

SOLUTION OF OPTIMAL TRANSPORTATION PROBLEMS USING A MULTIGRID LINEAR PROGRAMMING APPROACH*

Adam M. Oberman

Department of Mathematics, McGill University, Montreal, Canada

Email: adam.oberman@mcgill.ca

Yuanlong Ruan

Department of Mathematics, Beihang University, Beijing 100191, China

Email: ruanylpku@gmail.com

Abstract

We compute and visualize solutions to the Optimal Transportation (OT) problem for a wide class of cost functions. The standard linear programming (LP) discretization of the continuous problem becomes intractable for moderate grid sizes. A grid refinement method results in a linear cost algorithm. Weak convergence of solutions is established and barycentric projection of transference plans is used to improve the accuracy of solutions. Optimal maps between nonconvex domains, partial OT free boundaries, and high accuracy barycenters are presented.

Mathematics subject classification: 49M99, 65K15, 90C05.

Key words: Optimal Transportation, Linear Programming, Monge-Kantorovich, Barycenter.

1. Introduction

There has recently been a great deal of interest in Computational Optimal Transportation (OT) with a number of new, effective numerical methods being proposed, including Entropic Regularization [14, 15], extensions of the classical Benamou-Brenier methodology [4] and semi-discrete methods [19, 20]. Computational OT is being applied to Machine Learning [10], Inverse Problems [21, 30], Data Assimilation [27]. Extension of the numerical methods have lead to methods for Wasserstein Gradient Flows [6] and Mean Field Games [8]. These methods have advanced far beyond the early methods which include the classical Benamou-Brenier method [3], Haker and Tannenbaum's method [18] and more recent PDE based methods [9]. See also the book [22].

The applications of Optimal Transportation are varied, and the numerical methodology has advanced considerably. In this article, we focus on geometric applications of the OT problem, and visualization of classical problems. We achieve accurate results, using variable reduction for the linear programming method, but without the rigorous verification achieved by Schmitzer [24]. However our method is much simpler to implement: it can be implemented using vector operations in Matlab. Moreover, at least in the quadratic cost case, our heuristic is very close to the rigorous block data checking of Schmitzer. We implement a practical Linear Programming (LP) method for solving Optimal Transportation problems. The computational cost is linear in terms of solution time and memory requirements (see Table 4.1). We compute solutions

* Received September 19, 2017 / Revised version received April 26, 2019 / Accepted July 9, 2019 /
Published online September 13, 2019 /

with problem sizes going up to half a million variables on a laptop computer using academic or commercial Linear Programming optimization software. Available parallel LP solvers allow the method to scale to even larger problem sizes.

Linear Programming solutions have very low accuracy compared to PDE methods. This is because the Kantorovich solutions are much weaker: the approximate solutions are plans (see Figure 1.1). This reduces the accuracy of solutions, and makes visualization difficult. Barycentric projection is used to recover the approximate map and improves the accuracy of solutions, leading to accuracy comparable to the accuracy of the PDE method.

A weak convergence proof is established using available stability results. Barycentric projection commutes with weak convergence, in the case of convex costs, so the convergence proof remains valid.

We use a heuristic grid refinement method, where the support of the solution on a coarse grid is used to infer the support of the solution on a refined grid. A rigorous approach to grid refinement is available: the work of Schmitzer [23, 24] ensures that at each refinement, the sparse solution of the linear program is optimal for the full problem. However this method introduces significant complexity to the code.

The efficiency and accuracy of the method reveals solution features not otherwise available, including optimal maps between nonconvex sets, and for non-quadratic cost functions. The method is generalized to other problems, allowing for the computation of accurate free boundaries in partial optimal transportation, and high resolution barycenters for shapes.

1.1. Related numerical work

Entropic regularization methods modify the optimization problem by adding a small multiple of an entropy term [14, 15] to the objective (or cost) function. The regularized problem can be solved by a Bregman iterative solution method [5]. The recent article [26] applied the method to large scale geometric problems.

Multiscale solvers have previously been used to improve solver performance. Merigot solved a sequence of OT problems where the target is a sum of Diracs [20]. Schmitzer [23, 24] used a grid refinement procedure which was applied to both LP and combinatorial optimization solvers. A one step grid refinement was used in [13] to find the support of the barycenter. A grid refinement procedure was used in [7].

The early Benamou-Brenier formulation leads to a fluid mechanics solver, by adding a synthetic time variable to the problem [3], adding one dimension to the problem. The Monge-Ampère Partial Differential formulation was recently used to solve the OT problem using a convergent finite difference method [9]. This method is for quadratic cost, and places regularity requirements on the densities, one of which must have convex support. Our method has comparable accuracy to the PDE approach, see §4.3.

1.2. Background on the Optimal Transportation problem

The Monge formulation of the Optimal Transportation problem which seeks optimal *maps* while the Kantorovich formulation seeks optimal *transference plans*. Transference plans are a weaker notion of solution which can be computed using linear programming.

Given two probability measures, μ, ν with bounded supports $X, Y \subset \mathbb{R}^d$, respectively and the cost function $c(x, y) : X \times Y \rightarrow \mathbb{R}$. The goal is to rearrange one measure into the other,

while minimizing the cost of mapping x to y , weighted by the amount of mass transported. There are two notions of rearrangement.

In the Monge formulation, we consider the class of measurable, one-to-one mappings $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ which rearrange μ into ν . For general measures, the rearrangement condition is given by

$$\nu[B] = \mu[T^{-1}(B)], \quad \text{for any measurable set } B \subset Y. \tag{1.1}$$

We write $\nu = T\#\mu$, and say that T transports μ onto ν . Monge’s problem is to minimize the total work corresponding to a map T over the set of all measurable maps T which transport μ onto ν .

$$\text{Minimize } I[T] = \int_X c(x, T(x))d\mu(x) \tag{1.2}$$

Subject to: $T\#\mu = \nu$.

The Kantorovich formulation recasts the Optimal Transportation problem as an infinite dimensional Linear Program. A transference plan generalizes a mapping, in order to allow for mass from a point x to be split into multiple parts. A transference plan, π , is a Borel probability measure on the product space $X \times Y$, whose marginals are μ and ν . This means that

$$\pi[A \times Y] = \mu(A), \quad \pi[X \times B] = \nu(B)$$

for all measurable subsets A of X and B of Y . The set of transference plans is written

$$\Pi(\mu, \nu) = \{ \pi \in \mathcal{P}(X \times Y) \mid \text{marginals of } \pi \text{ are } \mu, \nu \}. \tag{1.3}$$

Kantorovich’s formulation of the optimal transportation problem is

$$\text{Minimize } I[\pi] = \int_{X \times Y} c(x, y)d\pi(x, y), \quad \text{for } \pi \in \Pi(\mu, \nu). \tag{KLP}$$

Since the constraints are linear, and the cost function is also linear, this is an infinite dimensional linear program. Under broad conditions, this problem has a minimal value, which is called the *optimal transportation cost* between μ and ν . However, in general, there may be more than one optimal transference plan, we refer to [28] for more details.

