Numerische Mathematik

A fast approach to optimal transport: the back-and-forth method

Matt Jacobs[1] · Flavien Léger[1]

© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

We present a method to efficiently solve the optimal transportation problem for a general class of strictly convex costs. Given two probability measures supported on a discrete grid with n points we compute the optimal transport map between them in $O(n \log(n))$ operations and $O(n)$ storage space. Our approach allows us to solve optimal transportation problems on spatial grids as large as 4096×4096 and $384 \times 384 \times 384$ in a matter of minutes.

1 Introduction

The optimal transportation problem was first introduced by Monge in 1781, to find the most cost-efficient way to transport mass from a set of sources to a set of sinks. The theory was modernized and revolutionized by Kantorovich in 1942, who found a key link between optimal transport and linear programming. In recent years, there has been an explosion of interest in optimal transport thanks in part to the discovery of deep connections between the quadratic-cost optimal transport problem and a diverse class of partial differential equations (PDEs) arising in statistical mechanics and fluid mechanics ([3,6,7,17,21] to name just a few of the most prominent results). In addition, optimal transport has become popular in data science (particularly machine learning and image processing), where it provides a very natural way to compare and interpolate probability distributions [1,14].

✉ Matt Jacobs
   majaco@math.ucla.edu

   Flavien Léger
   flavienleger@nyu.edu

1   Department of Mathematics, UCLA, Los Angeles 90095, CA, USA

In this work, we are interested in computing the optimal transport problem for large-scale applications that arise in image processing and in numerical methods for solving PDEs (in both two and three dimensions). For these applications, one needs to accurately compute the optimal transport map on enormous computational domains (millions of grid points or pixels). Up to now, computing the optimal map has been a notoriously difficult task. To the best of our knowledge, all previously known methods for solving the optimal transport problem either do not scale linearly with respect to the problem size [3,5], cannot provide an accurate computation of the optimal map [10], or are only applicable to a limited class of probability densities [4].

Our goal in this paper is to remedy the current situation and provide an efficient and accurate algorithm for computing optimal transport maps. To that end, we introduce a new method for solving optimal transport problems: the back-and-forth method. Given two probability densities μ and ν discretized on a grid with n points and a strictly convex cost function, our approach computes the optimal map using $O(n)$ storage space and $O(n \log(n))$ computation time per iteration. The number of iterations required for ϵ accuracy grows like $O\left(\max(\|\mu\|_\infty, \|\nu\|_\infty) \log(1/\epsilon)\right)$ in all tested experiments. As a result, the method converges extremely rapidly. Notably, in contrast to many other approaches, the back-and-forth method does not require positive lower bounds on the probability densities. Thus, we can solve optimal transport problems with densities that vanish on large portions of the domain. In total, the efficiency and flexibility of the back-and-forth method allows us to solve optimal transport problems that were out of reach for previous schemes. The back-and-forth method can solve problems on grids of size 4096×4096 and $384 \times 384 \times 384$ in a matter of minutes on a personal computer and could be accelerated even further by applying parallelization or other scientific computing techniques.

1.1 Overall approach

The back-and-forth method is based on solving the Kantorovich dual formulation of optimal transport. To introduce this formulation, suppose we are given two probability measures μ and ν and a cost function $c(x, y)$, which measures the cost to move a unit of mass from location x to location y (for now we shall be deliberately vague about underlying spaces). The optimal cost to transport μ to ν is given by the value of the Kantorovich dual problem:

$$\sup_{\phi, \psi} \int \phi \, d\nu + \int \psi \, d\mu,$$

where the Kantorovich potentials ϕ and ψ are two scalar functions constrained by the inequality

$$\phi(y) + \psi(x) \leq c(x, y). \tag{1}$$

In addition to computing the optimal cost, the dual problem also encodes information about the optimal map. If an optimal map from μ to ν exists (or vice-versa), it can be recovered from the maximizers ϕ_*, ψ_* of the dual problem. Indeed, in this case, the

pair (x, y) is in the graph of the map if and only if $\phi_*(y) + \psi_*(x) = c(x, y)$. Thus, to compute optimal maps it is enough to solve the dual problem.

In what follows, we will consider two equivalent unconstrained formulations of the dual problem. These formulations are based upon the observation that if ϕ_* and ψ_* are the maximizers of the dual problem, then one must have the relations

$$\phi_*(y) = \psi_*^c(y) := \inf_x c(x, y) - \psi_*(x),$$

and

$$\psi_*(x) = \phi_*^c(x) := \inf_y c(x, y) - \phi_*(y).$$

Here we make use of an operator $\phi \to \phi^c$ (resp. $\psi \to \psi^c$) called the c-transform, which plays an important role in optimal transport and in our proposed method. Note that if one views the dual problem as a linear programming problem, then the above two equations are precisely the complementary slackness condition.

Now we see that there are two equivalent ways to remove the constraint (1), we can either replace ϕ by ψ^c or we can replace ψ by ϕ^c. This leads to the twin functionals

$$J(\phi) = \int \phi \, dv + \int \phi^c \, d\mu,$$

and

$$I(\psi) = \int \psi^c \, dv + \int \psi \, d\mu,$$

which encode exactly the same problem, just posed in different spaces. J formulates the problem in "ϕ-space" and I formulates the problem in "ψ-space". Here one can draw an analogy to the Fourier transform: indeed, the reader is probably already quite familiar with functionals that can be equivalently expressed in either physical space or Fourier space.

The back-and-forth method solves the Kantorovich dual problem by hopping back-and-forth between gradient ascent updates on J in ϕ-space and gradient ascent updates on I in ψ-space (hence the name). Gradient are taken with respect to the $\dot{H}^1$ metric (see (2)). In between gradient steps, information in one space is propagated back to the other space by taking a c-transform (c.f. Algorithm 1).

The advantage of the back-and-forth approach is that certain features of the optimal solution pair (ϕ_*, ψ_*) may be easier to build in one space compared to the other. For example, the Hessian of $\psi = \phi^c$ is very closely related to the inverse of the Hessian of ϕ. Thus, large Hessian eigenvalues in one space correspond to small Hessian eigenvalues in the other space, and smaller features can be built in fewer gradient steps. By hopping back-and-forth between the two spaces, we get the best of both worlds—there is always an opportunity to build features in the space where they are smaller. As a result, the back-and-forth method converges far more rapidly than vanilla gradient ascent methods that operate only on ϕ-space or only on ψ-space. Indeed, on certain examples, our method produces a 10,000-fold reduction in error in just 2–4 additional iterations (c.f. Tables 1, 2).

In addition, let us highlight that the choice of gradient ascent steps in the $\dot{H}^1$ metric appears to be crucial for maintaining the stability of the algorithm. Indeed, it seems that it is not possible to guarantee an increase in the value of the dual problem if the gradient ascent steps are taken in any Hilbert space weaker than $\dot{H}^1$ (see Sect. 3.1 and Proposition 1 for more details).

Turning to the computational efficiency of the back-and-forth method, we see that the scheme amounts to performing $\dot{H}^1$ gradient ascent iterations alternatively on J and I (i.e. in ϕ-space and ψ-space), and computing c-transforms. Crucially, the derivative of J (and, by symmetry, the derivative of I) takes a simple form [12] which can be efficiently computed. Furthermore, for a large class of costs, the c-transform can be computed extremely efficiently. On a discrete grid with n points, an exact c-transform can be computed in $O(n \log(n))$ operations [9,19] (see Sect. 4.1 for more details).

Finally, let us note that a gradient ascent method on the Kantorovich dual problem was previously considered in [8]. In their paper, the authors took L^2 gradient ascent steps on only one form of the dual problem. Furthermore, the authors tried to avoid computing c-transforms, and thus were forced to use an inaccurate estimation of the true gradient. Unsurprisingly, the authors note that their scheme appears to be unstable. Indeed, we believe that the ideas behind the back-and-forth method are necessary to obtain a robust, efficient, and accurate gradient ascent type method for solving the Kantorovich dual problem.

1.2 Future work and paper outline

While the present work is focused on the optimal transport problem, we anticipate that the back-and-forth method will prove useful for many other problems. For instance, the back-and-forth method can be readily adapted to compute Wasserstein gradient flows. This opens the door to large-scale simulations of a wide class of important and interesting PDEs, [6,16,17,21] to name just a few. In addition, the method may prove useful to solve computational problems arising from the burgeoning area of mean field games [15,18]. We look forward to exploring these applications in future work.

The rest of the paper is organized as follows. In Sect. 2, we recall the necessary background information for optimal transport and gradient-based optimization schemes. In Sect. 3, we introduce the back-and-forth method and provide arguments for its stability and efficiency. In Sect. 4, we discuss the numerical implementation of the algorithm, and conduct various experiments to demonstrate its performance.

2 Background

2.1 Optimal transport and the *c*-transform

Let Ω be a convex and compact subset of $\mathbb{R}^d$. A *cost* on ω is a continuous function $c \colon \omega \times \omega \to \mathbb{R}$. In the theory of optimal transport fairly general costs can be considered,

however in the present work we will focus on the case

$$c(x, y) = h(y - x),$$

for a strictly convex and even function $h : \mathbb{R}^d \to \mathbb{R}$. Given two probability measures μ and ν supported in ω, the Monge formulation of the optimal transport problem is defined by

$$C(\mu, \nu) = \inf_T \int_\omega c(x, T(x)) \, d\mu(x),$$

where the infimum runs over maps $T : \omega \to \omega$ which transport μ to ν, i.e. such that $T_\# \mu = \nu$. We recall that the pushforward measure $T_\# \mu$ is defined by $T_\# \mu(A) = \mu(T^{-1}(A))$ for any measurable subset $A \subset \omega$. One can also characterize the pushforward by defining the integral of the pushforward measure against continuous test functions $f : \Omega \to \mathbb{R}$:

$$\int_\Omega f(y) \, d(T_\# \mu)(y) = \int_\Omega f(T(x)) \, d\mu(x).$$

This second formulation is extremely useful for studying pushforwards in the context of optimal transport.

Throughout the paper, we will concentrate on the special case where both probability measures μ and ν are absolutely continuous with respect to the Lebesgue measure. As a result, we will frequently conflate a measure and its density function. Under this assumption, there exists a unique optimal map T_*, which pushes μ to ν, and its inverse T_*^{-1} is the optimal map that pushes ν to μ [7,12]. Furthermore, one can find the optimal map by solving the Kantorovich dual formulation of the optimal transport problem.

