Numerical Analysis of the 1-D Parabolic Optimal Transport Problem

Abby Brauer
Lewis and Clark College
abrauer@lclark.edu

Megan Krawick
Youngstown State University
mekrawick@student.ysu.edu

Manuel Santana
Utah State University
manuelarturosantana@gmail.com

Advisors:
Farhan Abedin
abedinf1@msu.edu
Michigan State University

Jun Kitagawa
kitagawa@math.msu.edu
Michigan State University

Abstract

Numerical methods for the optimal transport problem is an active area of research. Recent work of Kitagawa and Abedin shows that the solution of a time-dependent equation converges exponentially fast as time goes to infinity to the solution of the optimal transport problem. This suggests a fast numerical algorithm for computing optimal maps; we investigate such an algorithm here in the 1-dimensional case. Specifically, we use a finite-difference scheme to solve the time-dependent optimal transport problem and carry out an error analysis of the scheme. A collection of numerical examples is also presented and discussed.

1 Introduction

1.1 The Optimal Transport Problem

The centuries old optimal transport problem asks how to find the cheapest way to transport materials from a given source to a target location [7]. In the 1-dimensional case and for the quadratic cost function, the mathematical formulation of the problem is as follows. Let $[A, B], [C, D] \subset \mathbb{R}$ be bounded intervals, representing, respectively, the source and target domains. Consider two positive functions $f : [A, B] \to \mathbb{R}$ and $g : [C, D] \to \mathbb{R}$ satisfying the condition

$$\int_A^B f(x) = \int_C^D g(x) = 1. \tag{1.1.1}$$

The function f can be thought of as describing the mass distribution of a pile of dirt while g describes the depth of the hole the dirt is intended to fill. The mass balance condition (1.1.1) encodes the fact that the amount of dirt is equal to the size of the hole.

Copyright © SIAM
Unauthorized reproduction of this article is prohibited

Define the class of admissible transport maps

$$\mathbb{M} = \left\{ S : [A,B] \to [C,D] \text{ satisfying } \int_{S^{-1}(E)} f(x)\ dx = \int_E g(y)\ dy \text{ for any open set } E \subset [C,D] \right\}.$$

For any $S \in \mathbb{M}$, define the total cost of S to be the quantity

$$\mathcal{C}(S) = \int_A^B |x - S(x)|^2\ dx.$$

The optimal transport problem is to find a map $T \in \mathbb{M}$ that minimizes the total cost among all maps $S \in \mathbb{M}$, i.e.

$$\mathcal{C}(T) = \min_{S \in \mathbb{M}} \mathcal{C}(S).$$

By a celebrated result of Brenier [5], under appropriate conditions on f and g, the optimal map T exists and is unique. In addition, $T(x) = u'(x)$ where $u : [A,B] \to \mathbb{R}$ is a convex function that satisfies the boundary-value problem

$$\begin{cases} u''(x) = \frac{f(x)}{g(u'(x))}, \\ u'(A) = C,\ u'(B) = D. \end{cases} \tag{O-T}$$

Notice that (O-T) implies

$$\int_A^x f(p)\ dp = \int_A^x g(u'(p))u''(p)\ dp = \int_{u'(A)}^{u'(x)} g(q)\ dq = \int_C^{u'(x)} g(q)\ dq.$$

If we define the cumulative distribution functions of f and g, respectively, as

$$F(x) := \int_A^x f(p)\ dp, \quad x \in [A,B], \qquad G(y) := \int_C^y g(q)\ dq, \quad y \in [C,D], \tag{1.1.2}$$

we then have the relation

$$F(x) = G(u'(x)) \quad \text{for all } x \in [A,B].$$

Since g is positive on $[C,D]$, we have $G'(y) = g(y) > 0$, so G is strictly increasing, hence invertible. Therefore, the optimal map T can be expressed in terms of F and G as

$$T(x) = u'(x) = G^{-1}(F(x)) \quad \text{for all } x \in [A,B]. \tag{1.1.3}$$

In practice, given f and g, it is difficult to compute F and G analytically. This provides motivation to develop alternate numerical methods of obtaining the optimal map T. Much work has been done on the numerical approximation of optimal maps in low dimensions [2, 3, 8]. Here we consider an approach based on a time-dependent version of (O-T) studied in [1, 6] and referred to as the *parabolic optimal transport problem.* In our setting, this problem can be stated as follows: find a time-dependent function $v(t,x)$ that satisfies

$$\begin{cases} v_t = \log(v_{xx}) - \log\left(\frac{f(x)}{g(v_x)}\right) & \text{in } (0,\infty) \times (A,B), \\ v_x(t,A) = C, \quad v_x(t,B) = D & \text{for all } t \geq 0, \\ v(0,x) = u_0(x), \\ v(t,\cdot) \text{ strictly convex for all } t \geq 0. \end{cases} \tag{Parabolic O-T}$$

Here, $u_0(x)$ is a given convex function on $[A,B]$ that satisfies $u_0'(A) = C$ and $u_0'(B) = D$. It is shown in [6] that $\lim_{t\to\infty} v(t,x) = u(x)$ where u solves (O-T). The more recent work [1] shows the convergence is exponentially fast in t.

1.2 Discretization of the Problem

The purpose of this paper is to carry out a numerical approximation of (Parabolic O-T) and study a number of examples. To numerically approximate (Parabolic O-T), we choose to use a finite difference scheme. This requires discretizing the interval $[A, B]$ using $J \in \mathbb{N}$ grid points

$$x_j = A + \frac{j(B-A)}{J}, \quad j = -1, \dots, J+1.$$

The range of indices is chosen this way in order to provide an extra point outside each end of the interval $[A, B]$. We will use the notation

$$\Delta x = \frac{B-A}{J}$$

to denote the spatial grid resolution and use the short-hand $\Delta x^2 := (\Delta x)^2$. To discretize the time interval $[0, \infty)$, we let $\{t_n\}_{n=0}^{\infty}$ be a non-negative sequence of strictly increasing time values with $t_0 = 0$. Denote the n-th time step by

$$\Delta t_n = t_{n+1} - t_n.$$

We denote by $\mathcal{G}$ the set of all grid points $\{(t_n, x_j) : n \in \mathbb{N},\ j \in -1, \dots, J+1\}$.

In order to motivate the finite difference scheme, we recall the following consequences of the Taylor Remainder Theorem:

$$v_x(t_n, x_j) = \frac{v(t_n, x_{j+1}) - v(t_n, x_{j-1})}{2\Delta x} - \frac{v_{xxx}(t_n, x_j + \psi) + v_{xxx}(t_n, x_j - \psi)}{12}\Delta x^2 \quad \text{for some } \psi \in (0, \Delta x), \tag{1.2.1}$$

$$v_{xx}(t_n, x_j) = \frac{v(t_n, x_{j+1}) + v(t_n, x_{j-1}) - 2v(t_n, x_j)}{\Delta x^2} - \frac{v_{xxxx}(t_n, x_j + \gamma) + v_{xxxx}(t_n, x_j - \gamma)}{24}\Delta x^2 \quad \text{for some } \gamma \in (0, \Delta x), \tag{1.2.2}$$

$$v_t(t_n, x_j) = \frac{v(t_{n+1}, x_j) - v(t_n, x_j)}{\Delta t_n} - \frac{v_{tt}(t_n + \kappa, x_j)}{2}\Delta t_n \quad \text{for some } \kappa \in (0, \Delta t_n). \tag{1.2.3}$$

Our goal is to construct a grid function $U : \mathcal{G} \to \mathbb{R}$ such that $U(t_n, x_j) \approx v(t_n, x_j)$ for $j = 0, \dots J$ and $n \in \mathbb{N}$, where v is the solution of (Parabolic O-T). We will, from here onward, use the short-hand $v_j^n = v(t_n, x_j)$ and $U_j^n = U(t_n, x_j)$. Neglecting the terms of order Δt_n and Δx^2 in the Taylor expansions of v above, we obtain the definition of the first and second order finite difference operators acting on the approximation U. For convenience we will define two operators for some arbitrary function ϕ.

