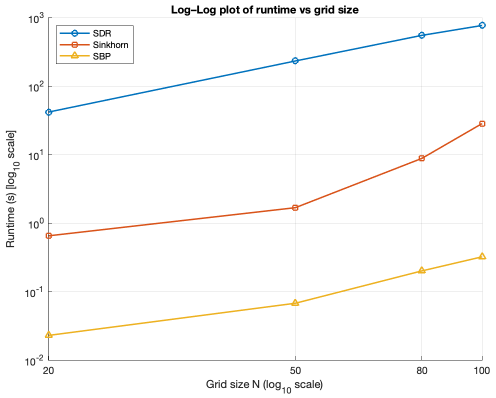
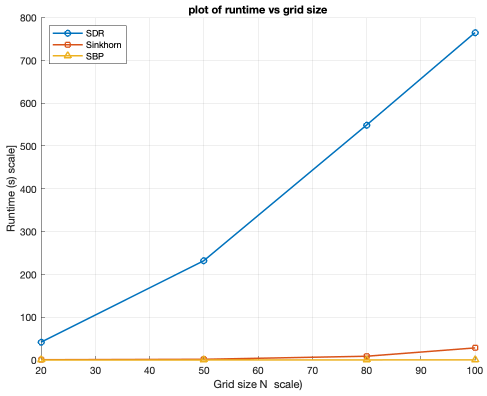


Figure 4.9: Energy estimates $\mathcal{J}(u,\rho)$ versus regularization ε for the three methods. The dashed horizontal line marks the PPXA baseline at $\varepsilon = 0$. All curves coincide for small ε, demonstrating consistent convergence to the same minimal-energy solution.

Computational Cost We assess how the computation time of PPXA, Sinkhorn and SBP -JKO 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.

4.4 Optimal Transport with Obstacles

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 $N = P = 50$. The temporal dimension is discretized into $Q = 2N = 100$ 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:

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

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.

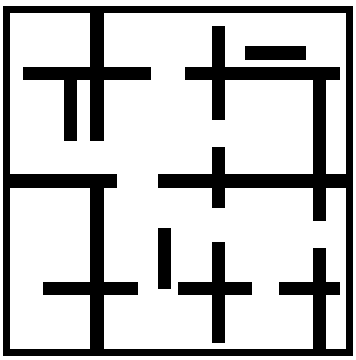


Figure 4.11: Maze-like obstacle domain used in the experiment. White regions are free space; black regions are impassable obstacles.

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

- Regularization parameter $\beta = 1$.
- Proximal variable $\tau = 3/230$.
- Number of iterations $K = 1000$.
- Initial density ρ_0 and target density ρ_1 as defined above.
- Algorithm convergence tolerance $\epsilon = 10^{-2}$.

The result of the numerical experiment is shown below: The algorithm totally takes 250 iterations to satisfy the convergence condition, with relative error $\|(m,\rho)_{n+1} - (m,\rho)_n\|/\|(m,\rho)_n\| = 9.879999e-03 < 10^{-2}$. And the energy functional $\mathcal{J}_\beta^w(\mathcal{I}(U))$

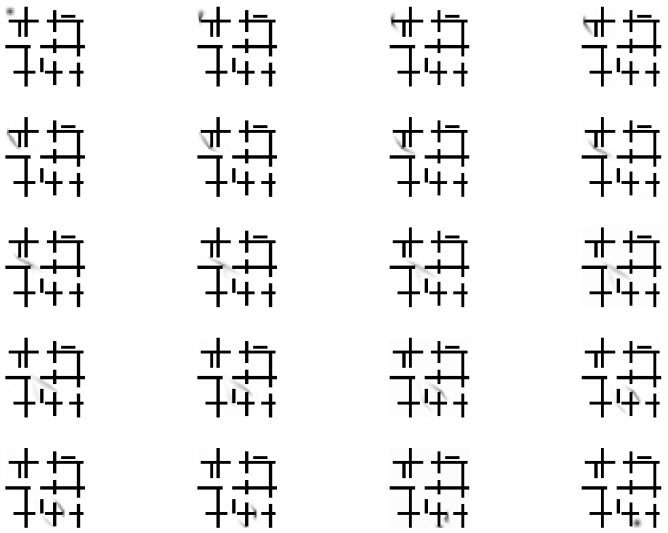


Figure 4.12: Optimal transport result for the maze-like obstacle domain. The initial density f_0 is shown in the left panel, and the target density f_1 is shown in the right panel. The mass is transported from the source to the target while avoiding obstacles.