Remark 1.1 (When an optimal plan is a map). For many costs, the transference plan solution from (KLP) is given by a map. This is the case for $c(x, y) = h(|x - y|)$, where $|\cdot|$ is the Euclidean norm, when h is strictly convex and with superlinear growth. In particular, $c(x, y) = |x - y|^p$ for $p > 1$, which includes the important case $p = 2$.

More generally, the *twist* condition, which requires that $g(y) = \nabla_x c(x, y)$ must be an injective function of y , ensures that the optimal plan is a map, see [17] [12], or [29, Chapter 10]. For other cost functions, an optimal transference plan need not be a map. In particular, this is the case for the Monge cost $c(x, y) = |x - y|$.

Example 1.1 (The discrete transportation problem). A special case of (KLP) is the transportation Linear Program, see Fig. 1.1(b). Let δ_x denote the Dirac mass at x . Given discrete probability measures

$$\mu = \sum_{i=1}^n \mu_i \delta_{x_i}, \quad \nu = \sum_{j=1}^m \nu_j \delta_{y_j}, \quad \mu_i, \nu_j \geq 0, \quad \sum_{i=1}^n \mu_i = \sum_{j=1}^m \nu_j = 1.$$

The cost function is given by a non-negative $n \times m$ matrix, $c = (c_{ij})$. A discrete transference plan, π , is a non-negative $n \times m$ matrix whose marginals are μ and ν . The set of transference plans (1.3) becomes

$$\Pi(\mu, \nu) = \left\{ \pi = (\pi_{ij}) \mid \sum_{j=1}^m \pi_{ij} = \mu_i, \quad \sum_{i=1}^n \pi_{ij} = \nu_j, \quad \pi_{ij} \geq 0 \right\}. \tag{1.4}$$

The transportation linear program is given by

$$\text{Minimize } I[\pi] = \sum_{i=1}^n \sum_{j=1}^m c_{ij} \pi_{ij}, \quad \text{for } \pi \in \Pi(\mu, \nu). \tag{LP}$$

The problem (LP) is easily written in the standard form for a linear program.

Remark 1.2. Notice that the number of variables used to represent the densities in (LP) is $n + m$, and the number of variables for the plan π is nm . So the size of the linear programming problem grows *quadratically* in the number of variables.

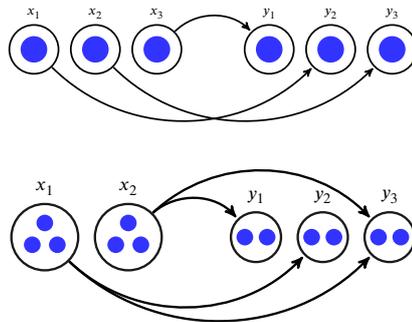


Fig. 1.1. (a) Illustration of a mapping from three points to three points. (b) An optimal transference plan can split points.

2. Discretization and Convergence

In this section we perform the discretization which reduces (KLP) to a finite dimensional Linear Program (LP). We prove that the solutions of the discrete problem converge weakly to the solution of (KLP) as the grid resolution parameter $h \rightarrow 0$. Even when the limiting solution is a map, the weak solution of (LP) can be a plan. By using barycentric projection of the transference plan, we recover the map (without affecting weak convergence), resulting in improved accuracy.

2.1. Discretization: finite dimensional linear programming

For clarity and simplicity, we impose a Cartesian grid with uniform spacing h on the domains X and Y . The initial grids, X^h , and Y^h are given by a uniform Cartesian grid with spacing h intersected with the support set,

$$X^h = h\mathbb{Z} \cap X, \quad Y^h = h\mathbb{Z} \cap Y.$$

We enumerate the grid points, (which could also be referred to as *quadrature* points),

$$X^h = \{x_1, \dots, x_n\}, \quad Y^h = \{y_1, \dots, y_m\}.$$

Each x_i is the center of a hypercube of width h ,

$$R_h(x_i) = \left\{ x \in \mathbb{R}^d \mid \|x - x_i\|_\infty \leq \frac{h}{2} \right\}.$$

Define the approximate measures μ^h, ν^h , to be a weighted sum of Dirac masses whose weights correspond to the integral of the measures over the hypercubes of width h centred at x_i .

$$\begin{aligned} \mu^h &= \sum_{i=1}^n \mu_i^h \delta_{x_i}, & \mu_i^h &= \mu(R_h(x_i)), \\ \nu^h &= \sum_{i=1}^m \nu_i^h \delta_{y_i}, & \nu_i^h &= \nu(R_h(y_i)). \end{aligned} \tag{2.1}$$

These values can be computed by quadrature on the grid. The discrete cost function is given by

$$c_{ij}^h = c(x_i, y_j). \tag{2.2}$$

With these definitions, the infinite dimensional optimal transportation problem (KLP) is reduced to the finite dimensional linear programming problem (LP). We record it in the definition which follows.

Definition 2.1. Define (LP) at grid resolution h to mean the costs and measures are given by c_{ij}^h, μ^h, ν^h using (2.1) and (2.2). Write $\pi^h = \pi_{ij}^h$ for an optimal solution of (LP) at resolution h . From π_{ij}^h , we recover the approximation to the optimal transference plan π by the imbedding

$$\Pi^h = \sum_{i=1}^n \sum_{j=1}^m \pi_{ij}^h \delta_{(x_i, y_j)}. \tag{2.3}$$

2.2. Convergence of the linear programming approximation

In this subsection we apply a fundamental stability result for solutions of the optimal transportation problem, along with the consistency of the approximation to obtain weak convergence for the solutions of (LP) to the solution of (KLP) as $h \rightarrow 0$.

Given μ, ν and $c(x, y)$ as described in §1.2, consider a sequence $h_k \rightarrow 0$. The following theorem is a special case of [29, Theorem 5.20].

Theorem 2.1 (Stability of optimal transportation). Assume X and Y are compact, and that $c(x, y)$ is continuous. Let μ^{h_k} and ν^{h_k} be a sequence of measures that converge weakly to μ and ν , respectively. For each k , let π^{h_k} be an optimal transference plan between μ^{h_k} and ν^{h_k} , and let Π^{h_k} be the imbedding given by (2.3). Then, up to extraction of a subsequence,

$$\Pi^{h_k} \text{ converges weakly to } \pi$$

where π is an optimal transference plan between μ and ν .