Definition 1.1. *The first order centered difference operator ∇_j^n and the second order centered difference operator Δ_j^n are defined as*

$$\nabla_j^n \phi := \frac{\phi_{j+1}^n - \phi_{j-1}^n}{2\Delta x}, \quad j = 0, \dots J,$$

$$\Delta_j^n \phi := \frac{\phi_{j+1}^n + \phi_{j-1}^n - 2\phi_j^n}{\Delta x^2} \quad j = 0, \dots J.$$

The approximation for v_{xx} at the boundary points $x = A, B$ requires using the boundary conditions in (Parabolic O-T). We use a backward difference first space derivative approximation for the boundary at A and a forward difference approximation for the first space derivative at B with the exact values for these derivatives as given by the Neumann boundary conditions,

$$C = v_x(t_n, A) = \frac{v(t_n, x_1) - v(t_n, x_{-1})}{2\Delta x} - \frac{v_{xxx}(t_n, A + \psi) + v_{xxx}(t_n, A - \psi)}{12}\Delta x^2 \quad \text{for some } \psi \in (0, \Delta x) \tag{1.2.4}$$

$$D = v_x(t_n, B) = \frac{v(t_n, x_{J+1}) - v(t_n, x_{J-1})}{2\Delta x} - \frac{v_{xxx}(t_n, B + \psi) + v_{xxx}(t_n, B - \psi)}{12}\Delta x^2 \quad \text{for some } \psi \in (0, \Delta x). \tag{1.2.5}$$

Utilizing the definition of the finite difference method presented in [9], we implement the following centered difference approximation of (Parabolic O-T)

$$\begin{cases} U_j^{n+1} = \left(\log\left(\Delta_j^n U\right) - \log\left(\frac{f(x_j)}{g(\nabla_j^n U)}\right) \right)\Delta t_n + U_j^n, \quad j = 0, \dots J, \\ \nabla U_0^n = C, \quad \nabla U_J^n = D, \\ U_{-1}^n := U_1^n - 2C\Delta x, \quad U_{J+1}^n = U_{J-1}^n + 2D\Delta x, \\ U_j^0 = u_0(x_j). \end{cases} \tag{F-D}$$

1.3 Structure of the Paper

The remainder of this paper is structured as follows. In Section 2 we show that the error between our numerical approximation and the true solution of (Parabolic O-T) is bounded by quantities depending on previous time-steps. In Section 3 we show how to measure the asymptotic closeness of (F-D) to (O-T). Finally in Section 4 we discuss the code for implementation of the finite difference scheme as well as numerical findings and applications to quantile functions. The proofs of explicit derivative bounds for the solution of (Parabolic O-T) used in calculations are given in [4] appendix.

2 Error Analysis of Finite Difference Scheme

In order for a numerical approximation to be effective the error at a given time step must be bounded by quantities known from the previous steps. In this section we establish such error bounds for (F-D). Recall $\Delta_j^n U$ is a finite approximation of the second derivative and therefore has some error. Although we would expect $\Delta_j^n U$ to stay positive because it is approximating a convex function, how to guarantee this is not yet clear. Therefore we must assume the condition of $\Delta_j^n U$ staying positive for all n in the following error analysis.
We first define the infinity norm $|| \cdot^n ||_\infty$ to be the maximum value of $\cdot$ at time step n for all points in $\mathcal{G}$. We will also now define several bounds on the derivatives of v.

Definition 2.1. *Define the derivative bounds K, Γ, Ψ δ_1, and δ_2 to satisfy*

$$\begin{aligned} K &\geq |v_{tt}(t,x)|, \\ \Gamma &\geq |v_{xxxx}(t_n, x_j)|, \\ \Psi &\geq |v_{xxx}(t_n, x_j)|, \\ 0 < \delta_1 &\leq v_{xx}(t,x) \leq \delta_2. \end{aligned}$$

for all $t \in [0,\infty)$ and $x \in [A,B]$.

It is possible to explicitly calculate the constants $K, \Gamma, \Psi, \delta_1, \delta_2$ in terms of the mass distributions f, g and the initial function v_0. A full discussion of these calculations is beyond the scope of this paper; however, they are available in the appendix of [4].

We are now prepared to state our error bound for our numerical scheme.

Theorem 2.1. *Assuming $\Delta_j^n U$ remains positive at every time step n, positive time and spatial grid steps, and that the following is true for Δx and Δt:*

$$\Delta x = \min\left\{ \frac{3\delta_1}{2\Psi}, \sqrt{\frac{6\delta_1}{\Gamma}} \right\},$$

$$\frac{\max|g'(y)|\Delta t_n}{2\min g(y)\Delta x} \leq \frac{\Delta t_n}{\Delta x^2 \min\left\{\frac{1}{2}\delta_1, \min\{\Delta_j^n U\}\right\}} \leq \frac{1}{2}, \tag{2.0.1}$$

(F-D) *has the following maximum error bound on the interior points (points not on the boundary, $j = 1, \dots, J-1$) at time step n.*

$$||U^n - v^n||_\infty \leq \sum_{i=0}^{n-1} \left(\Delta t_i \left(\frac{\Delta t_i}{2} K + \frac{\Delta x^2}{6\delta_1} \Gamma \right) + \frac{\max |g'|}{\min g} \left(\frac{\Delta x^2}{6} \Psi \right) \right) \tag{2.0.2}$$

This theorem shows that our finite difference scheme is close to the real solution of (Parabolic O-T) for all t_n. We will spend the rest of Section 2 proving Theorem 2.1. For a more formal discussion of the efficacy of finite difference schemes see [9].

2.1 Calculation of Local Error

In this section we prove the first necessary lemma for proving Theorem 2.1. We start with several definitions.

Definition 2.2. *(i) Local approximation:*

$$V_j^{n+1} := \left(\log\left(\Delta_j^n v\right) - \log\left(\frac{f(x_j)}{g\left(\nabla_j^n v\right)} \right) \right) \Delta t_n + v_j^n, \quad j = 1, \dots, J-1.$$

(ii) Local error τ at grid point (x_j, t_n):

$$\tau_j^n := V_j^n - v_j^n.$$

(iii) Local discretization error θ at grid point (x_j, t_n):

$$\theta_j^n := \frac{v_j^{n+1} - v_j^n}{\Delta t_n} - \log\left(\Delta_j^n v\right) + \log\left(\frac{f(x_j)}{g(\nabla_j^n v)} \right). \tag{2.1.1}$$

With these definitions in hand we state the first lemma necessary for proving Theorem 2.1:

Lemma 2.1. *Assuming the conditions* (2.0.1)*, the local error τ for any point on time step n has an upper bound*

$$||\tau^{n+1}||_\infty \leq \Delta t_n \left(\frac{\Delta t_n K}{2} + \frac{\Delta x^2 \Gamma}{6\delta_1} + \frac{\Delta x^2 \max |g'|\Psi}{6 \min g} \right)$$

Proof. First we note that local discretization error provides a useful identity

$$V_j^{n+1} - v_j^{n+1} = \tau_j^{n+1} = -\Delta t_n \theta_j^n. \tag{2.1.2}$$

Therefore, to calculate the bounds on the local error, τ_j^{n+1}, we first need to estimate θ_j^n. Using (Parabolic O-T), we can substitute for $f(x_j)$ and get

$$\theta_j^n = \frac{v_j^{n+1} - v_j^n}{\Delta t_n} - v_t(t_n, x_j) - \left(\log\left(\Delta_j^n v\right) - \log(v_{xx}(t_n, x_j))\right) - \left(\log\left(g\left(\nabla_j^n v\right)\right) - \log\left(g\left(v_x(t_n, x_j)\right)\right)\right). \tag{2.1.3}$$