The dual formulation can be derived by introducing Lagrange multipliers for the pushforward constraint. Indeed, the pushforward constraint $T_\# \mu = \nu$ holds if and only if

$$\int_\Omega \phi(T(x)) \, d\mu(x) = \int_\Omega \phi(y) \, d\nu(y)$$

for every continuous function ϕ. Therefore,

$$\inf_{T_\# \mu \,=\, \nu} \int_\omega c(x, T(x)) \, d\mu(x) = \inf_T \sup_\phi \int_\omega c(x, T(x)) \, d\mu(x)$$
$$- \phi(T(x)) \, d\mu(x) + \int_\omega \phi(y) \, d\nu(y).$$

Under the assumption that μ is absolutely continuous, the interchange of infimum and supremum is valid [22,23]. If we then group the terms involving T we arrive at

$$\sup_\phi \inf_T \int_\omega (c(x, T(x)) - \phi(T(x))) \, d\mu(x) + \int_\omega \phi(y) \, d\nu(y).$$

Now it is clear that the optimal choice of T at some point x_0 is completely decoupled from the choice at any other point x. Therefore, one can move the infimum inside of the integral, and it becomes clear that the operation $\inf_{T(x)} c(x, T(x)) - \varphi(T(x))$ plays an important role in the dual problem. This operation is known as the c-transform, and it is at the heart of optimal transport. The c-transform maps a function ϕ to another function ϕ^c and it can be seen as a generalization of the Legendre transform (convex conjugation) from convex analysis.

Definition 1 Given a continuous function $\phi \colon \omega \to \mathbb{R}$, we define its c-transform $\phi^c \colon \omega \to \mathbb{R}$ by

$$\phi^c(x) = \inf_{y \in \omega} c(x, y) - \phi(y).$$

Additionally, we say that ϕ is c-concave if there exists a continuous function $\psi \colon \omega \to \mathbb{R}$ such that $\phi = \psi^c$ and we say that a pair of functions (ϕ, ψ) is c-conjugate if $\phi = \psi^c$ and $\psi = \phi^c$.

We are now ready to introduce the Kantorovich dual functional

$$J(\phi) = \int_\omega \phi \, dv + \int_\omega \phi^c \, d\mu.$$

Our above discussion then shows that

$$C(\mu, v) = \sup_\phi J(\phi),$$

where the supremum runs over continuous functions $\phi \colon \omega \to \mathbb{R}$. Note that in this section we will just focus on one formulation of the dual problem. It is easy to see that any properties satisfied by J must also be satisfied by the alternative formulation, which we previously denoted in the introduction as I.

The following lemma encapsulates the most important properties of the c-transform (from the perspective of optimal transport).

Lemma 1 [12,13] *Let $\phi : \Omega \to \mathbb{R}$ be a continuous function. The c-transform satisfies the following properties:*

(i) *$\phi(x) \leq \phi^{cc}(x)$ for every $x \in \Omega$, and $\phi^{cc} = \phi$ if and only if ϕ is c-concave. As a result, $\phi^{ccc} = \phi^c$ for any continuous function ϕ.*

(ii) *If ϕ is c-concave, the minimization problem $\inf_{y \in \Omega} c(x, y) - \phi(y)$ has a unique minimizer $T_\phi(x)$ for almost every x. Furthermore, we have the explicit formula:*

$$T_\phi(x) = x - (\nabla h)^{-1}(\nabla \phi^c(x)),$$

where we recall that $c(x, y) = h(y - x)$.

(iii) If φ is c-concave and u is a continuous function on ω, then

$$\lim_{\epsilon \to 0} \frac{(\phi + \epsilon u)^c(x) - \phi^c(x)}{\epsilon} = -u(T_\phi(x))$$

for almost every $x \in \Omega$.

With these properties in hand, one can establish one of the most fundamental results in optimal transport.

Theorem 1 [7,12,13] *The dual problem*

$$\sup_{\phi} \int_\omega \phi \, d\nu + \int_\omega \phi^c \, d\mu =: J(\phi)$$

satisfies the following properties:

  (i) J is concave with respect to φ.
 (ii) If μ is absolutely continuous with respect to the Lebesgue measure, then J is maximized by a c-concave function ϕ_* and T_{ϕ_*} is the unique optimal map which pushes μ to ν.
(iii) If ν is also absolutely continuous, then T_{ϕ_*} is invertible almost everywhere, and $T_{\phi_*}^{-1}$ is the optimal map which pushes ν to μ.

From Theorem 1, we see that the optimal map can be computed by solving a concave maximization problem. Thus standard techniques of optimization, such as gradient ascent, can be leveraged to solve the optimal transport problem. However, in order to use gradient ascent effectively, one must choose the correct notion of distance and step size. For this reason, we review gradient ascent in Hilbert spaces below.

2.2 Gradient ascent

In this section we recall elementary results on constant step-size gradient ascent methods for concave functions. Let $(\mathcal{H}, \|\cdot\|_{\mathcal{H}})$ be a separable Hilbert space and suppose that F is a smooth convex functional

$$F : \mathcal{H} \to \mathbb{R}.$$

We first recall the notions of differential map and gradients.

Definition 2 Given a point $\phi \in \mathcal{H}$ we say that a bounded linear map $\delta F_\phi : \mathcal{H} \to \mathbb{R}$ is the first variation (Fréchet derivative) of F at ϕ if

$$\lim_{\|h\|_{\mathcal{H}} \to 0} \frac{\|F(\phi + h) - F(\phi) - \delta F_\phi(h)\|_{\mathcal{H}}}{\|h\|_{\mathcal{H}}} = 0.$$

Definition 3 Let $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ be the inner product associated with the Hilbert space $\mathcal{H}$. We say that a map $\nabla_{\mathcal{H}} F \colon \mathcal{H} \to \mathcal{H}$ is the $\mathcal{H}$-gradient of F if

$$\langle \nabla_{\mathcal{H}} F(\phi), h \rangle_{\mathcal{H}} = \delta F_{\phi}(h)$$

for all $(\phi, h) \in \mathcal{H} \times \mathcal{H}$.

In the back-and-forth method, we will make use of the $\dot{H}^1$-gradient where $\dot{H}^1(\Omega)$ is the Hilbert space

$$\dot{H}^1(\Omega) = \{\varphi \colon \Omega \to \mathbb{R} : \int_{\Omega} \varphi(x)\,dx = 0 \ \text{ and } \ \int_{\Omega} |\nabla\varphi(x)|^2\,dx < \infty\}, \quad (2)$$

with inner product

$$\langle \varphi_1, \varphi_2 \rangle_{\dot{H}^1} = \int_{\Omega} \nabla\varphi_1(x) \cdot \nabla\varphi_2(x)\,dx.$$

Let us also mention the dual space $\dot{H}^{-1}(\omega)$: for two probability densities ρ_1 and ρ_2 we define

$$\|\rho_2 - \rho_1\|_{\dot{H}^{-1}}^2 = \int_{\omega} |\nabla\phi(x)|^2\,dx, \quad (3)$$

where ϕ is the unique solution in $\dot{H}^1(\omega)$ to the Laplace equation $-\triangle \phi = \rho_2 - \rho_1$ with zero Neumann boundary conditions (for more details we refer to [22, Section 5.5.2]).

The following lemma shows that the $\dot{H}^1$-gradient has a particularly simple form. The proof is a straightforward application of integration by parts.

Lemma 2 *Suppose that $F \colon \dot{H}^1(\omega) \to \mathbb{R}$ is a Fréchet-differentiable function such that for any $\phi \in \dot{H}^1(\omega)$, the first variation δF_{ϕ} evaluated at any point $h \in \dot{H}^1(\omega)$ can be written as integration against a function f_{ϕ}, i.e.*

$$\delta F_{\phi}(h) = \int_{\Omega} h(x) f_{\phi}(x)\,dx.$$

Then the $\dot{H}^1$-gradient of F has the form

$$\nabla_{\dot{H}^1} F(\phi) = (-\Delta)^{-1} \bar{f}_{\phi},$$

where $(-\Delta)^{-1}$ is the inverse negative Laplacian operator taken with zero Neumann boundary conditions and $\bar{f}_{\phi} = f_{\phi} - \frac{1}{|\Omega|}\int_{\Omega} f_{\phi}$.

Now that we have seen an example of a $\mathcal{H}$-gradient, we are ready to discuss gradient ascent methods on general Hilbert spaces. Gradient ascent maximizes a Fréchet-differentiable functional $F \colon \mathcal{H} \to \mathbb{R}$ by iterating

$$\phi_{n+1} = \phi_n + \sigma \nabla_{\mathcal{H}} F(\phi_n),$$

where the step-size $\sigma > 0$ is a fixed constant. In order to obtain convergence to the supremum, i.e.

$$F(\phi_n) \to \sup F,$$

one needs to control the continuity of the gradient mapping $\nabla_{\mathcal{H}} F$. The following convergence theorem for gradient ascent is one of the cornerstones of optimization.

Theorem 2 (See for instance [20]) *Let $F: \mathcal{H} \to \mathbb{R}$ be a Fréchet-differentiable concave functional and suppose that there exists $\sigma > 0$ such that*

$$F(\phi) \geq F(\hat{\phi}) + \delta F_{\hat{\phi}}(\phi - \hat{\phi}) - \frac{1}{2\sigma} \|\phi - \hat{\phi}\|_{\mathcal{H}}^2$$

for all $\phi, \hat{\phi} \in \mathcal{H}$. Then the gradient ascent iterations

$$\phi_{n+1} = \phi_n + \sigma \nabla_{\mathcal{H}} F(\phi_n)$$

satisfy the ascent property

$$F(\phi_{n+1}) - F(\phi_n) \geq \frac{\sigma}{2} \|\nabla_{\mathcal{H}} F(\phi_n)\|_{\mathcal{H}}^2.$$

Furthermore, if F has a unique maximizer ϕ_* and if $\sup_n \|\phi_n\|_{\mathcal{H}} < \infty$ then the sequence $\{\phi_n\}_{n=0}^{\infty}$ converges weakly to ϕ_*.

For completeness, we give a short proof of Theorem 2 in the "Appendix". With these tools in hands we are now ready to introduce the back-and-forth method.

3 The back-and-forth method

Let $\omega \subset \mathbb{R}^d$ be a convex and compact region and consider a cost $c: \omega \times \omega \to \mathbb{R}$ of the form

$$c(x, y) = h(y - x),$$

for a strictly convex and even function $h: \mathbb{R}^d \to \mathbb{R}$. Given two probability densities μ and ν supported in ω, we are interested in the optimal transport problem in its dual form

$$C(\mu, \nu) = \sup_{\phi} \int_{\omega} \phi(y)\, \nu(y)\, dy + \int_{\omega} \phi^c(x)\, \mu(x)\, dx, \tag{4}$$

where the supremum runs over continuous functions $\phi: \omega \to \mathbb{R}$, and the c-transform is defined by $\phi^c(x) = \inf_{y \in \omega} c(x, y) - \phi(y)$. For more background on the optimal transport problem and on c-transforms we refer to Sect. 2.1.