Remark 2.1. Note that using this discretization, (2.2), the cost function is given by $c(x, y)$ the same for all values of h . However, the cost vector c_{ij}^h in the linear program depends on h . We remove the h superscript when the definition is clear from the context. So there is no need for a stability result with involving perturbations of the cost function.

Theorem 2.2 (Convergence of Linear Programming Solutions). *Suppose that X and Y are compact, and that $c(x, y)$ is continuous. Let π^h be a solution of (LP) given by the discretization (2.1)–(2.3). Then, up to extraction of a subsequence, π^h converges weakly to an optimal transference plan solution of (KLP).*

Proof. The measures given by (2.1) are approximations by weighted Diracs masses over intervals whose volumes go to zero as $h \rightarrow 0$. So clearly these measures converge weakly to the limits μ and ν .

The discrete cost function given by (2.2) can be extended to a continuous cost function defined on $X \times Y$ which converges uniformly to $c(x, y)$. This can be accomplished, for example, by defining

$$c(x_i, y_j) = c_{ij}^h, \quad i = 1, \dots, n, \quad j = 1, \dots, m$$

extend it continuously to $X \times Y$, by multilinear interpolation.

The solution of (LP) results in an optimal transference plan π^h given by (2.3). Apply the stability theorem above to obtain the desired weak convergence result.

2.3. Recovering the map from the plan using barycentric projection

Even when the limiting transference plan π is a map, the approximating discrete marginal π^h will not be a map, in general. However an approximation to the optimal map can be recovered from the transference plan using *barycentric projection*.

The transference plan obtained from the linear programming solution π^h is given by the imbedding (2.3) $\pi^h = \sum \pi_{ij}^h \delta_{(x_i, y_j)}$. Thus the mass μ_i at x_i is transported to multiple points y_j with weights given by $\mu_i x_i \mapsto \sum_{k=1}^m \pi_{ik} \delta_{y_k}$.

Definition 2.2 (Barycentric projection of transference plan). *Define for each i with $\mu_i > 0$, \bar{y}_i to be the Euclidean barycenter of the points (y_1, \dots, y_m) with weights $(\pi_{i1}, \dots, \pi_{im})$,*

$$\bar{y}_i = \frac{\sum_{k=1}^m \pi_{ik} y_k}{\sum_{k=1}^m \pi_{ik}}.$$

Then define the barycentric projection of the transference plan π^h by

$$\bar{\pi}^h = \sum_{i=1}^n \mu_i^h \delta_{(x_i, \bar{y}_i)}.$$

See Fig. 2.1.

See [2, Definition 5.4.2] for the continuous case. Then $\bar{\pi}^h$ is a (discrete) *map* which pushes forward μ^h to (an approximation) of ν^h .

Remark 2.2. Generally, \bar{y}_i no longer belongs to the set $\{y_1, \dots, y_m\}$. Correspondingly, while the first marginal of $\bar{\pi}^h$ is equal to μ^h , the second marginal of $\bar{\pi}^h$ is generally different from ν^h . However, it is still a weak approximation of ν , which converges weakly to ν as $h \rightarrow 0$.

Barycentric projection is discussed in [2]. In particular, by Theorem 5.4.4 and by Lemma 12.2.3, we can conclude that, for convex costs, the barycentric projection of the approximations converges to the barycentric projection of the limiting transference plan. In particular, when the unique limit is a map π , then $\bar{\pi}^h$ converges to π .

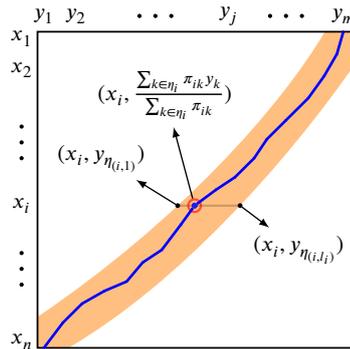


Fig. 2.1. Illustration of the transference plan $\pi = \pi_{ij}^h$ and the barycentric projection $\bar{\pi}$. The coloured band represents the support of the transference plan. Points in the same row are bundled into one as indicated by the small circle. The curve in the middle represents the resulting map.

3. Sparsity and a Grid Refinement Procedure

The full linear program (LP) which approximates (KLP) is too costly to solve for large problems. In the case where the solution from (KLP) is given by a map, we can expect that the discrete solution of (LP) will be *sparse*, provided h is small enough. By estimating the support of the transference plan, π^h , the number of variables in (LP) is reduced from quadratic to linear dependence on the inputs.

Estimating the support in this manner forms the basis of the grid refinement procedure. This process is iterated, leading to a multiple step grid refinement procedure.

By considering a sequence of linear programs parameterized by h we can find the support of the solution, π^h , from the support of the solution corresponding to a smaller sized problem. Then we can solve the reduced size linear program, (LPR) below, with the expectation that the solution of the sparse problem is the same as that of the full problem. This will be the case if the previously estimated support is exact.

Remark 3.1. It is possible that the grid refinement introduces errors. For the fairly smooth measures we used, this was not the case.

In the general case, for example, with noisy cost functions, an alternative is to use the method of Schmitzer [23] (see also [24]), which ensures global optimality of the solution to the sparse subproblem by using locally adapted neighborhood sizes.

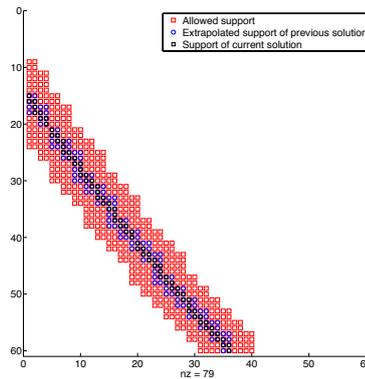


Fig. 3.1. Illustration of the multigrid solution procedure. The nonzero elements of the matrix are plotted. The black boxes indicate the support of the current solution. The grid is refined, and the current support is extended to the refined grid, illustrated by the blue boxes. In the next iteration of the algorithm, the possible support of the solution is extended to neighbors, as illustrated by the red boxes.