From (1.2.3), we have

$$\frac{v_j^{n+1} - v_j^n}{\Delta t_n} - v_t(t_n, x_j) = \frac{v_{tt}(t_n + \kappa, x_j)}{2} \Delta t_n. \tag{2.1.4}$$

To simplify $\log\left(\Delta_j^n v\right) - \log(v_{xx}(t_n, x_j))$ we use the Mean Value Theorem to find a number η between $v_{xx}(t_n, x_j)$ and $\Delta_j^n v$ such that

$$\log\left(\Delta_j^n v\right) - \log(v_{xx}(t_n, x_j)) = \frac{1}{\eta}(\Delta_j^n v - v_{xx}(t_n, x_j)),$$

The Taylor approximation (1.2.2) then shows

$$\log\left(\Delta_j^n v\right) = \log(v_{xx}(t_n, x_j)) + \frac{\Delta x^2}{\eta} \left(\frac{v_{xxxx}(t_n, x_j + \gamma) + v_{xxxx}(t_n, x_j - \gamma)}{24} \right). \tag{2.1.5}$$

The Mean Value Theorem implies there exists a number μ between $g(v_x(t_n, x_j))$ and $g\left(\nabla_j^n v\right)$, and a number χ between $g'(v_x(t_n, x_j))$ and $g'(\nabla_j^n v)$ such that

$$\log\left(g\left(\nabla_j^n v\right)\right) - \log\left(g\left(v_x(t_n, x_j)\right)\right) = \frac{1}{\mu}\left(g\left(\nabla_j^n v\right) - g(v_x(t_n, x_j))\right) = \frac{\chi}{\mu}\left(\nabla_j^n v - v_x(t_n, x_j)\right).$$

Using the Taylor expansion (1.2.1), we find that

$$\log\left(g\left(\nabla_j^n v\right)\right) - \log\left(g\left(v_x(t_n, x_j)\right)\right) = \frac{\chi \Delta x^2}{\mu}\left(\frac{v_{xxx}(t_n, x_j + \psi) + v_{xxx}(t_n, x_j - \psi)}{12}\right). \tag{2.1.6}$$

Substituting (2.1.4), (2.1.5) and (2.1.6) into (2.1.3) gives us

$$\begin{aligned}\tau_j^{n+1} =& -\Delta t_n \theta_j^n \\ =& \Delta t_n \left(-\frac{\Delta t_n}{2} v_{tt}(t_n + \kappa, x_j) + \frac{\Delta x^2}{24\eta}\left(v_{xxxx}(t_n, x_j + \gamma) + v_{xxxx}(t_n, x_j - \gamma)\right) + \frac{\Delta x^2 \chi}{12\mu}\left(v_{xxx}(t_n, x_j + \psi) + v_{xxx}(t_n, x_j - \psi)\right)\right).\end{aligned} \tag{2.1.7}$$

Using the derivative bounds in (2.1.7) shows

$$|\tau_j^{n+1}| \leq \Delta t_n \left(\frac{K \Delta t_n}{2} + \frac{\Delta x^2 \Gamma}{12\eta} + \frac{\Delta x^2 \Psi \chi}{6\mu}\right). \tag{2.1.8}$$

If we now choose Δx to satisfy

$$\Delta x^2 \leq \frac{6\delta_1}{\Gamma}, \tag{2.1.9}$$

we obtain the inequality

$$\frac{1}{12} \max_{x,t} |v_{xxxx}| \Delta x^2 \leq \frac{1}{2} \min_{x,t} v_{xx}.$$

Then by the Taylor expansion (1.2.2), we obtain

$$\min_j \{\Delta_j^n v\} \geq \frac{1}{2} \min_{x,t} v_{xx},$$

hence

$$\eta \geq \min_j \left\{\Delta_j^n v, v_{xx}(t_n, x_j)\right\} \geq \frac{\delta_1}{2}.$$

Additionally we know χ is bounded from above by $\max |g'|$ and μ is bounded from below by $\min g$. Substituting the constants K, Γ,Ψ, and δ_1 into (2.1.8) and using the triangle inequality with our g bounds we obtain

$$|\tau_j^{n+1}| \leq \Delta t_n \left(\frac{\Delta t_n K}{2} + \frac{\Delta x^2 \Gamma}{6\delta_1} + \frac{\Delta x^2 \max |g'| \Psi}{6 \min g}\right) \quad \text{for all } j \in \{1, \ldots, J-1\}. \tag{2.1.10}$$

under the restriction (2.1.9). Finally we can take the max norm of τ_j^{n+1} over all $j = 1, \ldots, J-1$ to obtain

$$||\tau^{n+1}||_\infty \leq \Delta t_n \left(\frac{\Delta t_n K}{2} + \frac{\Delta x^2 \Gamma}{6\delta_1} + \frac{\Delta x^2 \max |g'| \Psi}{6 \min g}\right) \tag{2.1.11}$$

□

2.2 Calculation of Total Error

In this section we prove an additional lemma. We must first define another error term.

Definition 2.3. *Define the error term ε*

$$\varepsilon_j^n := U_j^{n+1} - V_j^{n+1}.$$

Lemma 2.2. *Assuming $\Delta_j^n U$ stays positive for all n and the conditions in 2.0.1, the ε is bounded by:*

$$||\varepsilon^n||_\infty \leq \sum_{i=0}^{n-1} \Delta t_i \left(\frac{\Delta t_i K}{2} + \frac{\Delta x^2 \Gamma}{6\delta_1} + \frac{\Delta x^2 \max |g'|\Psi}{6 \min g} \right).$$

Proof. Note by definition

$$\varepsilon_j^n = \left(\log(\Delta_j^n U) - \log(\Delta_j^n v) - \left(\log\left(g\left(\nabla_j^n v\right)\right) - \log\left(g\left(\nabla_j^n U\right)\right)\right)\right) \Delta t_n + (U_j^n - v_j^n). \tag{2.2.1}$$

By the Mean Value Theorem, there is some number ρ between $g(\nabla_j^n U)$ and $g\left(\nabla_j^n v\right)$, and another number ω between $g'(\nabla_j^n U)$ and $g'(\nabla_j^n v)$ such that

$$\log\left(g\left(\nabla_j^n v\right)\right) - \log\left(g\left(\nabla_j^n U\right)\right) = \frac{\omega}{2\Delta x \rho}\left(v_{j+1}^n - U_{j+1}^n + v_{j-1}^n - U_{j-1}^n\right). \tag{2.2.2}$$

Similarly, there is some numnber ξ between $\Delta_j^n v$ and $\Delta_j^n U$ such that

$$\log(\Delta_j^n U) - \log(\Delta_j^n v) = \frac{1}{\xi}(\Delta_j^n U - \Delta_j^n v). \tag{2.2.3}$$

Substituting (2.2.3) and (2.2.2) into (2.2.1) shows

$$\varepsilon_j^{n+1} = \left(\frac{1}{\xi}(\Delta_j^n U - \Delta_j^n V) + \frac{\omega}{2\Delta x \rho}\left(v_{j+1}^n - U_{j+1}^n + v_{j-1}^n - U_{j-1}^n\right)\right) \Delta t_n + (U_j^n - V_j^n).$$

To continue we must bound ξ from below by known values. Note that the following inequality holds true.

$$\xi \geq \min\left\{\min_j\{\Delta_j^n v\}, \min_j\{\Delta_j^n U\}\right\}.$$

If (2.1.9) holds, then $\min_j\{\Delta_j^n v\} \geq \frac{1}{2}\delta_1$. Therefore,

$$\frac{1}{\xi} \leq \frac{1}{\min\left\{\frac{1}{2}\delta_1, \min\limits_j\{\Delta_j^n U\}\right\}}.$$

Let us define the quantities

$$r := \frac{\Delta t_n}{\Delta x^2 \min\left\{\frac{1}{2}\delta_1, \min\limits_j\{\Delta_j^n U\}\right\}}, \qquad s := \frac{\max|g'|\Delta t_n}{2\min g \Delta x}.$$