Our goal in this section is to develop an efficient algorithm to compute the maximizer of the dual problem. In what follows, we will consider the dual functional in the two following equivalent forms:

$$J(\phi) = \int_\omega \phi \, d\nu + \int_\omega \phi^c \, d\mu \tag{5}$$

and

$$I(\psi) = \int_\omega \psi^c \, d\nu + \int_\omega \psi \, d\mu. \tag{6}$$

Note that the functionals J and I are essentially identical, the only difference is that the roles of μ and ν are flipped. In order to proceed further, we need formulas for the variations of J and I.

Lemma 3 [11,12] *Consider the functionals J and I defined by* (5) *and* (6) *over the space of continuous functions $\phi\colon \omega \to \mathbb{R}$ and $\psi\colon \omega \to \mathbb{R}$ respectively. If ϕ is c-concave, the first variation of J can be expressed as*

$$\delta J_\phi = \nu - T_{\phi \#}\mu,$$

and if ψ is c-concave, the first variation of I can be expressed as

$$\delta I_\psi = \mu - T_{\psi \#}\nu.$$

Here we recall that for any c-concave function $\varphi : \omega \to \mathbb{R}$,

$$T_\varphi(x) = x - (\nabla h)^{-1}(\nabla \varphi^c(x)).$$

A short proof of Lemma 3 is provided in the appendix for the reader's convenience.

Now we are ready to introduce our approach to solve the optimal transport problem: the back-and-forth method. The method is outlined in Algorithm 1. It is based on two main ideas:

1. Gradient ascent steps in the $\dot{H}^1$ metric, where

$$\nabla_{\dot{H}^1} J(\phi) = (-\Delta)^{-1} \left(\nu - T_{\phi \#}\mu \right),$$
$$\nabla_{\dot{H}^1} I(\psi) = (-\Delta)^{-1} \left(\mu - T_{\psi \#}\nu \right). \tag{7}$$

2. A back-and-forth update scheme, alternating between gradient ascent steps on J and I.

In the following two subsections, we will motivate the choice of $\dot{H}^1$-gradient steps and the back-and-forth updates. For information about the step size σ, see Sect. 4.2.

Algorithm 1: The back-and-forth method

Given pro bability densities μ and ν, set $\phi_0 = 0$, $\psi_0 = 0$, and iterate:

$$\phi_{n+\frac{1}{2}} = \phi_n + \sigma \nabla_{\dot{H}^1} J(\phi_n),$$

$$\psi_{n+\frac{1}{2}} = (\phi_{n+\frac{1}{2}})^c,$$

$$\psi_{n+1} = \psi_{n+\frac{1}{2}} + \sigma \nabla_{\dot{H}^1} I(\psi_{n+\frac{1}{2}}),$$

$$\phi_{n+1} = (\psi_{n+1})^c.$$

3.1 $\dot{H}^1$-gradient ascent

The main steps of the back-and-forth method are the $\dot{H}^1$-gradient ascent steps

$$\phi_{n+\frac{1}{2}} = \phi_n + \sigma \nabla_{\dot{H}^1} J(\phi_n),$$

and

$$\psi_{n+1} = \psi_{n+\frac{1}{2}} + \sigma \nabla_{\dot{H}^1} I(\psi_{n+\frac{1}{2}}).$$

In order to obtain the convergence of the back-and-forth method, we will need to know if these steps increase the values of the dual functionals J and I respectively. In what follows, we will focus exclusively on the J update step, since any properties that hold for J will also hold for I by symmetry.

Recalling Theorem 2 from the previous section, one can show that for a general Hilbert space $\mathcal{H}$, the $\mathcal{H}$-gradient ascent steps

$$\phi_{n+1} = \phi_n + \sigma \nabla_{\mathcal{H}} J(\phi_n)$$

satisfy the ascent property

$$J(\phi_{n+1}) \geq J(\phi_n) + \frac{\sigma}{2} \|\nabla_{\mathcal{H}} J(\phi_n)\|_{\mathcal{H}}^2$$

if the inequality

$$J(\phi) \geq J(\hat{\phi}) + \delta J_{\hat{\phi}}(\phi - \hat{\phi}) - \frac{1}{2\sigma} \|\phi - \hat{\phi}\|_{\mathcal{H}}^2. \tag{8}$$

holds for some $\sigma > 0$ and any $\phi, \hat{\phi} \in \mathcal{H}$. Note that this inequality becomes easier to satisfy when the norm associated to $\mathcal{H}$ becomes stronger i.e. when right hand side becomes more negative.

To establish the inequality, we need to bound from below

$$J(\phi) - J(\hat{\phi}) - \delta_{\hat{\phi}} J(\phi - \hat{\phi}),$$

which can be recognized as the error in approximating J with a first-order Taylor expansion about $\hat{\phi}$. The easiest way to make progress is to use the concave inequality $J(\hat{\phi}) \leq J(\phi) + \delta J_\phi(\hat{\phi} - \phi)$ to get the bound

$$J(\phi) - J(\hat{\phi}) - \delta_{\hat{\phi}} J (\phi - \hat{\phi}) \geq \left(\delta J_\phi - \delta J_{\hat{\phi}} \right) (\phi - \hat{\phi}).$$

We can use the explicit formula for the first variation in Lemma 3, to write

$$\left(\delta J_\phi - \delta J_{\hat{\phi}} \right) (\phi - \hat{\phi}) = \int_\Omega (\phi - \hat{\phi})(T_{\hat{\phi} \#} \mu - T_{\phi \#} \mu).$$

Now inequality (8) will follow if we can show that

$$\int_\Omega (\phi - \hat{\phi})(T_{\hat{\phi} \#} \mu - T_{\phi \#} \mu) \geq -\frac{1}{2\sigma} \|\phi - \hat{\phi}\|_{\mathcal{H}}^2.$$

Thus, our goal is to find a Hilbert space $\mathcal{H}$ and a parameter σ that make this inequality true.

For any Hilbert space $\mathcal{H}$, we have the Cauchy–Schwartz inequality

$$\int_\Omega (\phi - \hat{\phi})(T_{\hat{\phi} \#} \mu - T_{\phi \#} \mu) \geq -\|\phi - \hat{\phi}\|_{\mathcal{H}} \|T_{\hat{\phi} \#} \mu - T_{\phi \#} \mu\|_{\mathcal{H}^*},$$

where $\mathcal{H}^*$ is the dual space to $\mathcal{H}$ with respect to the L^2 inner product (i.e. we dualize $\mathcal{H}$ with respect to the so-called *pivot space* $L^2(\omega)$). If we can show that for an appropriate choice of Hilbert space $\mathcal{H}$,

$$\|T_{\hat{\phi} \#} \mu - T_{\phi \#} \mu\|_{\mathcal{H}^*} \leq \frac{1}{2\sigma} \|\phi - \hat{\phi}\|_{\mathcal{H}}, \tag{9}$$

then combining this with our bound from Cauchy–Schwartz we will get (8) (note a more careful argument can eliminate the factor of $\frac{1}{2}$ on the right side of (9), but we won't worry about this here). Once again, inequality (9) becomes easier to satisfy as the norm associated to $\mathcal{H}$ becomes stronger. Indeed, as $\mathcal{H}$ becomes stronger $\mathcal{H}^*$ becomes weaker, thus the left hand side gets smaller while the right hand side gets larger.

We can interpret inequality (9) as saying that the function $\phi \mapsto T_{\phi \#} \mu$ needs to be Lipschitz continuous as a map from $\mathcal{H}$ to $\mathcal{H}^*$. Thus, to get a gradient ascent scheme with the ascent property, we will need to choose a Hilbert space $\mathcal{H}$ so that if ϕ and $\hat{\phi}$ are close in $\mathcal{H}$ then $T_{\phi \#} \mu$ and $T_{\hat{\phi} \#} \mu$ are close in $\mathcal{H}^*$. This is not an easy task. For example, if ϕ is not c-concave, the map $\phi \mapsto T_\phi$ may not even be well-defined, let alone have certain continuity properties.

If one tries to deal with the above complication head-on, it is not clear how to proceed. To make progress, let us show that the ascent property will hold in the Hilbert space $\dot{H}^1(\Omega) = \{\phi \colon \Omega \to \mathbb{R} : \int_\omega \phi = 0 \text{ and } \|\nabla \phi\|_{L^2(\Omega)} < \infty\}$ if we restrict our attention to the quadratic cost and certain well-behaved c-concave functions. Before

we state our result, let us note that the choice of $\dot{H}^1$ probably cannot be weakened. Indeed, the formula for T_ϕ depends on $\nabla\phi^c$, thus one must have some control on the gradients of ϕ and $\hat{\phi}$ to have any hope of showing either (8) or (9).

Proposition 1 *Consider the Kantorovich dual problem* $\sup_\phi J(\phi)$ *with the quadratic cost* $h(y - x) = \frac{1}{2}|x - y|^2$. *If we know that there exists* $\lambda > 0$ *such that the iterates*

$$\phi_n, \phi_{n+\frac{1}{2}} \in \mathcal{S}_\lambda := \{\phi : (1 - \lambda^{-1})I \leq D^2\phi(x) \leq (1 - \lambda)I\}$$

for every n, then the gradient ascent step

$$\phi_{n+\frac{1}{2}} = \phi_n + \sigma\nabla_{\dot{H}^1}J(\phi_n)$$

with step size $\sigma = \|\mu\|^{-1}\lambda^{d+1}$ *satisfies the ascent property*

$$J(\phi_{n+\frac{1}{2}}) - J(\phi_n) \geq \frac{1}{2}\|\mu\|^{-1}\lambda^{d+1}\|\nabla_{\dot{H}^1}J(\phi_n)\|_{\dot{H}^1}^2.$$

Moreover, the mass densities move closer to the target v,

$$\|\rho_{n+\frac{1}{2}} - v\|_{\dot{H}^{-1}}^2 - \|\rho_n - v\|_{\dot{H}^{-1}}^2 \leq -\|\rho_{n+\frac{1}{2}} - \rho_n\|_{\dot{H}^{-1}}^2,$$

where we denote $\rho_n = T_{\phi_n\#}\mu$ *and* $\rho_{n+\frac{1}{2}} = T_{\phi_{n+\frac{1}{2}}\#}\mu$ *and where the* $\dot{H}^{-1}$ *distance is defined by* (3).

See Appendix for proof.

Remark 1 Proposition 1 shows that the descent property holds over $\dot{H}^1(\Omega)$ when all of the iterates ϕ_n are assumed to have upper and lower Hessian bounds. Unfortunately, we cannot expect this property to hold throughout the simple ascent scheme $\phi_{n+1} = \phi_n + \sigma\nabla_{\dot{H}^1}J(\phi_n)$. Indeed, the gradient ascent updates may eventually push the iterates outside of the desired region. In fact, we cannot even guarantee that c-concavity is preserved by the gradient update steps.