3.1. The sparse linear program

Let $\pi = \pi_{ij}$ be the solution of (LP), and let K_0 be the basis set, the indices of the nonzero entries of π . If $K_0 \subset K$, for a known set, K , then we can recover π by solving the reduced linear program,

$$\begin{aligned} & \text{Minimize} && \sum_{(i,j) \in K} c_{ij} \pi_{ij}, \\ & \text{Subject to:} && \begin{cases} \sum_{\{j|(i,j) \in K\}} \pi_{ij} = \mu_i, & i \in [1, \dots, n], \\ \sum_{\{i|(i,j) \in K\}} \pi_{ij} = \nu_j, & j \in [1, \dots, m], \\ \pi_{ij} \geq 0, & (i, j) \in K. \end{cases} \end{aligned} \tag{LPR}$$

Remark 3.2. Note that, in contrast to (LP), the number of variables in (LPR) is $|K|$, and the number of constraints is $|K| + n + m$. In practice, $|K|$ is a small multiple of $n + m$. So the size of the reduced linear programming problem (LPR) grows linearly in the number of variables used to represent the measures.

3.2. Multigrid solution procedure

The multigrid solution procedure is described here. Refer to Fig. 3.1, which illustrates the process for a one dimensional example.

The full linear program (LP), with $c_{ij} = c_{ij}^h$ and $\mu = \mu^h, \nu = \nu^h$ as given by the quadrature rules (2.1) and (2.2), is solved on a coarse grid, $X^h \times Y^h$.

1. From the discrete solution, π_{ij}^h , recover the spatial support set,

$$S_h = \{R_h(x_i^h) \times R_h(y_j^h) \mid \pi_{ij}^h > 0\}, \quad S_h \subset X^h \times Y^h.$$

Grow the spatial support set, by including neighbours (in space),

$$\bar{S}_h = \{\text{neighbours } S_h\}.$$

Extract the corresponding indices

$$\bar{K}_h = \{\text{indices of center points in } \bar{S}_h\}.$$

2. Refine the grids in each domain by a factor of two, labelling the new grids $X^{h/2}$, $Y^{h/2}$. In each coordinate of $x = (x_1, \dots, x_d)$, the interval $[x_i - h, x_i + h]$ is halved, to become $[x_i - h, x_i]$, $[x_i, x_i + h]$, and the two new points are generated at the midpoints of the interval. Thus, the hypercube $R_h(x)$ is divided evenly into 2^d hypercubes, and the grid point x generates 2^d new points on the refined grid, each at the centres of the smaller hypercubes.
3. The allowed spatial support is interpolated onto the finer grid $S_{h/2} = \bar{S}_h$ and the corresponding indices are extracted

$$K_{h/2} = \{\text{child indices of } \bar{K}_h\}.$$

4. The sparsely supported linear programming problem (LPR) is solved with indices $K_{h/2}$ and with $c^{h/2}$ and $\mu^{h/2}, \nu^{h/2}$ given by the quadrature rules (2.1) and (2.2).
5. Repeat starting at (1), until a fine enough solution is computed.

Remark 3.3. Depending on the coarseness of the approximations, Step (1) can be skipped, or repeated, so that neighbours of neighbours are included. The computational cost of adding additional support indices is not significant. Step (3) is geometrically simple; algorithmically, an indexing formula is used to determine the child indices. In step (4) (LPR) has $|K_{h/2}|$ variables, which is on the order of $2^d(n + m)$.

4. Numerical Results

4.1. Performance

We used MATLAB to generate the full and sparse LP problems and to call the LP solver. We tried several LP solvers, and found that several commercial products (MOSEK, Gurobi, CPLEX) performed equally well. We called these products from CVX, and also called them directly from MATLAB.

For benchmarking performance we used the example of transporting uniform densities from a square to diamond, see Fig. 4.2. The LPs were generated in MATLAB using Gurobi as a solver.

Table 4.1 compares computation time and memory usage of the multigrid LP solver with that of the full LP solver. Computation time and memory usage grow linearly with the number of unknowns, $2N^2$, for the Linear Program, as shown in Fig. 4.1. We used a PC laptop (i3 1.9GHz CPU with 12GB RAM). For comparison, we also ran the example with $N = 512$ on a 2015 MacBook Air (2.2GHz intel i7 CPU with 8GB RAM) and the run time decreased by about 30%.

Table 4.1: Comparison of run time and memory usage for the multigrid LP (MGLP), the full LP (FLP). Precision for the LP solvers are both 10^{-8} . Memory usage as reported by Gurobi.

N (grid)	$2N^2$	MGLP		FLP	
		CPU	Memory	CPU	Memory
32	2048	0.9	14	14.0	400
48	4608	1.6	40	95.6	2000
64	8192	3.2	80	527.1	7000
96	18432	7.0	160	*	*
128	32768	13.5	300		
192	73728	35.3	700		
256	131072	58.9	1100		
384	294912	165.5	2500		
512	524288	287.6	4000		

For the full solver, the memory usage with $N = 64$ or 8192 variables was 7 GB and the computation time was nearly 527 seconds. In comparison, the multigrid method was an improvement in both time and memory of two orders of magnitude. The largest sized problem we computed using the multiscale solver corresponded to $N = 512$, or about half a million variables. Compared to the largest problem for the full solver, this is an increase of problem size two orders of magnitude.

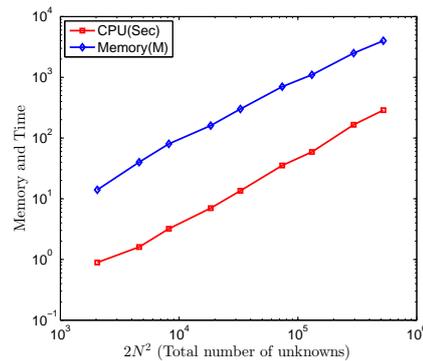


Fig. 4.1. CPU time and memory usage of the multiscale linear programming solver. Both memory usage and computation time scale linearly with problem size.

4.2. Smoothing the density

The target density obtained by the push forward of the barycentric projection of π^h is still given as a weighted sum of Dirac masses. We can improve the accuracy of the result by smoothing the density, in other words by simply replacing each Dirac mass by a sum of Gaussians with standard deviation on the same scale as h . We took values between 5 and 9 multiples of h . More general methods for Kernel Density Estimation in the context of statistical inference are discussed in [25].

Fig. 4.2 is a comparison of optimal transport map before and after applying Gaussian filter using the map between uniform densities on a square and a diamond, discussed further in §4.4. The second part of the figure shows smoothing of a computed barycenter, see §4.6.