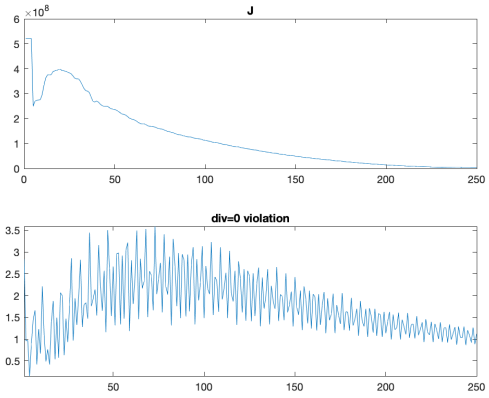


Figure 4.13: Convergence of the energy functional $\mathcal{J}_\beta^w(\mathcal{I}(U))$ during the optimization process. The energy decreases steadily, indicating convergence to a stable solution.

converges, which is shown in the following figure: 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.

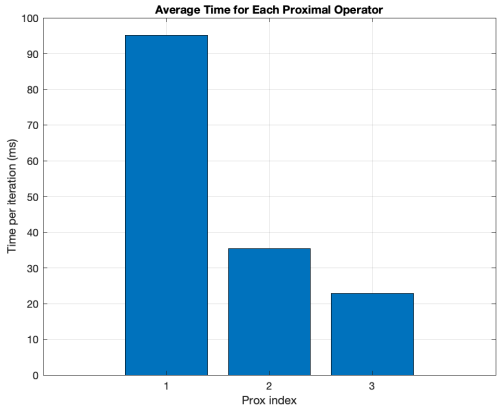


Figure 4.14: Bar chart of average runtime per proximal operator. Prox 1 dominates the cost per iteration.

This analysis clearly indicates that the majority of computation time is spent in Newton iteration steps.

4.5 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

$$\rho_0(x) \;\propto\; \exp\Big(-\tfrac{1}{2}(x-\mu_0)^\top \Sigma_0^{-1}(x-\mu_0)\Big), \quad \mu_0 = [0,0],\ \Sigma_0 = \mathrm{diag}(0.3, 0.4).$$

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

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

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

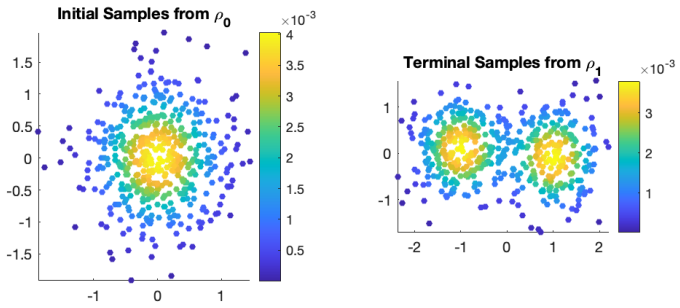


Figure 4.15: Initial and target distributions for the Schrödinger bridge experiment. The initial distribution is a single Gaussian (left), while the target is a bimodal Gaussian mixture (right).

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.

Sample Trajectories and PDF Contours Figure 4.16 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.