Remark 2 One way to avoid the aforementioned problem is to add a step where the iterates are projected back onto the set

$$\mathcal{S}_\lambda = \{\phi : (1 - \lambda^{-1})I \leq D^2\phi(x) \leq (1 - \lambda)I\}.$$

$\mathcal{S}_\lambda$ is convex, thus the projection step does not interfere with the ascent property. Unfortunately, it is expensive to compute projections onto $\mathcal{S}_\lambda$ (one must solve a semi-definite programming problem). Furthermore, the maximizer of the dual problem may not lie in $\mathcal{S}_\lambda$ for any $\lambda > 0$ (however for the quadratic cost it must be in $\mathcal{S}_0$ by c-concavity). In such a case, one would need to send $\lambda \to 0$ to obtain an arbitrarily accurate solution.

To circumvent some of the difficulties mentioned in the above remarks, we replace traditional gradient ascent steps with a back-and-forth scheme that alternates between maximizing J and maximizing I.

3.2 Back-and-forth updates

Let us recall that the Kantorovich dual problem can be written in the form

$$C(\mu, \nu) = \sup_{\phi, \psi} \int_\omega \phi(y)\nu(y)\,dy + \int_\omega \psi(x)\mu(x)\,dx, \tag{10}$$

where the supremum runs over continuous functions ϕ and ψ satisfying

$$\phi(y) + \psi(x) \le h(y - x),$$

for all $x, y \in \omega$. This formulation emphasizes the symmetric role played by the potentials ϕ and ψ. In what follows, it will be convenient to represent the dual functional in the form

$$D(\phi, \psi) = \int_\omega \phi\,\nu\,dy + \int_\omega \psi\,\mu\,dx - \iota_C(\phi, \psi), \tag{11}$$

where the constraints are encoded by the convex indicator function ι_C which takes value 0 on the convex set

$$\mathcal{C} = \{(\phi, \psi) : \forall x, y \in \omega \quad \phi(y) + \psi(x) \le h(y - x)\},$$

and $+\infty$ for all other pairs (ϕ, ψ). Then the Kantorovich problem (10) can be simply written $C(\mu, \nu) = \sup_{\phi, \psi} D(\phi, \psi)$. Moreover the functionals J and I can be obtained by either eliminating ϕ or ψ from the symmetric representation (11). Indeed, one can check that

$$\phi^c = \operatorname{argmax}_\psi D(\phi, \psi), \quad \psi^c = \operatorname{argmax}_\phi D(\phi, \psi),$$

and thus

$$J(\phi) = D(\phi, \phi^c) = \sup_\psi D(\phi, \psi), \quad I(\psi) = D(\psi^c, \psi) = \sup_\phi D(\phi, \psi).$$

A vanilla gradient ascent scheme $\phi_{n+1} = \phi_n + \sigma \nabla_{\dot{H}^1} J(\phi_n)$ focuses (arbitrarily) on ϕ-space, iterating gradient steps on the functional J. Alternatively, one could work instead in ψ-space and write a gradient ascent scheme on the functional I. A perhaps better idea is to alternate between ϕ-space and ψ-space. This is the core idea of the back-and-forth method given in Algorithm 1. Recall that a current iterate (ϕ_n, ψ_n) is

updated as follows:

$$\phi_{n+\frac{1}{2}} = \phi_n + \sigma \nabla_{\dot{H}^1} J(\phi_n),$$
$$\psi_{n+\frac{1}{2}} = (\phi_{n+\frac{1}{2}})^c,$$
$$\psi_{n+1} = \psi_{n+\frac{1}{2}} + \sigma \nabla_{\dot{H}^1} I(\psi_{n+\frac{1}{2}}),$$
$$\phi_{n+1} = (\psi_{n+1})^c.$$

The back-and-forth approach already corrects a difficulty we identified in the previous subsection. When one considers a pure gradient ascent scheme $\phi_{n+1} = \phi_n + \sigma \nabla_{\dot{H}^1} J(\phi_n)$ on ϕ and J only (or analogously on ψ and I only), there is no reason that the iterates remain c-concave. If ϕ or ψ is not c-concave, then the gradients $\nabla_{\dot{H}^1} J(\phi)$ or $\nabla_{\dot{H}^1} I(\psi)$ may not be well defined. In the back-and-forth method, whenever one takes a gradient, the function in question is always c-concave. Indeed, $\phi_n = \psi_{n-1}^c$ and $\psi_{n+\frac{1}{2}} = \phi_{n+\frac{1}{2}}^c$ thus, $\nabla_{\dot{H}^1} J(\phi_n)$ and $\nabla_{\dot{H}^1} I(\psi_{n+\frac{1}{2}})$ are always well-defined.

In addition, the back-and-forth updates can never perform worse than a pure gradient ascent scheme. Indeed, the intermediate steps where we take a c-transform can only increase the value of the dual problem. It follows from property (i) in Lemma 1 that

$$J(\phi_{n+\frac{1}{2}}) \leq J\left(\phi_{n+\frac{1}{2}}^{cc}\right) = I(\psi_{n+\frac{1}{2}}),$$

and

$$I(\psi_{n+1}) \leq I\left(\psi_{n+1}^{cc}\right) = J(\phi_{n+1}).$$

Thus, the intermediate c-transform steps help increase the value of the dual functional. If we combine this with our previous result from Sect. 3.1, we can conclude the following result about the back-and-forth method.

Proposition 2 *Consider the back-and-forth method (Algorithm 1) for the quadratic-cost optimal transport problem. Assume that there exists $\lambda > 0$ such that all the iterates $\phi_n, \phi_{n+\frac{1}{2}}$ and $\psi_n, \psi_{n+\frac{1}{2}}$ lie in the set of functions*

$$\mathcal{S}_\lambda = \{\varphi : (1 - \lambda^{-1})I \leq D^2\varphi(x) \leq (1 - \lambda)I\}.$$

Then we have the chain of inequalities

$$D(\phi_{n+1}, \psi_{n+1}) \geq D(\phi_{n+\frac{1}{2}}, \psi_{n+\frac{1}{2}}) \geq D(\phi_n, \psi_n),$$

for all integers n. Equivalently, the values of I and J alternatively increase:

$$I(\psi_{n+1}) \geq I(\psi_{n+\frac{1}{2}}) \geq J(\phi_{n+\frac{1}{2}}) \geq J(\phi_n) \geq I(\psi_n).$$

The short proof of Proposition 2 is given in the "Appendix".

Note that even with the back-and-forth updates we cannot give a rigorous convergence proof without assuming convexity bounds on the iterates. Nonetheless, for reasons that we do not fully understand, the back-and-forth method appears to be extremely stable even for relatively large step sizes (a vanilla ascent method on either J or I behaves less favorably). Furthermore, the back-and-forth method converges extremely rapidly to the maximizer. Heuristically, we believe this is because it is easier to build certain features in either "ϕ-space" or "ψ-space". Indeed, if (ϕ, ψ) is a pair of c-conjugate functions for the quadratic cost, one has the relationship $D^2\phi(x - \nabla\phi(x)) = (D^2\psi(x))^{-1}$ i.e. the quadratic c-transform inverts bounds on Hessians. By alternating between spaces, one can build features in the space where they are smaller, and thus converge to the solution more rapidly.

4 Numerical implementation and results

4.1 The fast c-transform

Computing the c-transform on a regular grid requires solving the optimization problem

$$\phi^c(x) = \inf_y h(y - x) - \phi(y)$$

for each grid point x. In one dimension, the strict convexity of the cost implies that the y derivative

$$h'(y - x) - \phi'(y)$$

is a decreasing function of x. As a result, the minimizers $y(x) = \text{argmin}_y h(y - x) - \phi(y)$ are monotone increasing with respect to x. In other words, if we have $x_1 \leq x_2$ then $y(x_1) \leq y(x_2)$. This observation can be exploited to design a divide-and-conquer algorithm that computes a 1-dimensional c-transform on n points in $O(n \log(n))$ operations [9].

The key idea is that if one knows the minimizers $y(x)$ for points the points $x = \frac{k}{n}, \frac{k+2}{n}$ then

$$y\left(\frac{k}{n}\right) \leq y\left(\frac{k+1}{n}\right) \leq y\left(\frac{k+2}{n}\right),$$

i.e. the minimizer for the middle point is "trapped" between the other two. Thus, one can separately compute a c-transform for the even grid points and a c-transform for the odd grid points then use the above interlacing property to reconstruct the full solution. See [9] for the explicit algorithm and more details.

If we restrict our attention to costs h which decompose along each dimension, i.e.

$$h(y - x) = \sum_{i=1}^{d} h_i(y_i - x_i),$$

then we can compute a d-dimensional c-transform by repeated applications of the 1-dimensional c-transform. For example, in 2 dimensions, we have

$$\inf_y h(y - x) - \phi(y) = \inf_{y_2} \inf_{y_1} h_2(y_2 - x_2) + h_1(y_1 - x_1) - \phi(y_1, y_2)$$
$$= \inf_{y_2} h_2(y_2 - x_2) + \phi(\cdot, y_2)^c(x_1).$$

If we write $\tilde{\phi}(x_1, y_2) = -\phi(\cdot, y_2)^c(x_1)$ then the last line is a c-transform with respect to y_2

$$\inf_{y_2} h_2(y_2 - x_2) - \tilde{\phi}(x_1, y_2).$$

Thus, in 2 dimensions, one can compute the full c-transform by first computing one-dimensional c-transforms along all horizontal lines, and then along all vertical lines. This idea generalizes to arbitrary dimension, and is the exact same mechanism which is used to compute the multidimensional fast Fourier transform (FFT).

In the (extremely important) special case $h(y - x) = \frac{1}{2}|y - x|^2$, the c-transform can be computed even more efficiently. Splitting the quadratic part into 3 terms, we get

$$\phi^c(x) = \frac{1}{2}|x|^2 + \inf_y \left[-x \cdot y + \frac{1}{2}|y|^2 - \phi(y) \right].$$

If we then set $\varphi(y) = \frac{1}{2}|y|^2 - \phi(y)$, and let $\varphi^*(x)$ denote the Legendre transform of φ:

$$\varphi^*(x) = \sup_y x \cdot y - \varphi(y), \qquad (12)$$

then

$$\phi^c(x) = \frac{1}{2}|x|^2 - \varphi^*(x).$$

Therefore, one can compute the quadratic c-transform by computing a Legendre transform instead.

A one-dimensional Legendre transform on a set of n points can be computed in $O(n)$ operations [19]. The idea of [19] uses two very important properties of the Legendre transform, namely that $\varphi^* = \varphi^{***} = (\varphi^{**})^*$ and that φ^{**} is the convex hull of φ. Therefore, computing (12) amounts to finding the values $y(x)$ where

$$x - (\varphi^{**})'(y) \qquad (13)$$

changes sign from positive to negative. φ^{**} is convex, so the slopes $(\varphi^{**})'(y)$ are increasing. Therefore, finding the sign change $y(x)$ that corresponds to each x in (13) can be done in a single sweep through the values of $(\varphi^{**})'(y)$. On a one-dimensional grid with n regularly spaced points, the sweep step takes $O(n)$ operations and crucially, the convex hull φ^{**} can also be computed in $O(n)$ operations [19]. Finally, a multidimensional Legendre transform on a regular grid can be computed by decomposing the problem along each dimension in the same manner as described above for the c-transform. For the explicit algorithm and more details, we refer to [19].