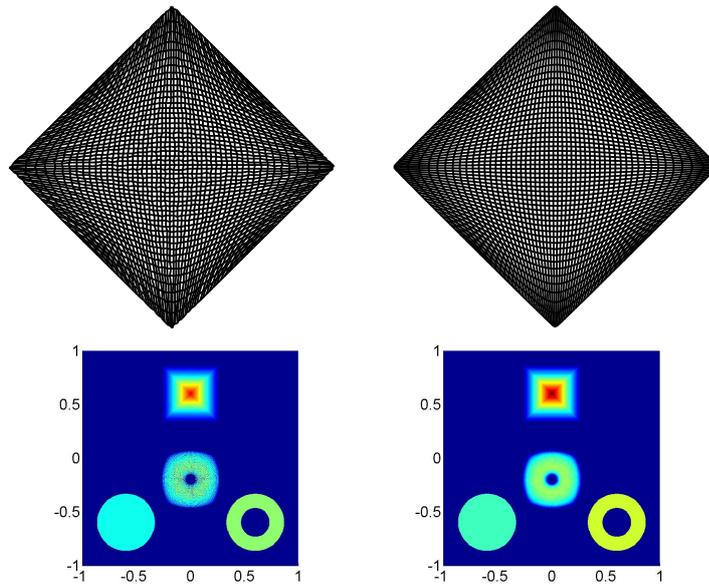


Fig. 4.2. Top: Optimal transport map from square to diamond recovered by barycentric projection. Left: Before smoothing. Right: After smoothing. Bottom: Barycenter of a square and 2 Cylinders. Left: Before smoothing. Right: After smoothing.

4.3. Accuracy of the solutions

In this section we report on the accuracy of solutions in terms of the grid spacing h , by considering exact solutions to the Monge-Ampère PDE. These examples were used in [9]. The first example is for a variable density. The second example is the well-known example of Caffarelli which splits a circle. In the case the mapping consists of two translations. In these examples, both maximum and L_2 error decrease approximately linearly with grid size which is comparable with the results in [9]. The solutions are plotted in Fig. 4.3.

Example 4.1. Transportation between two squares of the same size. The target has constant density, while the source square (Fig. 4.3) has a density given by

$$f(x, y) = 1 + 4(q''(x)q(y) + q(x)q''(y)) + 16(q(x)q(y)q''(x)q''(y) - q'(x)^2q'(y)^2),$$

where

$$q(z) = \left(-\frac{1}{8\pi}z^2 + \frac{1}{256\pi^3} + \frac{1}{32\pi}\right)\cos(8\pi z) + \frac{1}{32\pi^2}z\sin(8\pi z).$$

The optimal transportation map is given by

$$\begin{cases} u_x(x, y) = x + 4q'(x)q(y), \\ u_y(x, y) = y + 4q(x)q'(y). \end{cases}$$

The accuracy of the solution is presented in Table 4.2.

Example 4.2. Fig. 4.3 shows a circle is split into two half circles. The potential is given by

$$V(x_1, x_2) = \frac{1}{2}(|x_1| + x_1^2 + x_2^2).$$

The accuracy of the solution is presented in Table 4.2.

Table 4.2: Accuracy of the solution for Examples 4.1 and 4.2.

N (gridsize)	32	64	128	256	512
Max error Ex 4.1	0.00721	0.00892	0.00689	0.00241	0.00148
L_2 error Ex 4.1	0.00385	0.00379	0.00257	0.00103	0.00047
Max error Ex 4.2	0.0625	0.03125	0.01563	0.00781	0.00391
L_2 error Ex 4.2	0.01211	0.00302	0.00148	0.00050	0.00018

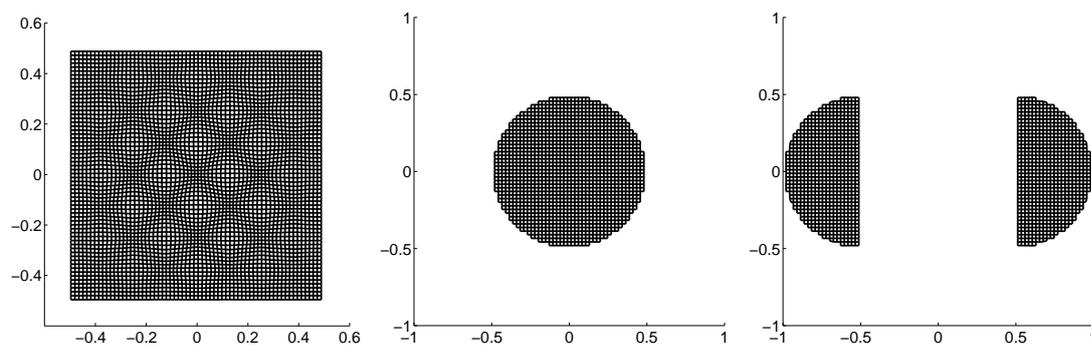


Fig. 4.3. Optimal transport map recovered by barycentric projection. Left: Target of Example 4.1. Middle: Source of Example 4.2. Right: Target of Example 4.2.

4.4. Computed solutions of Optimal Transportation Problem

It is difficult to visualize a map from the plane to the plane. Our strategy for visualizing maps begins by taking the source domain to be a square with uniform density, and by pushing forward the image of the grid lines onto the target domain. Subsequently, we consider more complicated source domains, still mapping horizontal and vertical lines from the target. We also label points in the source as well as their images for clarification.

We begin by computing examples of the Optimal Transportation problem with quadratic cost $c(x, y) = |x - y|^2$. In special cases, the solution is given by solving the Monge-Ampère Partial Differential Equation. We refer to [9] for computations in this setting. The first examples are uniform densities. Example 4.3 shows the optimal map from a square to a non-convex target. Example 4.4 shows the optimal map from a square to a diamond. Example 4.5 shows a map between two non-convex domains.

In Example 4.6, we consider costs of the form $c(x, y) = |x - y|^p$, and consider a piecewise constant density on the source rectangle.

Example 4.3. First consider the OT problem with μ the uniform density on a square and ν to the uniform density on the “Pac-Man”, a circle with a section of $\pi/2$ radians removed. The mapping is presented in Fig. 4.4. The map is discontinuous on a line segment market OG on the square, which is mapped to two lines in the target, from the center O to halfway down the radial segments. The map is continuous on the rest of the domain. Near the points E, F in the source, vertical lines of the form $x = c$ are mapped to corners, lines of the form $x = |y| + c$.