Notice that we can bound ρ from above by $\max|g'|$ and bound ω from below by $\min g$. Using the definition of the second order difference operator Δ_j^n and the triangle inequality, we find that

$$|\varepsilon_j^{n+1}| \leq |r - s||U_{j+1}^n - v_{j+1}^n| + |1 - 2r||U_j^n - v_j^n| + |r + s||U_{j-1}^n - v_{j-1}^n|.$$

From the triangle inequality we also know $|U_j^n - v_j^n| \leq |\varepsilon_j^n| + |\tau_j^n|$, thus

$$|\varepsilon_j^{n+1}| \leq |r - s|(|\varepsilon_{j+1}^n| + |\tau_{j+1}^n|) + |1 - 2r|(|\varepsilon_j^n| + |\tau_j^n|) + |r + s|(|\varepsilon_{j-1}^n| + |\tau_{j-1}^n|).$$

Replacing all quantities of $|\varepsilon_j^n|$ and $|\tau_j^n|$ with their maximum norms

$$|\varepsilon_j^{n+1}| \leq (|r - s| + |1 - 2r| + |r + s|)\left(||\varepsilon^n||_\infty + ||\tau^n||_\infty\right).$$

Now if we choose Δt_n and Δx such that $s < r \leq \frac{1}{2}$, then

$$\varepsilon_j^{n+1} \leq ||\varepsilon^n||_\infty + ||\tau^n||_\infty.$$

Iterating the inequality over n shows

$$||\varepsilon^n||_\infty \leq ||\varepsilon^0||_\infty + ||\tau^{n-1}||_\infty + \cdots + ||\tau^0||_\infty \quad \text{for all } n \geq 1.$$

Lastly recall that by definition $U^0 - V^0 = 0$ therefore $||\varepsilon^0||_\infty = 0$ and we have

$$||\varepsilon^n||_\infty \leq ||\tau^{n-1}||_\infty + \cdots + ||\tau^0||_\infty.$$

Using the bound for τ from (2.1.11), we finally get

$$||\varepsilon^n||_\infty \leq \sum_{i=0}^{n-1} \Delta t_i \left(\frac{\Delta t_i K}{2} + \frac{\Delta x^2 \Gamma}{6\delta_1} + \frac{\Delta x^2 \max|g'|\Psi}{6 \min g} \right). \tag{2.2.4}$$

□

We now have all the tools at hand to prove Theorem 2.1.

Proof of Theorem 2.1. Using the triangle inequality and taking the maximum norm we can say.

$$||U_j^n - v(t_n, x_j)||_\infty \leq ||\tau_j^n||_\infty + ||\varepsilon_j^n||_\infty$$

By Lemmas 2.1 and 2.2,

$$||U^n - v^n||_\infty \leq \sum_{i=0}^{n-1} \left(\Delta t_i \left(\frac{\Delta t_i}{2} K + \frac{\Delta x^2}{6\delta_1} \Gamma \right) + \frac{\max|g'|}{\min g} \left(\frac{\Delta x^2}{6} \Psi \right) \right). \tag{2.2.5}$$

□

2.3 Addressing Boundary Conditions

Due to the Neumann boundary conditions we must analyze error conditions at the boundaries separately. This process is nearly identical as sections 2.1 and 2.2. Additionally calculations on the left and right boundary are almost identical, and we will just focus on the left boundary point.

We begin with our calculation of the local error. Using the same method of section 3.1, the error term from (1.2.3), and (1.2.4) gives us

$$\tau_0^n = \Delta t_n \left(-\frac{\Delta t_n}{2}(v_{tt}(t_n + \kappa, 0)) + \frac{1}{\lambda} \left(\frac{\Delta x}{3}(v_{xxx}(t_n, \psi) \right) + \frac{\chi}{\mu} \left(\frac{\Delta x^2}{6} v_{xxx}(t_n, x_j + \gamma) \right) \right),$$

with λ being some point between $\Delta_0^n v$ and $v_{xx}(t_n, 0)$. We assume Δx in the boundary case must also satisfy $\frac{1}{3}|v_{xxx}|\Delta x \leq \frac{1}{2} v_{xx}$, which leads to

$$\min_j \left\{ \Delta_j^n v \right\} \geq \frac{1}{2} \min_{x,t} \{v_{xx}\}.$$

This implies that $\lambda < \frac{1}{2}\delta_1$ by similar methods as used before, and results in a bound on τ_j^n at boundary point a

$$|\tau_0^n| \leq \Delta t_n \left(\frac{\Delta t_n}{2} K + \frac{2\Delta x}{3(\delta_1)} \Psi + \frac{\max|g'|}{\min g} \left(\frac{\Delta x^2}{6} \Psi \right) \right).$$

Now calculating our total error at the boundary begins the same way as in section 3.2. If we let r and s be the same quantities then, by use of a similar log approximation as well as the triangle inequality

$$|\varepsilon_0^n| \leq |r + s||U_1^{n-1} - V_1^{n-1}| + |1 - r - s||U_0^{n-1} + V_0^{n-1}|.$$

We can now replace $|U_0^n - V_0^n|$ terms with $|\varepsilon_0^n| + |\tau_0^n|$ terms by another triangle inequality

$$|\varepsilon_0^n| \leq |r + s|(|\varepsilon_1^{n-1}| + |\tau_1^{n-1}|) + |1 - r - s|(|\varepsilon_0^{n-1}| + |\tau_0^n|).$$

If $s < r \leq \frac{1}{2}$ as was necessary for (2.2) then it is implied that $r + s < 1$. Taking maximums over j in $\{0, 1\}$ gives us

$$|\varepsilon_0^n| \leq \max_j\{\tau_j^{n-1}\} + ... + \max_j\{\tau_j^0\}.$$

This allows us to show a final bound on the boundary conditions,

$$|U_0^n - v_0^n| \leq \sum_{i=0}^{n-1} \left(\Delta t_i \left(\frac{\Delta t_i}{2} K + \frac{\Delta x}{3\delta_1} \Psi + \frac{\max|g'|}{\min g} \left(\frac{\Delta x^2}{6} \Psi \right) \right) \right). \tag{2.3.1}$$

3 Asymptotic Error Analysis

To ensure that our implementation of the code provides an accurate numerical approximation of the optimal map $T(x) = u'(x)$, where u solves (O-T), we must show that $\max_j |\nabla_j^n U - u'(x_j)|$ is within a desired tolerance. In this section we show the error between (O-T) and (F-D) is controlled by a quantity that can be calculated at each time step.

Let $\Omega \subset [A, B]$ and let $S : \Omega \to [C, D]$ be an increasing function. Recall the definition of the cumulative distribution functions F and G in (1.1.2). Define the error function of S as

$$E(S, x) := |F(x) - G(S(x))|, \quad x \in \Omega.$$

Notice that if T is the optimal map between f and g, then $E(T, x) = 0$ for all $x \in [A, B]$. Next, we see that for any $x \in \Omega$,

$$E(S, x) = |F(x) - G(S(x))| = |G(T(x)) - G(S(x))| \geq \left(\min_{[C,D]} g\right) |T(x) - S(x)|.$$

Therefore,

$$\max_{x\in\Omega} |T(x) - S(x)| \leq \frac{\max_{x\in\Omega} E(S, x)}{\min_{[C,D]} g}.$$

It follows that for any $\sigma > 0$,

$$\max_{x\in\Omega} E(S, x) \leq \sigma \min_{[C,D]} g \quad \Rightarrow \quad \max_{x\in\Omega} |T(x) - S(x)| \leq \sigma. \tag{3.0.1}$$

Theorem 3.1. *Let $T(x) = u'(x)$ be the optimal map, where u solves* (O-T)*. Asumming $\Delta_j^n U$ is strictly positive for all n. Given a tolerance $\sigma > 0$.*

$$\max_{j=0,\ldots,J} E(\nabla_j^n U, x_j) \leq \sigma \quad \Rightarrow \max_{j=0,\ldots,J} |T(x_j) - \nabla_j^n U| \leq \frac{\sigma}{\min_{[C,D]} g} \tag{3.0.2}$$

Proof. We apply the estimate (3.0.1) to the finite-difference scheme (F-D). Let $\Omega = \{x_0, \ldots, x_J\} \subset [A, B]$ be the set of spatial grid points. For each $n \in \mathbb{N}$, denote the map $S_n : \Omega \to [C, D]$ as

$$S_n(x_j) := \nabla_j^n U, \quad x_j \in \Omega.$$

Therefore, we have $S_n(x_0) = C$ and $S_n(x_J) = D$. We check that S_n is an increasing function on Ω, and hence maps into $[C, D]$. This is a condition of the optimal map as implied by (O-T).