4.2 Step sizes

As we saw in Sect. 3, it is not easy to determine the Lipschitz constant of the gradient mapping at an arbitrary function ϕ. As such, it is not clear how to optimally choose σ for a fixed step size gradient ascent scheme. Thus, rather than choose a single fixed value σ at the outset of the algorithm, we update $\sigma = \sigma_n$ throughout the algorithm using Armijo–Goldstein type update rules [2].

After each gradient step, we compare the difference $J(\phi_{n+1}) - J(\phi_n)$ to the squared norm $\|\nabla_{\dot{H}^1} J(\phi_n)\|_{\dot{H}^1}^2$ (and correspondingly for I and ψ_{n+1}, ψ_n). Given parameters $0 < \beta_1 < \beta_2 < 1, \alpha_1 > 1$ and $\alpha_2 < 1$ we check whether

$$-\sigma_n \beta_2 \|\nabla_{\dot{H}^1} J(\phi_n)\|_{\dot{H}^1}^2 \leq J(\phi_n) - J(\phi_{n+1}) \leq -\sigma_n \beta_1 \|\nabla_{\dot{H}^1} J(\phi_n)\|_{\dot{H}^1}^2.$$

If the upper inequality fails, we decrease σ_n by taking $\sigma_{n+1} = \alpha_2 \sigma_n$, and if the lower inequality fails then we increase σ_n by taking $\sigma_{n+1} = \alpha_1 \sigma_n$. In all of our experiments we take $\beta_1 = \frac{1}{4}, \beta_2 = \frac{3}{4}, \alpha_1 = \frac{5}{4}$, and $\alpha_2 = \frac{4}{5}$ and we choose $\sigma = 8 \min(\|\mu\|_{L^\infty}^{-1}, \|\nu\|_{L^\infty}^{-1})$ as our starting value.

Note that we do not perform any backtracking. The new value of $\sigma = \sigma_{n+1}$ is simply used in the next update step. Indeed, evaluating J or I at a function requires computing a c-transform, thus it is not worth trying to optimize the step size at a particular iterate.

4.3 Experiments

Throughout this section we will assume that $\Omega = [0, 1]^d$ is the unit cube in $\mathbb{R}^d$. When implementing our algorithms, we discretize Ω using a regular finite d-dimensional grid. Recall that Laplace equations on this domain should be solved with zero Neumann boundary conditions. This ensures that there is no flux of mass outside of the computational domain.

All numerical algorithms were coded in C and executed on a single 1.6 GHz core with 8 GB RAM. Inversion of the Laplace operator was performed using the fast Fourier transform (FFT). All FFTs were calculated using the free FFTW C library.

In what follows, we will consider three sets of experiments. In the first two sets of experiments, we restrict our attention to optimal transport with the quadratic cost $\frac{1}{2}|x - y|^2$. We start by computing optimal transport maps between geometric shapes where explicit answers are available. This allows us to verify the accuracy and convergence

rate of the method. Next, we compute optimal maps between images, and use the maps to compute image interpolations. As we shall see, the optimal maps are able to capture the fine-scale details of the images. Finally, in the last set of experiments, we consider the more general class of costs outlined in Sect. 4.1, and study how the optimal map varies as the underlying cost changes.

Quadratic cost optimal transport

In this subsection, we focus on computing optimal transport with the quadratic cost

$$h(y - x) = \frac{1}{2}|y - x|^2,$$

which represents the most important special case of optimal transport.

We begin with a simple example where the two probability densities μ and ν differ only by a translation, i.e. $\mu(x) = \nu(x + a)$ for a fixed constant $a \in \mathbb{R}^d$. In this case, it is known that the optimal map T_* pushing μ to ν is itself a translation: $T_*(x) = x + a$, and the transportation cost is $\frac{1}{2}|a|^2$. Thus, we can directly check the accuracy of our method on this class of elementary examples.

First, in two dimensions, suppose that μ is uniform and supported on a disc of radius $\frac{1}{8}$ centered at $(\frac{1}{4}, \frac{1}{4})$ and ν is uniform and supported on a disc of the same radius centered at $(\frac{3}{4}, \frac{3}{4})$ (both densities are normalized to have unit mass).

μ $\qquad$ ν
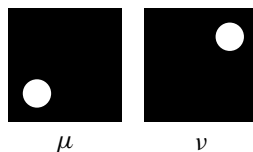
It then follows that the quadratic transportation cost between these densities is

$$\frac{1}{2}\left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = \frac{1}{4}.$$

Our results for this experiment are presented in Table 1. We run our algorithm until the difference between the computed transportation cost and the exact solution is less than the prescribed error tolerance. We see that to get an error of 10^{-4} we only need 3 iterations, and to get an error of 10^{-8} we only need 5 iterations. Let us also highlight that the performance of the algorithm is completely independent of the tested grid size.

Next we repeat the same experiment in three dimensions. In this case, μ is uniform and supported on a ball centered at $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ with radius $\frac{1}{8}$ and ν is uniform and supported on a ball centered at $(\frac{3}{4}, \frac{3}{4}, \frac{3}{4})$ with the same radius. The cost to transport the densities to one another is $\frac{3}{8}$. Our results for this experiment are presented in Table 2. Again, we run our algorithm until the difference between the computed transportation cost and the exact solution is less than the prescribed error tolerance. Once again

Table 1 Two discs in 2 dimensions

Grid size	Error 10^{-4}		Error 10^{-8}	
	Iterations	Time (s)	Iterations	Time (s)
512×512	3	1.00	5	1.25
1024×1024	3	4.23	5	5.21
2048×2048	3	16.7	5	21.75
4096×4096	3	56.46	5	75.70

Table 2 Two balls in 3 dimensions

Grid size	Error 10^{-4}		Error 10^{-8}	
	Iterations	Time (s)	Iterations	Time (s)
$128 \times 128 \times 128$	6	17.89	10	27.00
$256 \times 256 \times 256$	6	156.94	9	205.5
$384 \times 384 \times 384$	6	472.14	9	699.21

we see that the algorithm converges extremely rapidly, and the convergence rate is independent of the grid size.

Now we turn to a more difficult example where the optimal transport map is discontinuous. In two dimensions, we take μ to be the renormalized characteristic function of a square with side lengths $\frac{1}{4}$ centered at $(\frac{1}{2}, \frac{1}{2})$ and ν to be the renormalized characteristic function of the union of four squares each with side length $\frac{1}{8}$ and centers at $(\frac{3}{16} + \frac{5i}{8}, \frac{3}{16} + \frac{5j}{8})$ for $i, j \in \{0, 1\}$.

μ ν
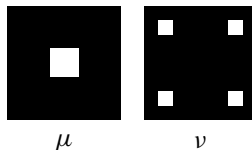
In this case, there is an explicit solution for the optimal transport map. One cuts μ into 4 squares along the lines $y = \frac{1}{2}$ and $x = \frac{1}{2}$ and then each cut square moves to the closest square in ν by translation. Thus, we see that the map is discontinuous along the lines $y = \frac{1}{2}$ and $x = \frac{1}{2}$. Each cut square is translated by a vector of the form $(\pm\frac{1}{4}, \pm\frac{1}{4})$, therefore, the transportation cost is $\frac{1}{2}\left((\frac{1}{4})^2 + (\frac{1}{4})^2\right) = \frac{1}{16}$. The results for this experiment are presented in Table 3. Note that due to the discontinuity in the optimal map, it is not possible to achieve the same level of accuracy as the first example. Nonetheless, we see that the algorithm still converges to a highly accurate solution in a small number of iterations.

Finally, we consider an analogue of this discontinuous example in 3 dimensions. In this setting, we take μ to be the renormalized characteristic function of a cube with side length $\frac{1}{4}$ centered at $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ and ν to be the renormalized characteristic function of the union of eight cubes each with side length $\frac{1}{8}$ and centers at $(\frac{3}{16} + \frac{5i}{8}, \frac{3}{16} + \frac{5j}{8}, \frac{3}{16} + \frac{5k}{8})$ for $i, j, k \in \{0, 1\}$. For this problem, the optimal transportation map is obtained by cutting

Table 3 One square to four squares

Grid size	Error 10^{-4}		Error 10^{-5}		Error 10^{-6}	
	Iterations	Time (s)	Iterations	Time (s)	Iterations	Time (s)
512×512	3	1.08	5	1.45	13	2.96
1024×1024	3	4.98	5	5.82	14	12.2
2048×2048	3	18.9	5	25.2	14	53.9
4096×4096	3	68.3	5	93.6	13	217

Table 4 One cube to eight cubes

Grid size	Error 10^{-3}		Error 10^{-5}	
	Iterations	Time (s)	Iterations	Time (s)
$128 \times 128 \times 128$	3	13.36	6	21.63
$256 \times 256 \times 256$	3	117.44	8	255.17
$384 \times 384 \times 384$	3	377.05	13	1577.23

μ into eight cubes along the lines $x = \frac{1}{2}$, $y = \frac{1}{2}$, and $z = \frac{1}{2}$ and translating each of the eight cubes to the closest cube in ν. Each cube is translated by a vector of the form $(\pm\frac{1}{4}, \pm\frac{1}{4}, \pm\frac{1}{4})$, thus, the total transportation cost is $\frac{1}{2}\left((\frac{1}{4})^2 + (\frac{1}{4})^2 + (\frac{1}{4})^2\right) = \frac{3}{32}$. Once again, due to the discontinuity of the map, the accuracy depends on the grid resolution. As a result, for grids of size $384 \times 384 \times 384$ and smaller, the method becomes stationary once the accuracy drops slightly below 10^{-5}. Nonetheless, we continue to see that the algorithm converges rapidly to a stationary state.

Optimal transport of images

In our next set of experiments, we consider optimal transport between two arbitrary black and white images. We shall assume that the image pixels take values in $[0, 1]$, where 0 represents black and 1 represents white. The images can then be converted into probability densities by renormalizing pixel values so that each image has total mass 1.

In the context of image processing, one typically wishes to compute optimal transport maps to obtain realistic looking interpolations between images. We will focus on the quadratic cost, which is the most natural for this task. Given two images with associated probability densities μ, ν and the optimal map T_* between them, one can define for each time $t \in [0, 1]$ the interpolation

$$\rho(t, x) = S_t \#\mu,$$

where $S_t(x) = tT_*(x) + (1-t)x$. At time zero, S_0 is the identity map, so $\rho(0, x) = \mu(x)$, and at time 1, $S_1 = T_*$, so $\rho(1, x) = \nu(x)$. The intermediate times $t \in (0, 1)$ essentially give a "video" which shows how the optimal map deforms one probability density into the other.