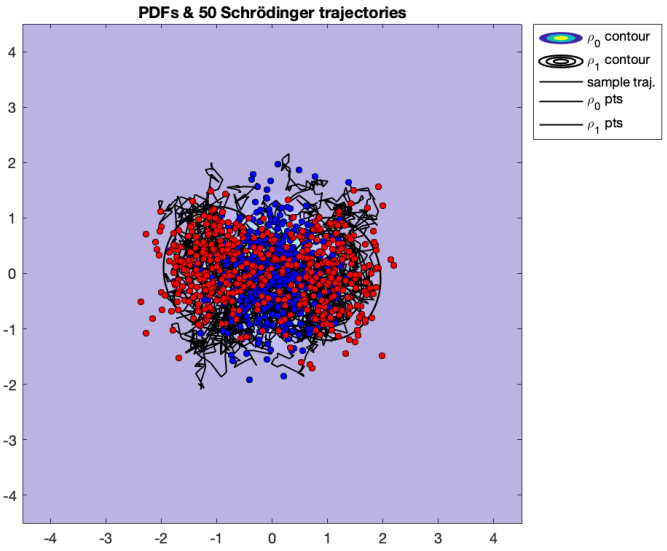


Figure 4.16: Fifty Schrödinger bridge sample trajectories. Filled contours: ρ_0; solid contours: ρ_1.

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

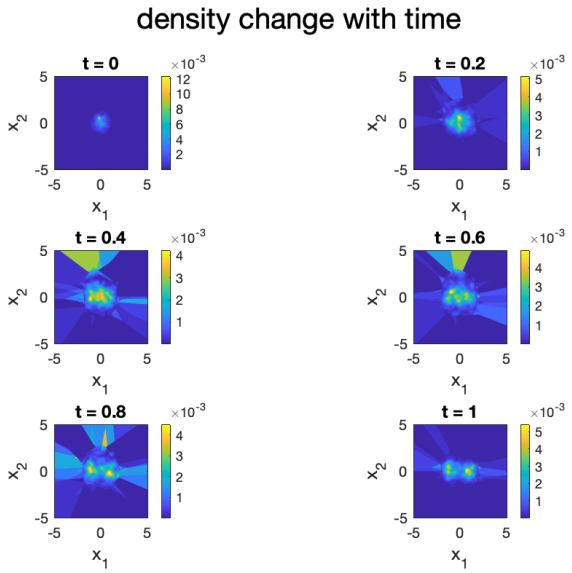


Figure 4.17: Snapshots of the Schrödinger bridge density at six time points.

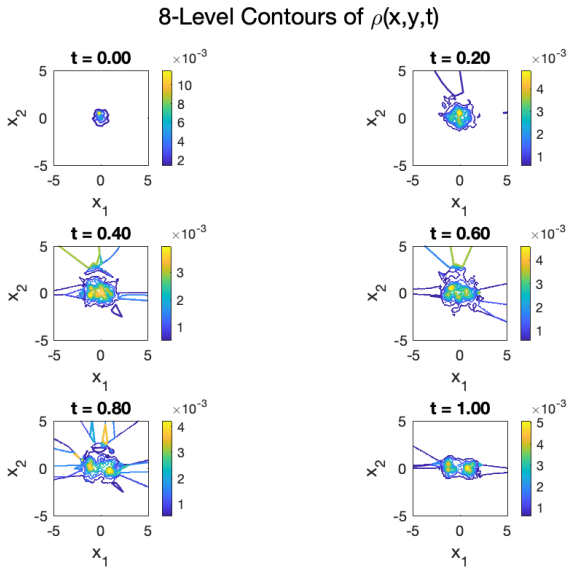


Figure 4.18: Snapshots of the Schrödinger bridge density at six time points with Countour plots(continued).

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

In the following time, I will further explore methods for solving obstacle problems in optimal transport, particularly focusing on the Schrödinger bridge framework. Additionally, I will focus on another new method- primal dual method with sinkhorn algorithm in solving dynamic optimal transport problems.

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

Arne Beurling. An automorphism of product measures. *Annals of Mathematics*, 72 (1):189–200, 1960. pages 3

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 3, 31

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, 2021. pages 3, 30

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

Paolo Dai Pra. A stochastic control approach to reciprocal diffusion processes. *Applied mathematics and Optimization*, 23(1):313–329, 1991. 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

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

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 6

Benton Jamison. The markov processes of schrödinger. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 32(4):323–331, 1975. pages 3

Leonid V Kantorovich. On a problem of monge. In *CR (Doklady) Acad. Sci. URSS (NS)*, volume 3, pages 225–226, 1948. pages 1

Nicolas Papadakis, Gabriel Peyré, and Edouard Oudet. Optimal transport with proximal splitting. *SIAM Journal on Imaging Sciences*, 7(1):212–238, 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 1

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 2

Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967. pages 2

Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021. pages 1