Assume $j \in \{1, \ldots J\}$. Then

$$\begin{aligned}
S_n(x_j) - S_n(x_{j-1}) &= \nabla_j^n U - \nabla_{j-1}^n U \\
&= \left(\frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x}\right) - \left(\frac{U_j^n - U_{j-2}^n}{2\Delta x}\right) \\
&= \frac{\Delta x}{2}\left[\left(\frac{U_{j+1}^n - U_{j-1}^n}{\Delta x^2}\right) - \left(\frac{U_j^n - U_{j-2}^n}{\Delta x^2}\right)\right] \\
&= \frac{\Delta x}{2}\left[\left(\frac{U_{j+1}^n + U_{j-1}^n - 2U_j^n - U_{j-1}^n + 2U_j^n - U_{j-1}^n}{\Delta x^2}\right) - \left(\frac{U_j^n - U_{j-2}^n}{\Delta x^2}\right)\right] \\
&= \frac{\Delta x}{2}\left[\left(\frac{U_{j+1}^n + U_{j-1}^n - 2U_j^n}{\Delta x^2}\right) + \left(\frac{-U_{j-1}^n + 2U_j^n - U_{j-1}^n - U_j^n + U_{j-2}^n}{\Delta x^2}\right)\right] \\
&= \frac{\Delta x}{2}\left[\left(\frac{U_{j+1}^n + U_{j-1}^n - 2U_j^n}{\Delta x^2}\right) + \left(\frac{U_j^n + U_{j-2}^n - 2U_{j-1}^n}{\Delta x^2}\right)\right] \\
&= \frac{\Delta x}{2}\left(\Delta_j^n U + \Delta_{j-1}^n U\right) \geq 0.
\end{aligned}$$

□

Notice that the Theorem above requires the condition $\Delta_j^n U \geq 0$ for all $j = 0, \ldots, J$. We note that in practice, if the code does not encounter a domain error at time step $n+1$, then $\Delta_j^n U > 0$ for all j. The theorem above also shows that given a tolerance $\sigma > 0$, if there exists some $N(\sigma) \in \mathbb{N}$ such that

$$\max_{j=0,\ldots J} E(\nabla_j^{N(\sigma)} U, x_j) \leq \sigma$$

then

$$\max_{j=0,\ldots,J} |T(x_j) - \nabla_j^{N(\sigma)} U| \leq \frac{\sigma}{\min_{[C,D]} g}. \tag{3.0.3}$$

Since the quantity $\max_j E(\nabla_j^n U, x_j)$ can be computed at each time step n, we can run our code to the time step $n = N(\sigma)$ for which $\max_j E(\nabla_j^n U, x_j)$ is less than a specified tolerance σ. We are not able to guarantee our scheme will always be able to reach such a specified tolerance in a finite number of steps, but if this tolerance is reached, then we can conclude using (3.0.3) that the map $\nabla_j^n U$ is equal to the optimal map on the grid points $x_1, \ldots, x_J$ up to a quantifiable error. It should be noted that in practice it is often simpler to use numerical integration to evaluate $E(\nabla_j^n U, x_j)$. Therefore, (3.0.3) will hold up to the accuracy of the numerical integration method used.

4 Computational Examples and Results

This section is dedicated to describing the code used for implementing (F-D) and certain relevant numerical examples computed using this code. We first note that empirically when Δt and Δx are chosen to satisfy 2.0.1 then $\Delta_j^n U$ stays above $\frac{1}{2}\delta_1$. Additionally from [6], we know that (Parabolic O-T) converges exponentially to the actual solution of the optimal transport problem. In the following examples this fast convergence can be observed as the results are graphed over time using a uniformly spaced color gradient. Exponential convergence is observed due to the relatively small change in approximation at later time steps.

In testing our code for functionality, we attempted to cover a variety of situations using appropriate choices of f and g. Some of the more interesting cases tested have been shown here. For simplicity all cases were run with initial choice $u_0(x) = \frac{1}{2}x^2$. We chose not to graph the function U_j^n, as the function that is relevant for the optimal transport theory is $\nabla_j^n U$, which is meant to approximate the function $u'(x)$ for u solving (O-T). We note that the theory for (Parabolic O-T) only guarantees convergence to the solution of (O-T) when $f(x)$, $g(y)$ are continuous and bounded away from zero and infinity on $[A, B]$ and $[C, D]$, respectively. Some of our numerical examples test the limits of the theory by considering cases where g is only piece-wise continuous and also where f gets very close to zero.

Before discussing the examples we will briefly discuss the algorithm. The full implementation in python is available at `https://github.com/manuelarturosantana/ParabolicOptimalTransport`

4.1 Algorithm

```
Result: Returns the Approximated Solution of OT
Calculate $\Delta x, \Delta t_0, \delta_1, \delta_2, \Psi, K, \Gamma$
current row = initial row based off $v_0$
while Max Error > Tolerance do
    Calculate $\Delta t_n$
    current row = calculate next row
    Calculate Max Error
end
```

Algorithm 1: Simple Finite Difference Algorithm

Calculating each row follows the finite difference scheme (F-D). The boundaries and the interior points are calculated separately. Checking the error at every grid point for every time-step is computationally expensive.

To combat this, a subset of the spatial grid points, which we denote by $\mathcal{G}_x$, is selected, and at each time step only grid points in $\mathcal{G}_x$ are tested to be within tolerance. If at a certain time step all grid points in $\mathcal{G}_x$ are within tolerance, we proceed by calculating the error at *all* spatial grid points from that time step onward until tolerance is reached at every grid point.

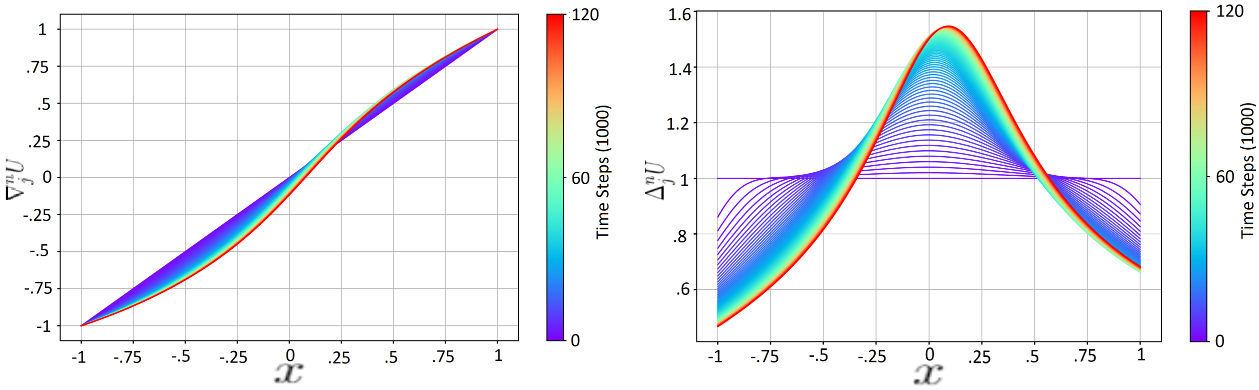
We now describe several computational examples. In examples 4.2.1, 4.3.1, 4.3.2, 4.4.1, 4.5.1 the domains $[A,B] = [C,D] = [-1,1]$. In 4.6.1 $[A,B] = [0,1]$ and $[C,D] = [-\pi,\pi]$. The graphs with color maps show the solution plotted every 1000 iterations. In the tables below t_{total} represents the value of t in Parabolic O-T at the final time-step, and CPU Time represents the computer run time in seconds. All were run on the CoCalc cloud computing environment with the academic research package.