Fig. 1 Several image examples used to construct interpolations

We compute two different examples of image interpolation, Figs. 2 and 3, using the images in Fig. 1 as our initial and final densities. In Fig. 2, the starting image is the silhouette of a jack-o'-lantern and the final image is a silhouette of a winged horse. In Fig. 3, the starting configuration is the union of 4 differently sized discs and the ending configuration is the silhouette of a man holding a sword. Note that in both examples, the intermediate time interpolations remain sharp. In addition, near times 0 and 1, the discontinuities of the optimal map are well localized and resolved. Figure 2 is particularly interesting, as one can see how the delicate features of the legs, wings, and head of the horse form over time.

Even though both examples are large (1024 × 1024 pixels) and computed to a high degree of accuracy, the total computation time to find the optimal map and construct the interpolation does not exceed 20 s in either case. To the best of our knowledge, the size of the examples and the accuracy that we achieved were out of reach for all previous computational optimal transport methods.

Optimal transport with general costs

In the final set of experiments, we consider optimal transport with more general cost functions. As we noted earlier, the *c*-transform corresponding to any cost of the form
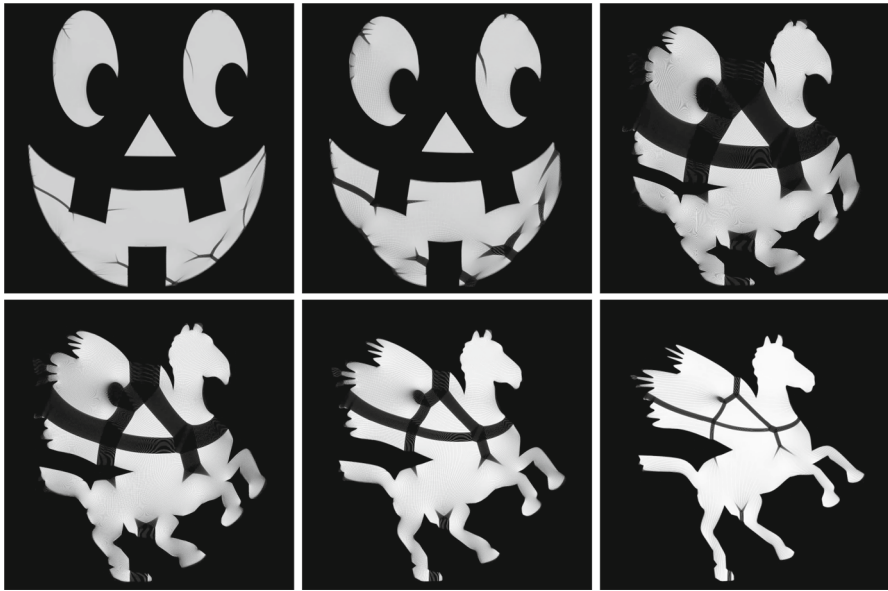
Fig. 2 Quadratic cost optimal transport between two 1024×1024 pixel images. The total computation time to compute the optimal map and then compute the interpolation was 20 s
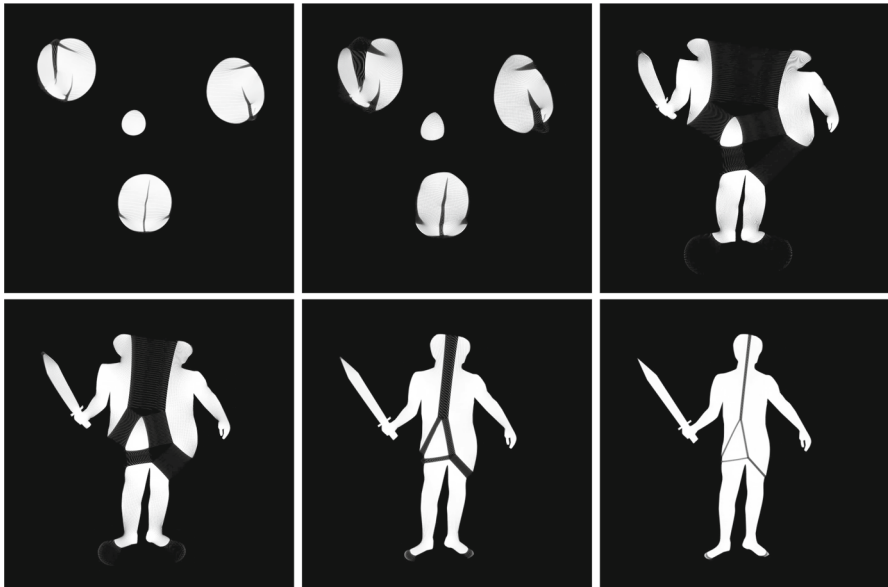

Fig. 3 Quadratic cost optimal transport between two 1024×1024 pixel images. The total computation time to compute the optimal map and then compute the interpolation was 18 s

$c(x, y) = \sum_{i=1}^{d} h_i(y_i - x_i)$ for h_i strictly convex can be computed efficiently using the fast c-transform. The most interesting costs of this form are sums of pth powers, i.e.

$$c(x, y) = \sum_{i=1}^{d} \frac{1}{p_i} |y_i - x_i|^{p_i}, \tag{14}$$

where $p_i > 1$ for each coordinate index i. These costs are particularly interesting when the powers p_i are different for each i. Indeed, in this case the optimal map can change considerably as the values of p_i are varied.

We illustrate this phenomenon in two dimensions by choosing μ to be the union of two discs of radius $\frac{1}{8}$ centered at $(\frac{1}{4}, \frac{1}{4})$ and $(\frac{3}{4}, \frac{3}{4})$ respectively, and ν to be the union of two discs of radius $\frac{1}{8}$ centered at $(\frac{1}{4}, \frac{3}{4})$ and $(\frac{3}{4}, \frac{1}{4})$ respectively:

$$\mu \qquad \nu$$
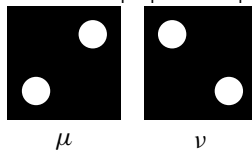
We then consider the optimal transport trajectory between μ and ν for several different choices of p_1 and p_2 (c.f. Fig. 4).

We start off with $p_1 = p_2 = 2$. In this case, each of the discs in μ splits evenly along the line $y = x$. The portion of the discs that lie above the line are transported to the disc in ν centered at $(\frac{1}{4}, \frac{3}{4})$ and the parts below the line are transported to the disc in ν centered at $(\frac{3}{4}, \frac{1}{4})$. When $p_1 < p_2$, it is cheaper to travel in the vertical direction as compared to the horizontal direction (for distances smaller than 1). As a result, when we decrease p_1 and increase p_2 the optimal map changes to transport more mass in the vertical direction and less mass in the horizontal direction. When the values become sufficiently extreme, (for example $p_1 = 1.1$ and $p_2 = 3$), horizontal transport becomes so expensive that the optimal cost map moves mass in the vertical direction only. See Fig. 4 for a visualization of the optimal trajectories for various choices of p_1 and p_2.

A Proofs

Proof of Theorem 2 The ascent property follows from plugging in the choices $\phi = \phi_{n+1}$ and $\hat{\phi} = \phi_n$ into the given inequality.

Now we suppose that F has a unique maximizer ϕ_* and that the sequence $\{\phi_n\}_{n=0}^{\infty}$ lies in a bounded subset of $\mathcal{H}$. We can then extract a weak limit point $\tilde{\phi}$ of the sequence, and by weak upper semi-continuity $F(\tilde{\phi}) \geq \lim_{n \to \infty} F(\phi_n)$ (note weak upper semi-continuity is automatic since we assume that F is concave and real-valued). The values $F(\phi_n)$ are bounded above by $F(\phi_*)$, therefore $\lim_{n \to \infty} \|\nabla_{\mathcal{H}} F(\phi_n)\|_{\mathcal{H}} = 0$. We can then establish the string of inequalities

$$F(\phi_*) \leq F(\phi_n) + \langle \nabla_{\mathcal{H}} F(\phi_n), \phi_* - \phi_n \rangle \leq F(\phi_n) + 2R \|\nabla_{\mathcal{H}} F(\phi_n)\|_{\mathcal{H}},$$
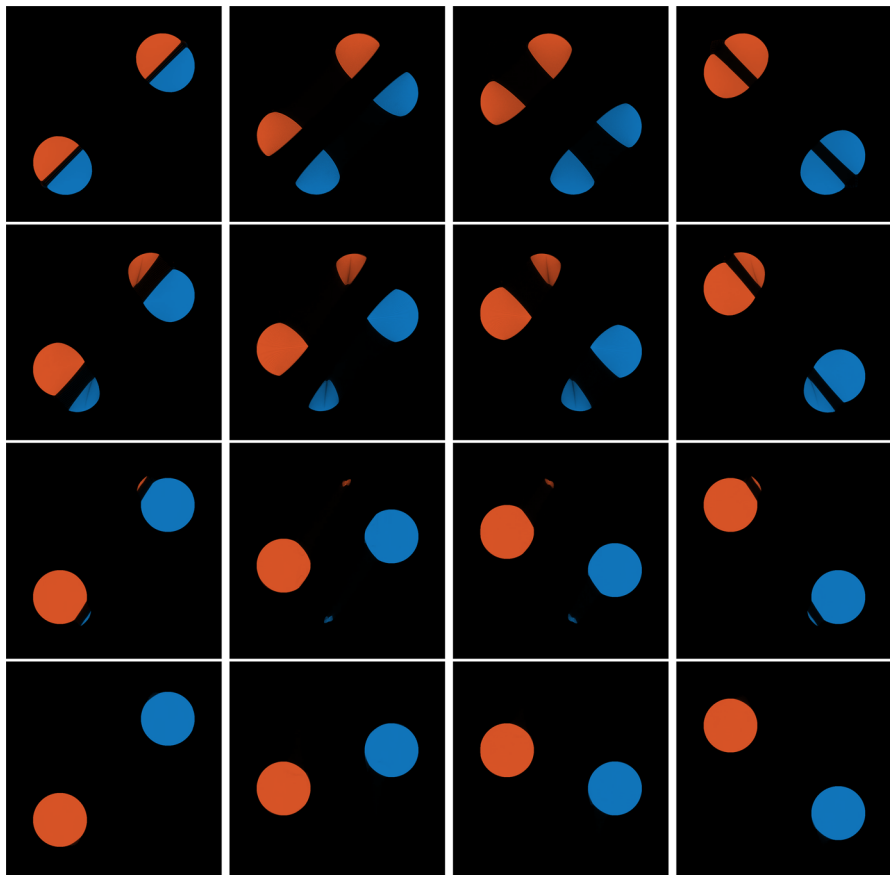
Fig. 4 Optimal transport with different cost functions. Each row depicts the optimal trajectory for a cost of the form (14). Row 1: $p_1 = 2$, $p_2 = 2$; Row 2: $p_1 = 1.75$, $p_2 = 2.25$; Row 3: $p_1 = 1.5$, $p_2 = 2.5$; Row 4: $p_1 = 1.1$, $p_2 = 3$. Colors represent an assignment: orange pixels move to the upper left disk while blue pixels move to the lower right disk (color figure online)

where the first inequality follows from the concavity of F and the second follows from choosing $R = \max(\|\phi_*\|_{\mathcal{H}}, \sup_n \|\phi_n\|_{\mathcal{H}})$. Taking a limit on the right hand side we can conclude that $F(\tilde{\phi}) \geq F(\phi_*)$, and thus, $\tilde{\phi} = \phi_*$ since the maximizer is unique. □

Proof of Fact 3 Here we prove the result on J only, the argument for I being identical by symmetry. Given a continuous function $u \colon \Omega \to \mathbb{R}$ we consider the perturbation

$$J(\phi + tu) = \int_\Omega (\phi + tu)^c(x) d\mu(x) + (\phi(x) + tu(x)) d\nu(x).$$

Using property (iii) of Lemma 1 we see that

$$\lim_{t \to 0} \frac{J(\phi + tu) - J(\phi)}{t} = \int_\Omega u(x)\nu(x) - u(T_\phi(x))\mu(x).$$

Thus,

$$\delta J_\phi(u) = \int_\Omega u \, (v - T_{\phi \#}\mu)$$

and the result then follows. □

We now turn our attention to Proposition 1. The notion of Bregman divergence will play an important role in its proof.