Example 4.4. Fig. 4.2 shows the optimal map from μ the uniform density on a square, X , to μ , the uniform density on a diamond. The source square is not plotted, but it can be seen in Fig. 4.4. The map shares the symmetries of the target. But while the central horizontal and vertical lines are mapped to straight lines, lines on the boundary of the square bend to the

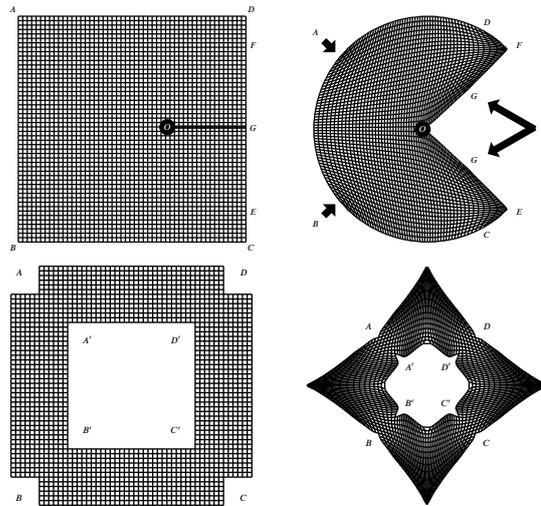


Fig. 4.4. Top: Optimal map from the square to “Pac-Man”. Bottom: Optimal map between two non-convex shapes.

corners. Squares near the center of X are mapped to squares, while squares near the edges of X are mapped to rectangular shapes with large aspect ratios.

Example 4.5. Refer to Fig. 4.4. The source domain, X is given by a square with a medium square removed from the center, and with smaller squares removed from the four corners. The target domain, Y , is the set $\{|x|^{4/5} + |y|^{4/5} \leq 1\}$ with an inner diamond shape removed, given by $\{|x| + |y| \leq 9/25\}$.

We mapped the vertical and horizontal lines in the source domain to the target domain. The problem was solved on a 512^2 grid, but plotted on a 256^2 grid for clarity. We indicated the images of eight points, the inner and outer diagonal corners of X . Both shapes are symmetric about diagonal lines, and this symmetry is preserved by the map. The inner points in the target domain show discontinuities in the inverse map; points from far away in the source are mapped to points near, for example, A' . The dent near A' indicates the discontinuity: as more lines are used in the plot the size of the dent decreases.

Example 4.6. In this example we vary the cost function. Refer to Fig. 4.5, which is plotted for a grid of size 256^2 . The source density μ is piecewise constant on the rectangle $X = R = [-1, 1] \times [-1/2, 1/2]$. It is given, up to normalization, by

$$1_R + 1_{\{|x| \geq \frac{3}{4}, |y| \leq \frac{1}{2}\}} + 1_{\{|x| \geq \frac{3}{4}, |y| \leq \frac{1}{4}\}}$$

The regions are illustrated in the figure by color. Note that the density is lowest on the blue middle strip $R_m = \{|x| \leq \frac{3}{4}\} \subset R$ and highest in the red section.

The target density ν is uniform on

$$Y = \left\{ |x|^{8/5} + |y|^{8/5} \leq 1 \right\}.$$

The grid lines on X are mapped to Y , along with the colors of the lines. Since the density is not uniform on X , squares coming from higher density areas will be mapped to larger regions, compared to squares from lower density areas.

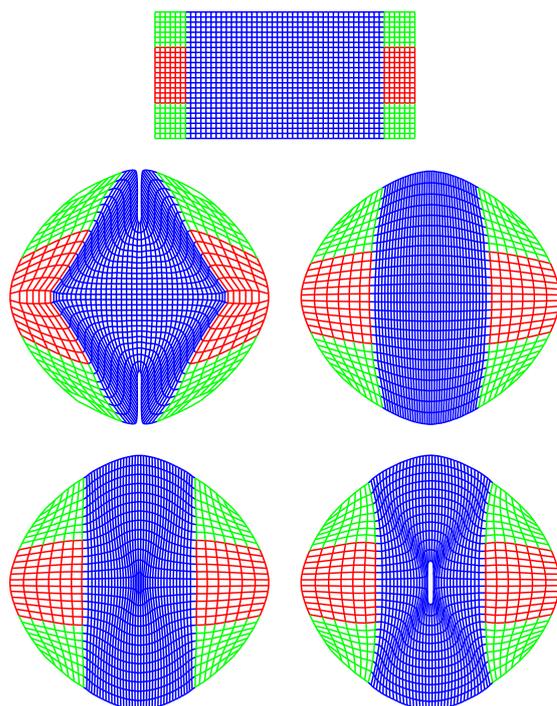


Fig. 4.5. Top: source density is piecewise constant on a rectangle, with different values indicated by color. Optimal map from source to uniform density on the target, with cost $c(x, y) = |x - y|^p$, for $p = 1.05, 2, 3, 6$, from left to right and top to bottom.

Consider the second figure, which corresponds to $p = 2$. In this case the low density middle strip of the rectangle, R_m , is mapped to a slightly convex vertical middle strip in the target. The higher density edges go to corresponding vertical cap shapes in the target. A square in the center of R is mapped to a rectangle in Y of aspect ratio about 2.

As p increases, a square in the center of R is mapped to a shape in Y with increasing height and decreasing width. The image of the boundary vertical lines $|x| = \frac{3}{4}$ in R_m go from convex ($p = 2$) to nearly linear ($p = 3$) to concave ($p = 6$).

Consider the first image, which corresponds to $p = 1.05$. In this case, R_m is mapped to an elongated diamond shape in the middle of Y . The midpoint of the top and bottom edges are mapped into the interior of the target, and their horizontal neighbors are folded in on each other. The two vertical dents the top and bottom of the target illustrate this folding, which means the inverse map from Y to R is discontinuous. (Similar results were obtained when the source was changed to a rounded rectangle.)

4.5. Partial Optimal Transportation

The partial optimal transportation problem, studied by Caffarelli and McCann [11] and Figalli [16], extends the OT problem with quadratic costs to the case where there is an excess of mass, and only a given fraction is to be transported. The result is a free boundary problem to determine which mass should be transported to minimize the total transportation cost. Uniqueness of the mapping was established in [11] when X and Y are disjoint, and then by Figalli for when $X \cap Y \neq \emptyset$ when the total mass to be transported is in excess of the measure

on the intersection. Figalli proved local C^1 regularity of the free boundaries away from the intersection when X and Y are strictly convex.

The linear programming discretization (LP) is easily modified to include partial mass transportation, by replacing the marginal projection equalities with an inequality, along with a single equality constraint to reflect the total mass to be transported.

$$\begin{aligned} \text{Minimize } I[\pi] &= \sum_{i=1}^n \sum_{j=1}^m c_{ij} \pi_{ij}, \\ \text{Subject to: } &\begin{cases} \sum_{j=1}^m \pi_{ij} \leq \mu_i, & i \in \{1, \dots, n\}, \\ \sum_{i=1}^n \pi_{ij} \leq \nu_j, & j \in \{1, \dots, m\}, \\ \sum_{i,j} \pi_{ij} = m, \\ \pi_{ij} \geq 0. \end{cases} \end{aligned}$$

We consider simple geometries for X and Y , using uniform measures on the domains. We verified that free boundaries are never mapped to free boundaries, which is consistent with Figalli’s prediction.