4.2 'Nice' Functions

In testing our code we tried an initial variety of computationally nice functions for both f and g. These functions are bounded well above 0 (> 0.1), continuous, and did not change convexity more than twice. Such examples include logarithmic, exponential, linear, quadratic, constant, and concave cosine functions all modified to fit the conditions of (Parabolic O-T). The following is an example of numerical output using two functions from this set:

Example 4.2.1.

$$f(x) = \frac{\log(x+2)}{3\log(3)+2} + \frac{2}{3\log(3)+2}, \quad g(x) = \frac{1}{2}x^2 + \frac{1}{3}$$



(a) $\nabla_j^n U$ Over Time

(b) $\Delta_j^n U$ Over Time

Figure 1: Graphs of 4.2.1, 152.9s to Reach Tolerance
$\epsilon = 0.01, \quad \max_{j=0,\ldots,J} |T(x_j) - \nabla_j^n U| \leq 0.03$

Tolerance	Iterations	t_{total}	CPU Time (s)
0.1	814	0.0053	1.06
0.01	119880	0.7815	150
0.001	289020	1.88	360
0.0001	459807	2.997	573

Table 1: The computational time and iterations to reach tolerance. The sum of time steps is also given.

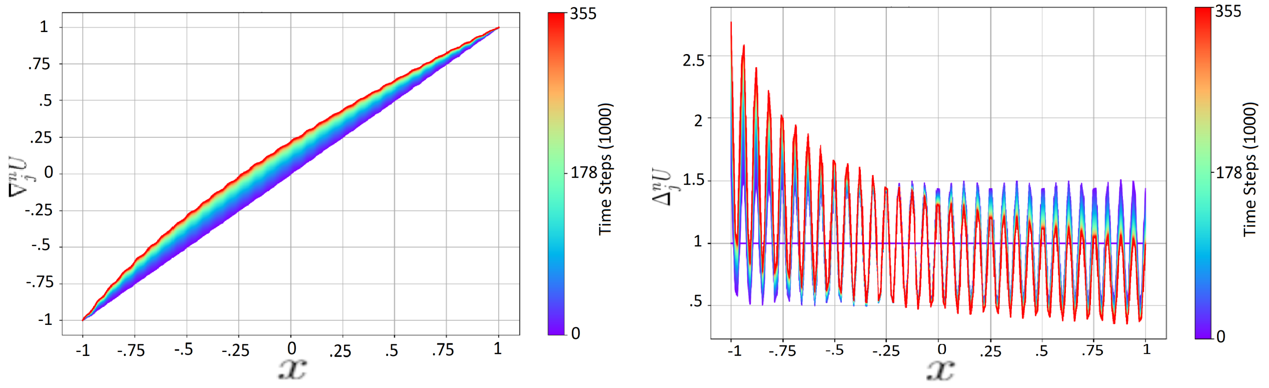
As expected the graph in Figure 1 shows exponential convergence to the optimal map.

4.3 High Frequency Functions

Example 4.3.1.

$$f(x) = \frac{50}{\sin(100) + 200}(\cos(100x) + 2), \quad g(x) = \frac{1}{4}(x + 2)$$

This case has frequent convexity changes of the initial mass function, $f(x)$. Testing this case allows us to know that our our code is able to handle more complex smooth cases.



(a) $\nabla_j^n U$ Over Time

(b) $\Delta_j^n U$ Over Time

Figure 2: Graphs of 4.3.1, 430s to Reach Tolerance
$\epsilon = 0.01, \quad \max_{j=0,\ldots,J} |T(x_j) - \nabla_j^n U| \leq 0.04$

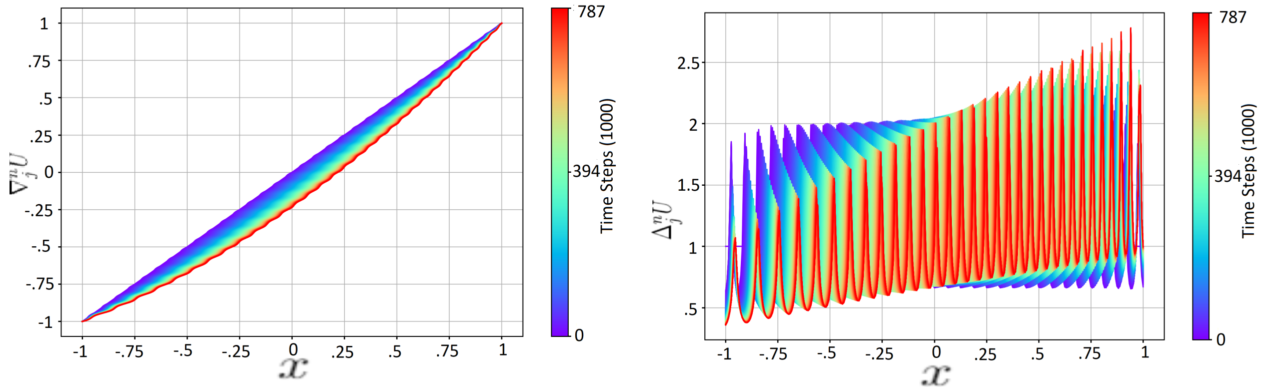
Tolerance	Iterations	t_{total}	CPU Time(s)
0.1	35877	0.1006	41.31
0.01	354534	0.9947	414
0.001	692928	1.944	817

Table 2: Numerics for high frequency function to quadratic.

Example 4.3.2.

$$f(x) = \frac{1}{4}(x + 2), \quad g(x) = \frac{50}{\sin(100) + 200}(\cos(100x) + 2)$$

This example switches $f(x)$ and $g(x)$ in Example 4.3.1. According to the optimal transport theory, the corresponding optimal map will be the inverse of the optimal map from Example 4.3.1. We also expected the runtime to be longer in this case, due to the high oscillation in the term $g(\nabla_j^n U)$ from (F-D).



(a) $\nabla_j^n U$ Over Time

(b) $\Delta_j^n U$ Over Time

Figure 3: Graphs of 4.3.1, 5195s to Reach Tolerance

Tolerance	Iterations	t_{total}	CPU Time(s)
0.1	83180	0.1001	596
0.01	786271	0.9454	5195
0.001	1254499	1.509	8635

Table 3: Numerics for quadratic function to high frequency.

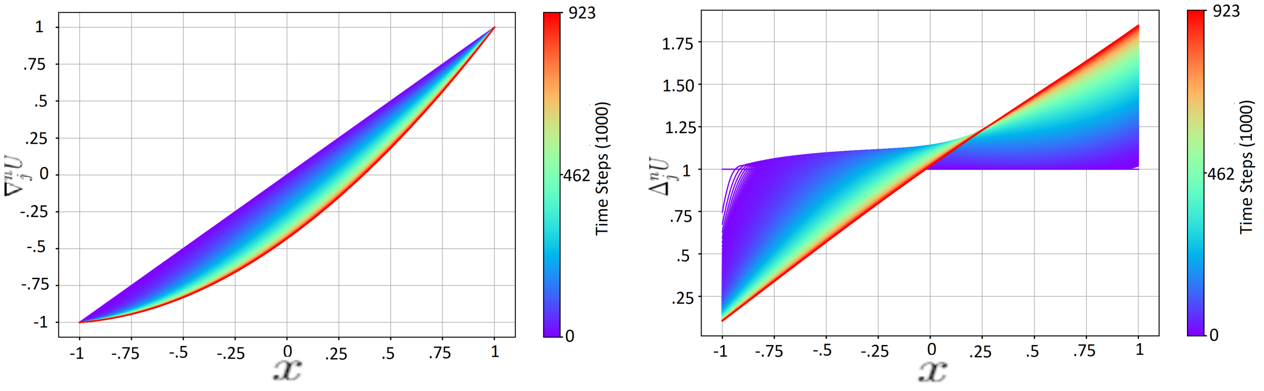
As expected, this example required more computational time and iterations to reach tolerance. Furthermore, we observe that the graph of $\nabla_j^n U$ in (Figure 3(a)) is the inverse of the graph of $\nabla_j^n U$ in (Figure 2(a)), which is predicted by the optimal transport theory. Due to the large differences in computational time, it would likely be more efficient to let the initial mass distribution function f be the more complicated one, and then computing the inverse of the optimal map between f and g if that is what one needs. However, it is worth keeping in mind the limitations of inverting a grid function, as the inverse is not necessarily defined on a well distributed set of grid points. This is illustrated in the next example.