Definition 4 Let $\mathcal{H}$ be a Hilbert space and consider a Fréchet-differentiable function $F : \mathcal{H} \to \mathbb{R}$. The *Bregman divergence* associated to F is defined by

$$F(\phi_2|\phi_1) = F(\phi_2) - F(\phi_1) - \delta F_{\phi_1}(\phi_2 - \phi_1),$$

for all $\phi_1, \phi_2 \in \mathcal{H}$. Similarly, if ω is a closed and convex subset of $\mathbb{R}^d$ and $\xi : \omega \to \mathbb{R}$ is a differentiable function, we define

$$\xi(x_2|x_1) = \xi(x_2) - \xi(x_1) - \nabla \xi(x_1) \cdot (x_2 - x_1),$$

for all $x_1, x_2 \in \omega$.

As hinted by the above definition, we will make use of Bregman divergences both for Kantorovich potentials (defined on ω) and dual functionals (defined on the Hilbert space $\dot{H}^1(\omega)$). The next results gathers properties which are well-known in the theory of Bregman divergences.

Lemma 4 *Let ξ be a convex and differentiable function defined on ω, and denote by ξ^* its convex conjugate $\xi^*(p) = \sup_x p \cdot x - \xi(x)$. Then*

(i) *For any $\lambda \geq 0$, ξ is λ-strongly convex $\Leftrightarrow \xi(x_2|x_1) \geq \frac{\lambda}{2}|x_2 - x_1|^2$.*
(ii) *$\xi(x_2|x_1) = \xi^*(p_1|p_2)$ for all $x_1, x_2 \in \omega$, where we set $p_i = \nabla \xi(x_i)$.*
(iii) *For any $\lambda > 0$, if ξ is λ-strongly convex then $\xi(x_1|x_2) \leq \frac{\lambda^{-1}}{2}|p_2 - p_1|^2$, for all $x_1, x_2 \in \omega$, where $p_i = \nabla \xi(x_i)$.*
(iv) *Fix $x_0 \in \omega$. Let $f(u) = \xi(x_0 + u|x_0)$ and $g(v) = \xi^*(p_0 + v|p_0)$, where $p_0 = \nabla \xi(x_0)$. Then $f^*(v) = g(v)$.*

Now, recall that that the dual functional J is defined by $J(\phi) = \int \phi \, v + \int \phi^c \, \mu$. It is a concave functional, and note that the first term $\int \phi \, v$ is linear in ϕ and therefore plays no role in the convexity of J. We therefore set

$$F(\phi) = - \int \phi^c \, \mu, \tag{15}$$

a convex functional which essentially has the same convexity properties as J since they only differ by a linear term. For instance one can check directly that for any potentials ϕ_1 and ϕ_2 we have

$$J(\phi_2|\phi_1) = -F(\phi_2|\phi_1). \tag{16}$$

Finally, the convex conjugate of F is defined here by

$$F^*(\rho) = \int_\omega \phi\,\rho - F(\phi).$$

We are at this point ready to state the next lemma which is at the core of obtaining stability estimates in the gradient schemes.

Lemma 5 *Let ϕ_1 and ϕ_2 be two twice-differentiable real-valued functions defined on ω such that*

$$(1 - \lambda^{-1})I \le D^2\phi_i(x) \le (1 - \lambda)I,$$

for $i = 1, 2$, for some $\lambda > 0$. Moreover, let $\rho_i = T_{\phi_i\#}\mu$ the associated mass densities, for $i = 1, 2$. Then

$$F(\phi_2|\phi_1) \le \frac{1}{2}\|\mu\|\,\lambda^{-(d+1)}\|\phi_2 - \phi_1\|^2_{\dot{H}^1}. \tag{17}$$

Additionally,

$$F(\phi_2|\phi_1) \ge \frac{1}{2}\|\mu\|^{-1}\lambda^{d+1}\|\rho_2 - \rho_1\|^2_{H^{-1}}. \tag{18}$$

Proof **Step 1.** We start the proof by establishing the equality

$$F(\phi_2|\phi_1) = \int_\omega \left[\frac{1}{2}\left|T_{\phi_1}(x) - T_{\phi_2}(x)\right|^2 - \phi_2\left(T_{\phi_1}(x)\big|T_{\phi_2}(x)\right)\right]\mu(x)\,dx. \tag{19}$$

To do so, first note that a simple variant of Fact 3 tells us that the derivative of F is precisely

$$\delta F_\phi = T_{\phi\#}\mu. \tag{20}$$

Therefore the Bregman divergence associated to F can be written

$$F(\phi_2|\phi_1) = \int_\omega \left[-\phi_2^c(x) + \phi_1^c(x) - \phi_2\left(T_{\phi_1}(x)\right) + \phi_1\left(T_{\phi_1}(x)\right)\right]\mu(x)\,dx.$$

The above expression can be simplified using the identity

$$-\phi^c(x) = \phi\left(T_\phi(x)\right) - \frac{1}{2}\left|T_\phi(x) - x\right|^2,$$

which follows immediately from the definitions of the c-transform and of T_ϕ. Consequently,

$$F(\phi_2|\phi_1) = \int_\omega \left[\frac{1}{2}\left|T_{\phi_1}(x) - x\right|^2 - \frac{1}{2}\left|T_{\phi_2}(x) - x\right|^2 + \phi_2\left(T_{\phi_2}(x)\right) - \phi_2\left(T_{\phi_1}(x)\right)\right]$$
$$\mu(x)\,dx.$$

To proceed further we make use of another equality,

$$\nabla\phi(T_\phi(x)) = T_\phi(x) - x, \tag{21}$$

which is again a direct consequence of the c-transform definition. Adding the term $-\big(T_{\phi_2}(x) - x\big)\cdot\big(T_{\phi_1}(x) - T_{\phi_2}(x)\big) + \nabla\phi_2(T_{\phi_2}(x))\cdot\big(T_{\phi_1}(x) - T_{\phi_2}(x)\big)$ (which is thus always 0) to the expression of $F(\phi_2|\phi_1)$ is enough to obtain the desired equality (19).

Step 2. We now establish the inequality

$$F(\phi_2|\phi_1) \le \frac{\lambda^{-1}}{2}\int_\omega |\nabla\phi_2(y) - \nabla\phi_1(y)|^2\,\rho_1(y)\,dy, \tag{22}$$

where $\rho_1 = T_{\phi_1\#}\mu$. Set

$$\xi_2(x) = \frac{1}{2}|x|^2 - \phi_2(x),$$

which is a λ-strongly convex function because of the assumption made on ϕ_2. Then, expression (19) can be written

$$F(\phi_2|\phi_1) = \int_\omega \xi_2\big(T_{\phi_1}(x)\big|T_{\phi_2}(x)\big)\,\mu(x)\,dx.$$

By Lemma 4(iii) we derive the upper bound

$$\xi_2(t_1|t_2) \le \frac{\lambda^{-1}}{2}|\nabla\xi_2(t_1) - \nabla\xi_2(t_2)|^2,$$

for any $t_1, t_2 \in \omega$. Employing again identity (21) we see that on the one hand

$$\nabla\xi_2\big(T_{\phi_1}(x)\big) = x + \nabla\phi_1\big(T_{\phi_1}(x)\big) - \nabla\phi_2\big(T_{\phi_1}(x)\big),$$

and on the other hand $\nabla\xi_2\big(T_{\phi_2}(x)\big) = x$. We deduce the upper bound

$$\xi_2\big(T_{\phi_1}(x)\big|T_{\phi_2}(x)\big) \le \frac{\lambda^{-1}}{2}|\nabla\phi_1\big(T_{\phi_1}(x)\big) - \nabla\phi_2\big(T_{\phi_1}(x)\big)|^2,$$

which after a change of measure implies the desired inequality (22).

Step 3. The next part of the proof consists of estimating the L^∞ norm of ρ_1,

$$\|\rho_1\| \le \lambda^{-d}\|\mu\|. \tag{23}$$

The strong convexity of ϕ_1 implies that the map T_{ϕ_1} is injective. Since ρ_1 is the (density of) the pushforward measure $T_{\phi_1\#}\mu$, we have the change of variable formula

$$\big|\det DT_{\phi_1}(x)\big|\,\rho_1\big(T_{\phi_1}(x)\big) = \mu(x),$$

which is valid for injective and differentiable maps. We now show that DT_{ϕ_1} is a symmetric positive-definite matrix whose eigenvalues are bounded below by λ. To do so, set

$$\xi_1(x) = \frac{1}{2}|x|^2 - \phi_1(x).$$

By the assumption made on ϕ_1 we have $D^2\xi_1(x) \leq \lambda^{-1}$. Making use of identity (21), we can write $x = \nabla\xi_1(T_{\phi_1}(x))$, which can be inverted into $\nabla\xi_1^*(x) = T_{\phi_1}(x)$. Since the Hessian of ξ_1 has eigenvalues bounded above by λ^{-1}, its convex conjugate is λ-strongly convex. Therefore

$$\det DT_{\phi_1}(x) \geq \lambda^d,$$

for any $x \in \omega$. The bound (23) directly follows, and combining it with (22) concludes the proof of (17).