Each plot shows multiple free boundaries, corresponding to increasing the total mass to be transported. See Fig. 4.6. Note that the apparent kinks in the free boundary lines are plotting artifacts, since we plotted the support squares of the free boundaries without smoothing. It would also be reasonable to smooth the free boundaries using piecewise linear interpolation.

Example 4.7. Here X and Y are upward and downward facing parabolas, separated by a gap. The plot shows several the partial optimal transportation free boundaries as the mass to be transported is increased. The free boundary is both flat, and smooth. See Fig. 4.6 top left.

The remaining examples consider X and Y which intersect. In these cases, the total mass to be transported is expressed as a multiple of the total mass of the overlap.

Example 4.8. Here X and Y are intersecting squares. The problem was solved on a 512^2 grid. See Fig. 4.6 top right. For small excess mass ratios, the partial transportation free boundary is small set near the corners of overlap of the squares. Only when the excess mass ratio is increased to 1.11 does the free boundary cover the full intersection of the squares.

Example 4.9. Now X and Y are upward and downward facing parabolas which overlap. See Fig. 4.6 bottom left and right. We see very different results compared to the case with squares. In this case, even for very small excess mass ratios, the partial transportation free boundary contains the intersection set $X \cap Y$. We plotted two figures. In the first, the mass ratio goes from 1.01 to 1.11, with steps of 0.02. For a mass ratio of 1.01, the excess mass is not large compared to the resolution of the grid, but it is clear that as the mass is increased to 1.11, the full intersection is transported. In the second figure, we steps of 0.04 in the mass ratio, going from 1.05 to 1.25. At the mass ratio of 1.05, the free boundary is several grid points away from the intersection set.

Notice the difference between the two examples. At small excess mass ratios, for the squares, the free boundary is clearly a subset of the intersection, while for the parabolas, it covers the

entire intersection. An obvious difference between the two examples is that the furthest point of the parabolas are not nearly as far away as the furthest points of the squares. Another way to describe the difference is in terms of the tangents to the free boundaries. For the parabolas, the tangents to the free boundaries change very little, from a small positive slope to a small negative one. For the squares, the tangents to the free boundaries go from nearly horizontal, to nearly vertical.

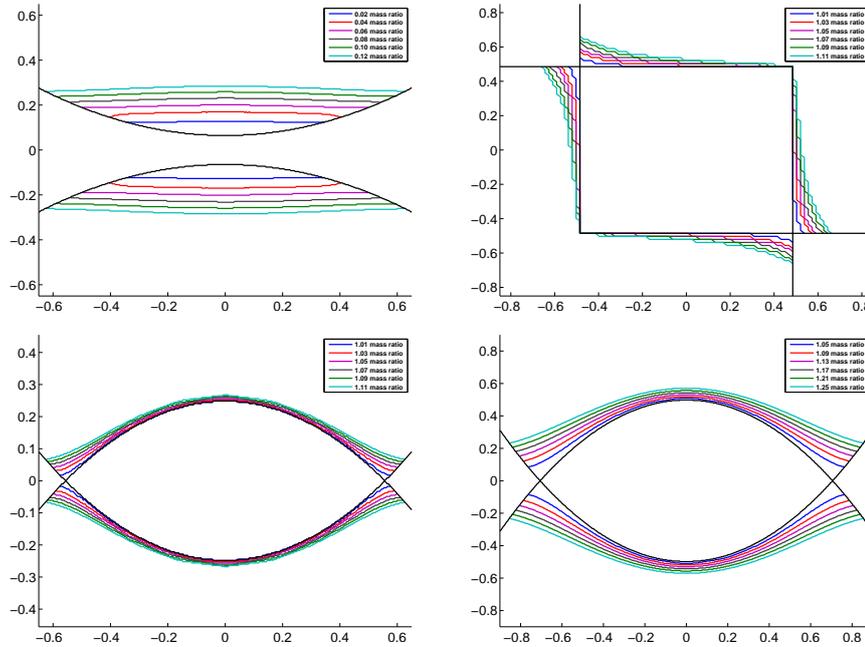


Fig. 4.6. Partial optimal transportation free boundaries corresponding to different levels of transported mass. Top left: Example 4.7 measures supported on non-overlapping on parabolas. Top right: Example 4.8 overlapping squares. Bottom: Example 4.9 overlapping paraboloids.

4.6. Barycenter

Given the points $x = (x_1, \dots, x_n)$ and the positive weights μ_1, \dots, μ_n , which sum to unity, the Euclidean barycenter $\bar{\mu} = \sum_{i=1}^n \mu_i x_i$ is the minimizer of

$$I(x) = \sum_{i=1}^n \mu_i |x - x_i|^2.$$

The Wasserstein distance allows us to define the barycenter of multiple measures.

By analogy, given the measures ρ_1, \dots, ρ_n and the positive weights μ_1, \dots, μ_n , which sum to unity, the barycenter is the minimizer over probability measures of

$$J(\rho) = \sum_{i=1}^n \mu_i W_2^2(\rho, \rho_i),$$

where W_2^2 is the value of the optimal transportation problem with quadratic costs. Variants include using other costs, $c(x, y) = |x - y|^p$. Refer to Agueh and Carlier [1]. The barycenter problem can be discretized as a linear program, see [13].

Example 4.10. The barycenter of three shapes is shown in Fig. 4.2. This example is taken from [5, Figure 7], where it was computed on a lower resolution grid. We also computed the barycenters of two rectangles, as in [13] and found very similar results (not plotted).

Example 4.11. Fig. 4.7 shows the barycenter of three sections of an annulus. The support of the barycenter becomes smaller as p increases. The barycenter inherits some of the symmetries of the measures.

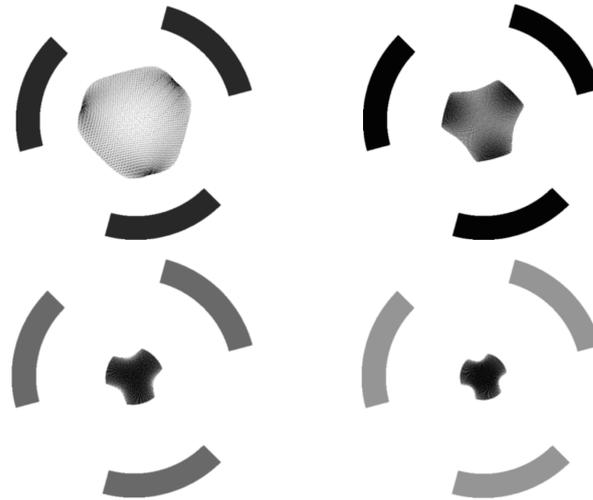


Fig. 4.7. Barycenter of three sections of an annulus. From left to right and top to bottom: cost $c(x, y) = |x - y|^p$, $p = 1.05, 2, 5, 9$. Grid size 512^2 .