4.4 Mapping Functions That Are Close To Zero

Example 4.4.1.

$$f(x) = \frac{9}{20}x + \frac{1}{2}, \quad g(x) = \frac{1}{2}$$

This is a case where the minimum of the initial mass distribution, $f(x)$, is close to 0. Although the theory implies that any smooth function bounded away from 0 will work for f and g, cases such as this cause the error to become large and the code to fail unless we incorporate the error conditions (2.0.1) into our code. After the conditions (2.0.1) were properly incorporated into our code, we found that the finite difference scheme is able to converge within tolerance to the real solution of (O-T).



(a) $\nabla_j^n U$ Over Time

(b) $\Delta_j^n U$ Over Time

Figure 4: Graphs of 4.4.1, 985.7s to Reach Tolerance
$\epsilon = 0.01, \quad \max_j |T(x_j) - \nabla_j^n U| \leq 0.02$

Tolerance	Iterations	t_{total}	CPU Time (s)
0.1	263683	0.3087	280
0.01	922803	1.0806	975
0.001	1568238	1.8364	1653
0.0001	2212321	2.59074	2326

Table 4: Numerics for Near Zero Function

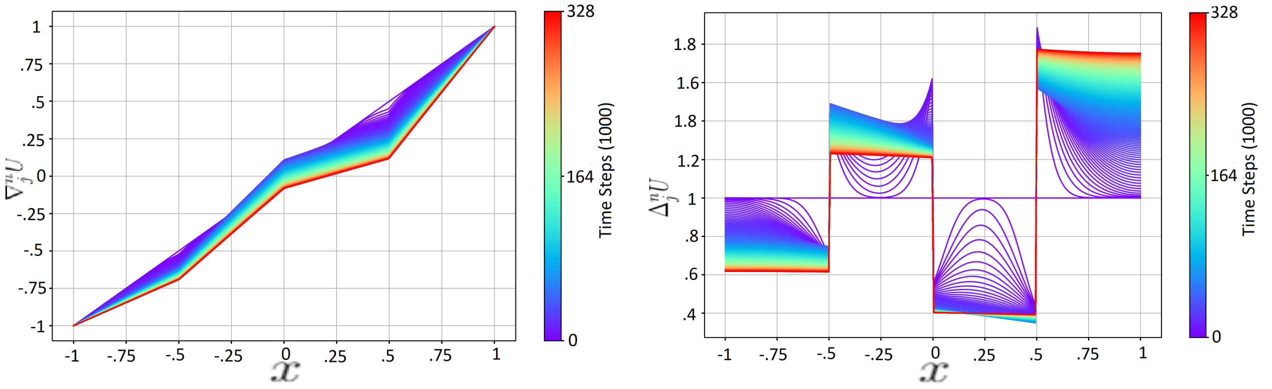
Now consider the more general case of (4.4.1); $f : [-1, 1] \to \mathbb{R}$ such that $f(x) = \beta x + \frac{1}{2}$. If β is very close to $\frac{1}{2}$ then $\min f$ is very close to zero. It follows from Theorem 2.1 that Δt gets very close to zero. In practice this has meant millions of iterations to reach convergence and a very slow program.

4.5 Mapping Piecewise Functions

In this section we discuss results involving mass distributions that are not guaranteed by (Parabolic O-T) to converge to the solution of (O-T). Yet, experimentally, with our finite difference scheme, we were able to show for some of these examples that the numerical solution can approach a desired tolerance and hence is close to the solution of (O-T). The following example uses a piecewise constant function to show this.

Example 4.5.1.

$$f(x) = \begin{cases} 0.3, & \mathit{if}\ x \leq -0.5 \\ 0.6, & \mathit{if}\ -0.5 < x \leq 0 \\ 0.2, & \mathit{if}\ 0 < x \leq 0.5 \\ 0.9, & \mathit{if}\ x > 0.5 \end{cases} \quad , \quad g(x) = \frac{1}{2}$$



(a) $\nabla_j^n U$ Over Time

(b) $\Delta_j^n U$ Over Time

Figure 5: Graphs of 4.5.1, 1211.4s to reach tolerance $\epsilon = 0.01, \quad \max_j |T(x_j) - \nabla_j^n U| \leq 0.021$

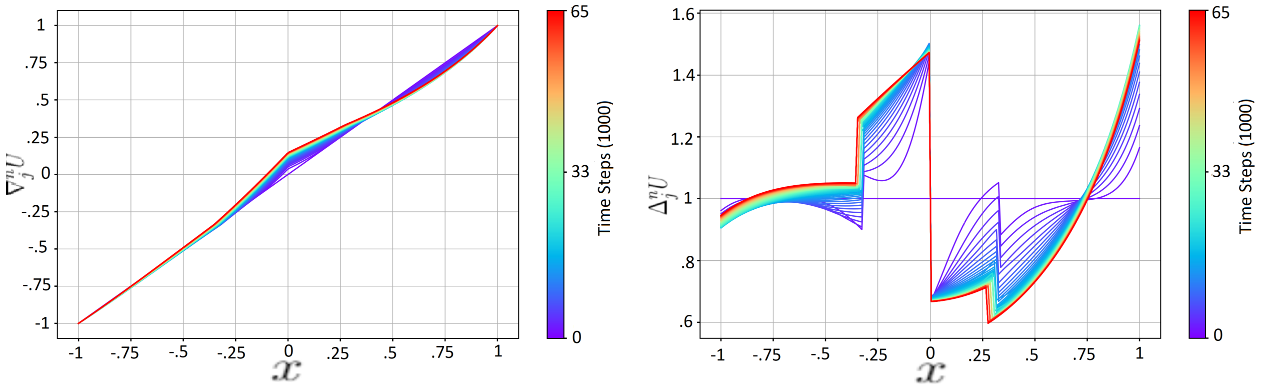
Tolerance	Iterations	t_{total}	CPU Time (s)
0.1	31325	0.1088	38.47
0.01	327070	1.1356	400
0.001	672739	2.3356	821

Table 5: Numerics for Piecewise Function

We were also motivated to test whether a piecewise function would be able to converge within tolerance when mapped to another piecewise function, with discontinuities at different points. Furthermore, we wanted to see the effects of functions that were not piecewise constant. This led us to test the following example:

Example 4.5.2.

$$f(x) = \begin{cases} \frac{1}{2}\log(x+2) + \frac{13}{12} - 2\log(2), & \mathit{if}\ x \leq 0 \\ \frac{1}{4}x^2 + \frac{1}{3}, & \mathit{if}\ x > 0 \end{cases}, \quad g(x) = \begin{cases} \frac{3}{10}x + \frac{7}{10}, & \mathit{if}\ x \leq \frac{-1}{3} \\ \frac{1}{2}, & \mathit{if}\ \frac{-1}{3} < x \leq \frac{1}{3} \\ \frac{-3}{10}x + \frac{7}{10}, & \mathit{if}\ \frac{1}{3} < x \end{cases}$$



(a) $\nabla_j^n U$ Over Time

(b) $\Delta_j^n U$ Over Time

Figure 6: Graphs of 4.5.2, 356s to reach tolerance

$$\epsilon = 0.01, \quad \max_j |T(x_j) - \nabla_j^n U| \le 0.025$$

			CPU Time (s)	
Tolerance	Iterations	t_{total}	Numerical	Analytical
0.1	1	7.616e-06	0.1283	0.0709
0.01	64445	0.4909	355	71.4
0.001	184774	1.408	1166	202

Table 6: Numerics for Piecewise Function

From this we were able to see that our scheme seems to also converge within tolerance even when given a piecewise function for both f and g. Although only the points of discontinuity in f are seen in $\nabla_j^n U$, we see all points of discontinuity in both f and g appear in the graph of $\Delta_j^n U$. This aligns with the expectation that $\Delta_j^n U$ approximates u''. We observed that piecewise functions tend to take more computational time to reach tolerance compared against smooth cases with similar upper and lower bounds on f and g. Even so, experimentally we found that closeness to zero had more of an effect on computational time.