Step 4. Now that the upper bound (17) is proven, it directly implies the lower bound (18). Indeed, rewrite (17) as

$$F(\phi_1 + h|\phi_1) \leq \frac{C}{2}\|h\|_{\dot{H}^1}^2,$$

with $h = \phi_2 - \phi_1$ and $C = \|\mu\|\lambda^{-(d+1)}$. Taking convex conjugation on both sides, which reverses the sign of inequalities, yields

$$F(\phi_1 + \cdot|\phi_1)^*(u) \geq \frac{C^{-1}}{2}\|u\|_{\dot{H}^{-1}}^2.$$

Indeed it is easy to see that $\frac{1}{2}\|\cdot\|_{\dot{H}^1}^2$ and $\frac{1}{2}\|\cdot\|_{\dot{H}^{-1}}^2$ are conjugate to each other. We then apply the Bregman divergence property (iv) from Lemma 4 to the functional F and obtain

$$F^*(\rho_1 + u|\rho_1) \geq \frac{C^{-1}}{2}\|u\|_{\dot{H}^{-1}}^2,$$

where $\rho_1 = \delta F_{\phi_1} = T_{\phi_1\#}\mu$. This is precisely the desired inequality when $u = \rho_2 - \rho_1$.
$\square$

We have now at our disposition all the necessary tools to prove Proposition 1.

Proof of Proposition 1 *Ascent property.* Consider two consecutive iterates ϕ_n and ϕ_{n+1}. By Fact 3 the derivative of J takes the form $\delta J_\phi = \nu - T_{\phi\#}\mu$, which implies that

$$J(\phi_{n+1}) - J(\phi_n) = J(\phi_{n+1}|\phi_n) + \int_\omega (\phi_{n+1} - \phi_n)(\nu - T_{\phi_n\#}\mu).$$

Here we introduced a *Bregman divergence* $J(\cdot|\cdot)$ (see Definition 4). Let F be defined by (15), then as previously noted in (16) we have $J(\phi_{n+1}|\phi_n) = -F(\phi_{n+1}|\phi_n)$. Additionally the gradient step $\phi_{n+1} - \phi_n = \sigma(-\triangle)^{-1}(v - T_{\phi_n \#}\mu)$ implies that

$$\int_\omega (\phi_{n+1} - \phi_n)(v - T_{\phi_n \#}\mu) = \sigma^{-1}\|\phi_{n+1} - \phi_n\|^2_{\dot{H}^1}.$$

Combining these two expressions we arrive at

$$J(\phi_{n+1}) - J(\phi_n) = -F(\phi_{n+1}|\phi_n) + \sigma^{-1}\|\phi_{n+1} - \phi_n\|^2_{\dot{H}^1},$$

and the choice $\sigma = \frac{\lambda^{d+1}}{\|\mu\|}$ combined with Lemma 5 is enough to obtain the ascent property

$$J(\phi_{n+1}) - J(\phi_n) \geq \frac{1}{2}\|\mu\|\lambda^{-(d+1)}\|\phi_{n+1} - \phi_n\|^2_{\dot{H}^1}.$$

Decrease in H^{-1} norm A simple computation, expanding and rearranging quadratic terms, shows that

$$\|\rho_{n+1} - v\|^2_{H^{-1}} - \|\rho_n - v\|^2_{H^{-1}} = \|\rho_{n+1} - \rho_n\|^2_{H^{-1}} - 2\int_\omega (-\triangle)^{-1}(v - \rho_n)(\rho_{n+1} - \rho_n).$$

Since the iterate ϕ_{n+1} is defined by $\phi_{n+1} = \phi_n + \sigma(-\triangle)^{-1}(v - \rho_n)$, we obtain

$$\|\rho_{n+1} - v\|^2_{H^{-1}} - \|\rho_n - v\|^2_{H^{-1}} = \|\rho_{n+1} - \rho_n\|^2_{H^{-1}} - 2\sigma^{-1}\int_\omega (\phi_{n+1} - \phi_n)(\rho_{n+1} - \rho_n).$$

Next, as was explained by (20) the derivative of the functional F defined by (15) is $\delta F_\phi = \rho$, with $\rho = T_{\phi \#}\mu$, and since F is a convex functional this relation can be inverted into

$$\phi = \delta F^*_\rho,$$

where F^* denotes the convex conjugate of F defined by $F^*(\rho) = \sup_\phi \int \phi\,\rho - F(\phi)$. Thus we can write

$$\begin{aligned}
\|\rho_{n+1} - v\|^2_{H^{-1}} &- \|\rho_n - v\|^2_{H^{-1}} \\
&= \|\rho_{n+1} - \rho_n\|^2_{H^{-1}} - 2\sigma^{-1}\left(\delta F^*_{\rho_{n+1}} - \delta F^*_{\rho_n}\right)(\rho_{n+1} - \rho_n) \\
&= \|\rho_{n+1} - \rho_n\|^2_{H^{-1}} - 2\sigma^{-1}\left(F^*(\rho_{n+1}|\rho_n) + F^*(\rho_n|\rho_{n+1})\right) \\
&= -2\sigma^{-1}\left[F^*(\rho_{n+1}|\rho_n) + F^*(\rho_n|\rho_{n+1}) - \frac{\sigma}{2}\|\rho_{n+1} - \rho_n\|^2_{H^{-1}}\right].
\end{aligned}$$

By Lemma 5 we have $F^*(\rho_{n+1}|\rho_n) + F^*(\rho_n|\rho_{n+1}) \geq \sigma\|\rho_{n+1} - \rho_n\|^2_{H^{-1}}$, which is enough to conclude the proof. $\qquad\square$

Proof of Proposition 2 Fix an iterate ϕ_n. Then $J(\phi_n) = D(\phi_n, \phi_n^c)$. By Proposition 1 we have

$$J(\phi_{n+\frac{1}{2}}) \geq J(\phi_n),$$

since the step $\phi_n \to \phi_{n+\frac{1}{2}}$ is a gradient ascent step of J; also note that $J(\phi_{n+\frac{1}{2}}) = D(\phi_{n+\frac{1}{2}}, (\phi_{n+\frac{1}{2}})^c) = D(\phi_{n+\frac{1}{2}}, \psi_{n+\frac{1}{2}})$. Next, it is easy to see that the value of D always increases (or plateaus) when taking c-transforms, thus

$$D\left((\psi_{n+\frac{1}{2}})^c, \psi_{n+\frac{1}{2}}\right) \geq D(\phi_{n+\frac{1}{2}}, \psi_{n+\frac{1}{2}}).$$

Therefore, we have shown that $I(\psi_{n+\frac{1}{2}}) \geq J(\phi_{n+\frac{1}{2}})$; the remaining inequalities can be treated in a similar fashion. $\square$

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 214–223. PMLR (2017). http://proceedings.mlr.press/v70/arjovsky17a.html
2. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pac. J. Math. **16**(1), 1–3 (1966)
3. Benamou, J.D., Brenier, Y.: A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem. Numer. Math. **84**(3), 375–393 (2000). https://doi.org/10.1007/s002110050002
4. Benamou, J.D., Froese, B.D., Oberman, A.M.: Numerical solution of the optimal transportation problem using the Monge–Ampère equation. J. Comput. Phys. **260**, 107–126 (2014). https://doi.org/10.1016/j.jcp.2013.12.015
5. Bertsekas, D.P., Castanon, D.A.: The auction algorithm for the transportation problem. Ann. Oper. Res. **20**(1), 67–96 (1989). https://doi.org/10.1007/BF02216923
6. Brenier, Y.: A combinatorial algorithm for the Euler equations of incompressible flows. In: Proceedings of the Eighth International Conference on Computing Methods in Applied Sciences and Engineering (Versailles, 1987), vol. 75, pp. 325–332 (1989). https://doi.org/10.1016/0045-7825(89)90033-9
7. Brenier, Y.: Polar factorization and monotone rearrangement of vector-valued functions. Comm. Pure Appl. Math. **44**(4), 375–417 (1991). https://doi.org/10.1002/cpa.3160440402
8. Chartrand, R., Wohlberg, B., Vixie, K.R., Bollt, E.M.: A gradient descent solution to the Monge–Kantorovich problem. Appl. Math. Sci. (Ruse) **3**(21–24), 1071–1080 (2009)
9. Corrias, L.: Fast Legendre–Fenchel transform and applications to Hamilton–Jacobi equations and conservation laws. SIAM J. Numer. Anal. **33**(4), 1534–1558 (1996). https://doi.org/10.1137/S0036142993260208
10. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Proceedings of the 26th International Conference on Neural Information Processing Systems—Volume 2, NIPS'13, pp. 2292–2300. Curran Associates Inc., USA (2013). http://dl.acm.org/citation.cfm?id=2999792.2999868
11. Gangbo, W.: An elementary proof of the polar factorization of vector-valued functions. Arch. Rat. Mech. Anal. **128**(4), 381–399 (1994). https://doi.org/10.1007/BF00387715
12. Gangbo, W.: Quelques problèmes d'analyse non convexe, Habilitation à diriger des recherches en mathématiques. Habilitation. Université de Metz, Metz (1995)
13. Gangbo, W., McCann, R.J.: The geometry of optimal transportation. Acta Math. **177**(2), 113–161 (1996). https://doi.org/10.1007/BF02392620
14. Haker, S., Zhu, L., Tannenbaum, A., Angenent, S.: Optimal mass transport for registration and warping. Int. J. Comput. Vis. **60**(3), 225–240 (2004)

15. Huang, M., Malhamé, R.P., Caines, P.E.: Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. Commun. Inf. Syst. **6**(3), 221–251 (2006)
16. Jacobs, M., Kim, I., Mészáros, A.R.: Weak solutions to the Muskat problem with surface tension via optimal transport. arXiv e-prints (2019)
17. Jordan, R., Kinderlehrer, D., Otto, F.: The variational formulation of the Fokker–Planck equation. SIAM J. Math. Anal. **29**(1), 1–17 (1998). https://doi.org/10.1137/S0036141096303359
18. Lasry, J.M., Lions, P.L.: Mean field games. Jpn. J. Math. **2**(1), 229–260 (2007). https://doi.org/10.1007/s11537-007-0657-8
19. Lucet, Y.: Faster than the fast Legendre transform, the linear-time Legendre transform. Numer. Algorithms **16**(2), 171–185 (1997). https://doi.org/10.1023/A:1019191114493
20. Nesterov, Y.: Introductory lectures on convex optimization. In: Applied Optimization, vol. 87. Kluwer Academic Publishers, Boston, MA (2004). https://doi.org/10.1007/978-1-4419-8853-9
21. Otto, F.: The geometry of dissipative evolution equations: the porous medium equation. Comm. Partial Differ. Equ. **26**(1–2), 101–174 (2001). https://doi.org/10.1081/PDE-100002243
22. Santambrogio, F.: Optimal transport for applied mathematicians. *Progress in Nonlinear Differential Equations and their Applications*, vol. 87. Birkhäuser/Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20828-2
23. Villani, C.: Topics in Optimal Transportation, *Graduate Studies in Mathematics*, vol. 58. American Mathematical Society, Providence, RI (2003). https://doi.org/10.1007/b12016