Example 4.12. The next example shows a range of barycenters of uniform measures supported on four shapes, as in [26]. Fig. 4.8 shows the solution computed on a 1024^2 grid, without smoothing. The solution is accurate, and the boundary of the shapes are sharp, compared to the blurring of boundaries introduced by entropic regularization.

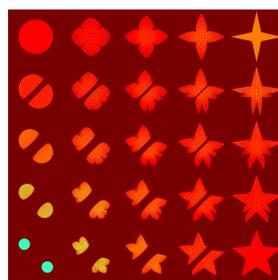


Fig. 4.8. Barycenters of the four shapes in the corners, computed on a grid of size 1024^2 .

5. Conclusions

We implemented a Linear Programming (LP) method for solving Optimal Transportation problems on general domains for a wide range of cost functions.

In practice, the method is linear in complexity (in terms of both time and memory), and allowed for problem sizes of up to half a million variables to be solved in a few minutes on a laptop. Available parallel LP solvers allow even larger problems to be solved.

We proved weak convergence of the approximation, and implemented the barycentric projection of the measure to improve the accuracy of the mapping. The barycentric projection also converges to the solution.

The method was accurate enough to compare solution features for optimal transportation maps between non-convex shapes for different costs functions.

The method was also applied to generalizations of the Optimal Transportation problem, including barycenters and partial Optimal Transportation. Other generalizations are possible, as long as these problems can be expressed as Linear Programs.

References

- [1] M. Agueh and G. Carlier, Barycenters in the wasserstein space, *SIAM Journal on Mathematical Analysis*, **43**:2 (2011), 904–924.
- [2] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient flows: in metric spaces and in the space of probability measures*, Springer Science & Business Media, 2006.
- [3] J.D. Benamou and Y. Brenier, A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem, *Numer. Math.*, **84**:3 (2000), 375–393.
- [4] J.D. Benamou and G. Carlier, Augmented lagrangian methods for transport optimization, mean field games and degenerate elliptic equations, *Journal of Optimization Theory and Applications*, **167**:1 (2015), 1–26.
- [5] J.D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, Iterative bregman projections for regularized transportation problems, *SIAM Journal on Scientific Computing*, **37**:2 (2015), A1111–A1138.
- [6] J.D. Benamou, G. Carlier, and M. Laborde, An augmented lagrangian approach to wasserstein gradient flows and applications, *ESAIM: Proceedings and Surveys*, **54** (2016), 1–17.
- [7] J.D. Benamou, G. Carlier, and L. Nenna, A numerical method to solve optimal transport problems with coulomb cost, *arXiv preprint arXiv:1505.01136*, 2015.
- [8] J.D. Benamou, G. Carlier, and F. Santambrogio, Variational mean field games, In *Active Particles, Volume 1*, 141–171. Springer International Publishing, 2017.
- [9] J.D. Benamou, B.D. Froese, and A.M. Oberman, Numerical solution of the optimal transportation problem using the monge–ampère equation, *Journal of Computational Physics*, **260**:0 (2014), 107–126.
- [10] N. Bonneel, G. Peyré, and M. Cuturi, Wasserstein barycentric coordinates: Histogram regression using optimal transport, *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2016)*, **35**:4 (2016).
- [11] L. Caffarelli and R.J. McCann, Free boundaries in optimal transport and monge-ampere obstacle problems, *Annals of Mathematics*, **171**:2 (2010), 673–730.
- [12] G. Carlier, Duality and existence for a class of mass transportation problems and economic applications, In *Advances in mathematical economics*, 1–21. Springer, 2003.
- [13] G. Carlier, A.M. Oberman, and E. Oudet, Numerical methods for matching for teams and wasserstein barycenters, *arXiv preprint arXiv:1411.3602*, 2014.
- [14] M. Cuturi, Sinkhorn distances: Lightspeed computation of optimal transport, In *Advances in Neural Information Processing Systems*, (2013), 2292–2300.
- [15] M. Cuturi, G. Peyré, and A. Rolet, A smoothed dual approach for variational wasserstein problems, *arXiv preprint arXiv:1503.02533*, 2015.

- [16] A. Figalli, The optimal partial transport problem, *Archive for rational mechanics and analysis*, **195**:2 (2010), 533–560.
- [17] W. Gangbo and R.J. McCann, The geometry of optimal transportation, *Acta Mathematica*, **177**:2 (1996), 113–161.
- [18] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent, Optimal mass transport for registration and warping, *International Journal of computer vision*, **60**:3 (2004), 225–240.
- [19] B. Lévy, A numerical algorithm for l2 semi-discrete optimal transport in 3d, *ESAIM: Mathematical Modelling and Numerical Analysis*, **49**:6 (2015), 1693–1715.
- [20] Q. Mérigot, A multiscale approach to optimal transport, In *Computer Graphics Forum*, volume 30, 1583–1592. Wiley Online Library, 2011.
- [21] L. Métivier, R. Brossier, Q. Mérigot, E. Oudet, and J. Virieux, An optimal transport approach for seismic tomography: application to 3d full waveform inversion, *Inverse Problems*, **32**:11 (2016), 115008.
- [22] F. Santambrogio, Optimal transport for applied mathematicians, *Birkhäuser, NY*, 2015.
- [23] B. Schmitzer, A sparse algorithm for dense optimal transport, In *Scale Space and Variational Methods in Computer Vision*, 629–641. Springer, 2015.
- [24] B. Schmitzer and C. Schnörr, A hierarchical approach to optimal transport, In *Scale Space and Variational Methods in Computer Vision: 4th International Conference, SSVM 2013*, Springer, **7893** (2013), 452.
- [25] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, volume 26. CRC press, 1986.
- [26] J. Solomon, F.P. de Goes, P.A. Studios, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas, Convolutional wasserstein distances: Efficient optimal transportation on geometric domains, *ACM Transactions on Graphics (TOG)*, **34**:4 (2015), 1–11.
- [27] E. G. Tabak and G. Triglia, Data-driven optimal transport, *Commun. Pure. Appl. Math.* doi, 2014, 10:1002.
- [28] C. Villani, *Topics in Optimal Transportation*, American Mathematical Soc., 2003.
- [29] C. Villani, *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media, 2008.
- [30] Y. Yang, B. Engquist, J. Sun, and B. D. Froese, Application of optimal transport and the quadratic Wasserstein metric to full-waveform inversion, *Geophysics*, **83** (2018), R43-R62.