Standard numerical integrators can have difficulties integrating discontinuous functions accurately and efficiently. Therefore, in cases involving piecewise functions it may be necessary to alter the error tolerance methods. One solution may be to implement function for the exact integral, calculated analytically if possible. Another would be to implement a specialized numerical integrator capable of handling piecewise functions. Both methods can also significantly improve computational time. See Table 6 for CPU time differences of the standard numerical integrator and exact analytical integrator for 4.5.2.

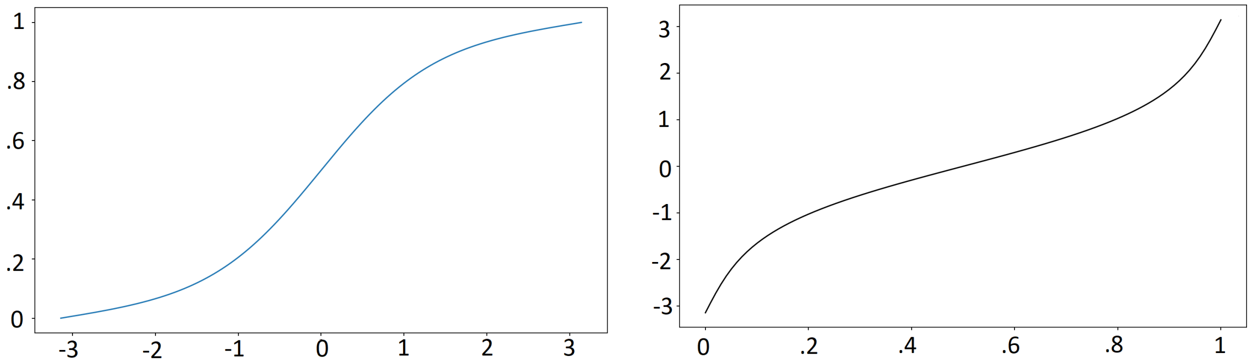
4.6 Quantile Example

Note that (1.1.3) provides a way to calculate the inverse of $G(x)$ if $F(x) = x$; that is, if we let $f \sim \text{Unif}[A, B]$. The numerical scheme (F-D) thus provides a way to compute the quantile function of any probability distribution that is supported on a bounded interval and stays away from zero.

Example 4.6.1.

$$f(x) = 1, \quad g(x) = \frac{e^{\kappa \cos(x)}}{2\pi \mathcal{I}_0(\kappa)}$$

The function g is an example of a von Mises distribution with $\mu = 0$ and $\kappa = 1$. Recall that $\mathcal{I}_0$ is the modified Bessel function of the first kind and of order zero. In this case $f : [0, 1] \to \mathbb{R}$ and $g : [-\pi, \pi] \to \mathbb{R}$ with $u_0 = \pi(x^2 - x)$.



(a) CDF of von Mises Distribution

(b) Quantile of von Mises Distribution

Figure 7: Graphs 4.6.1, 801.8s to Reach Tolerance
$\epsilon = 0.001, \quad \max_{j=0,\ldots,J} |T(x_j) - \nabla_j^n U| \leq 0.022$

The usual method for numerically approximating a quantile function involves numerically integrating the probability density function, g to get an approximation of the cumulatively distribution function G, then inverting the x and y coordinates of G. Inverting a grid function can result in a non-uniform grid for the numerical inverse, and if G has large derivative, then the grid points of the domain of G^{-1} will be concentrated along the points where the derivative of G is large. Using F-D to compute an approximation of G^{-1} does not involve inverting a grid function, and therefore provides better resolution, though it is computationally much slower.

Conclusion and Outlook

We have shown error bounds on the finite difference scheme for the 1-D parabolic optimal transport problem and provided relevant numerical examples. If the error conditions (2.0.1) are met, we have seen empirically that $\Delta_j^n U$ stays positive when calculating the optimal map, allowing for the application of Theorem 2.1. We hope to further investigate this condition and prove that $\Delta_j^n U$ always stays positive in future work.

The error given by (2.0.2) provides a way to quantify the error at any given time step n, but it is not guaranteed to stay small as $n \to \infty$. In practice, this is not detrimental to the efficacy of (F-D) as the scheme is only run for a finite number of time steps until it is within tolerance. We also hope to be able to bound the error in (F-D) for all n in future work.

We have a first order term in the error bound at the boundary, which adversely impacts our schemes accuracy in approximating (Parabolic O-T). This does not seem to have much impact on the accuracy our scheme in approximating the optimal map as it is empirically it is still able to get within tolerance. In the future we hope to investigate how to replace this first order error term with a second order one in order to create a scheme that can more accurately approximate (Parabolic O-T).

Additionally our research was only carried out for one spatial dimension. Further research is necessary to devise robust numerical methods for the optimal transport problem in two dimensions and higher. Additional work needs to be done to understand why piecewise functions used in the (Parabolic O-T) are able to converge to (O-T) and proven mathematically.

Acknowledgements

This research was supported by the following grants; NSA Award No. H98230-20-1-0006 and NSF Award No. 1852066. We would like to thank Professor Robert Bell and Michigan State University for organizing SURIEM, and we would like to thank our research mentors Farhan Abedin and Jun Kitagawa for their guidance.

References

[1] Farhan Abedin and Jun Kitagawa. Exponential convergence of parabolic optimal transport on bounded domains. *Anal. PDE 13 (2020), no. 7, 2183–2204.*

[2] Jean-David Benamou, Brittany D. Froese, and Adam M. Oberman. Two numerical methods for the elliptic Monge-Ampère equation. *M2AN Math. Model. Numer. Anal.*, 44(4):737–758, 2010.

[3] Jean-David Benamou, Brittany D. Froese, and Adam M. Oberman. Numerical solution of the optimal transportation problem using the Monge-Ampère equation. *J. Comput. Phys.*, 260:107–126, 2014.

[4] Abigail Brauer, Megan Krawick, and Manuel Santana. Numerical analysis of the 1-d parabolic optimal transport problem, 2020. arXiv:2008.12815.

[5] Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Comm. Pure Appl. Math.*, 44(4):375–417, 1991.

[6] Jun Kitagawa. A parabolic flow toward solutions of the optimal transportation problem on domains with boundary. *J. Reine Angew. Math.*, 672:127–160, 2012.

[7] Gaspard Monge. Mémoire sur la the orie des déblais et de remblais. *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année,*, page 666–704, 1781.

[8] Michael Neilan, Abner J. Salgado, and Wujun Zhang. The Monge-Ampère equation. *Handbook of Numerical Analysis, vol 21:105-219, 2020.*

[9] Eitan Tadmor. A review of numerical methods for nonlinear partial differential equations. *Bull. Amer. Math. Soc. (N.S.)*, 49(4):507–554, 2012.