



## **3/4 discrete optimal transport**

Frédéric de Gournay, Jonas Kahn, Léo Lebrat

### **► To cite this version:**

Frédéric de Gournay, Jonas Kahn, Léo Lebrat. 3/4 discrete optimal transport. SIAM Journal on Scientific Computing, 2020, 42 (4), pp.A2088-A2107. hal-01823115

**HAL Id: hal-01823115**

**<https://hal.science/hal-01823115>**

Submitted on 25 Jun 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# $\frac{3}{4}$ -discrete optimal transport

Frédéric de Gournay

Jonas Kahn

Léo Lebrat

June 25, 2018

## Abstract

This paper deals with the  $\frac{3}{4}$ -discrete 2-Wasserstein optimal transport between two measures, where one is supported by a set of segment and the other one is supported by a set of Dirac masses. We select the most suitable optimization procedure that computes the optimal transport and provide numerical examples of approximation of cloud data by segments.

## Introduction

The numerical computation of optimal transport in the sense of the 2-Wasserstein distance has known several breakthroughs in the last decade. One can distinguish three kinds of methods : The first method is based on underlying PDEs [2] and is only available when the measures are absolutely continuous with respect to the Lebesgue measure. The second method deals with discrete measures and is known as the Sinkhorn algorithm [7, 3]. The main idea of this method is to add an entropic regularization in the Kantorovitch formulation. The third method is known as semi-discrete [19, 18, 8] optimal transport and is limited to the case where one measure is atomic and the other is absolutely continuous. This method uses tools of computational geometry, the Laguerre tessellation which is an extension of the Voronoi diagram. The aim of this paper is to develop an efficient method to approximate a discrete measure (a point cloud) by a measure carried by a curve. This requires a novel way to compute an optimal transportation distance between two such measures, none of the methods quoted above comply fully to this framework.

**Pre-existing methods.** The first method, the PDEs formulation, requires the measures to be absolutely continuous, which is not the case.

The Sinkhorn algorithm can be applied to this problem if the curve is sampled by points. But to the best of our knowledge [10], taking Dirac masses along anisotropic "objects", here, a curve may dwindle the efficiency of this algorithm. Indeed, in this particular case the parameter of regularization of the Sinkhorn algorithm has to be chosen smaller than the curve-sampling precision, which causes numerical issues.

The semi-discrete optimal transport is the more favorable setting for such problem. In a previous paper [11] we described method to approximate an image by a curve. The method developed in [11] relies on a sampling of the curve by points and computes the Wasserstein distance between a density and a point cloud. This approach suffers from several drawbacks, the first one is that solving the optimal transport problem is more and more difficult as discretization step decreases. Indeed the closest the Dirac masses are taken along the curve the more stretched the Laguerre tessellation is. The second flaw is the poor quality of approximation of a curve by a sum of Dirac masses. The third problem is the need to transform the discrete measure into a measure with regular density.

Using ideas from the semi-discrete optimal transport, a solution to the above objections is to compute the optimal transport between an absolutely continuous measure and a measure carried by segments. The main tool of such computations is an extension of the Laguerre tessellation with conic boundaries. The difficulty arises when it comes to integrates the continuous density over theses Laguerre cells. In fact, a robust algorithm with exact integration is complex to develop. This is the reason why computer graphic community [16, 15] implements numerical approximations for those Voronoi cells (Laguerre cells with equals weights) using shape primitives and rasterisation

with graphic hardware. However numeric precision is intrinsically tied to grid (pixel). Hence, refinement for this method scales badly when the dimension increases.

Based on these observations we prefer to approximate the measure supported by curve by a measure supported by segments and to compute the optimal transport between an atomic measure and a measure supported on a set of segments. We coined this problem the  $\frac{3}{4}$ -discrete optimal transport.

**Contributions.** This paper provides and studies an efficient algorithm to compute the 2-Wasserstein distance between a discrete measure and a measure supported by a set of segments. This algorithm is scalable in 2D and 3D and is parallelized. We also study the problem of optimizing the parameter of the measure carried by the set of segments and provide the formula of the gradient with respect to the parameter of the measure. The  $\frac{3}{4}$ -discrete optimal transport benefits from the strength of Laguerre tessellation. Moreover it avoids the integration of the density over convex polygons (2D) or polyhedrons (3D), a common difficulty of semi-discrete optimal transport. Indeed the integrations needed for the computation of the cost function boils down to computing intersections between polyhedrons and to integrate the moments of  $\nu$  over segments.

The ideas of this paper owes a much to [8, 19, 18] and the semi-discrete approach but it considers a measure supported by a set of segments instead of a measure with regular density. This modification annihilates the convergence theory developed in [17]. The optimal transport plan is no longer unique and Kantorovitch functional is no longer  $\mathcal{C}^2$ . But under slight geometric condition (see (H)) one can show that the dual functional is  $\mathcal{C}^1$  with respect to the dual variables.

**Outline of the paper.** Section 1 is devoted to set up the notations and the known results used in this paper. In Section 2 we compute the different derivatives of the cost function, in Section 3 we discuss the effective numerical implementation of the algorithm. In Section 4 we settle for the optimization procedure that solves the optimal transport, and in Section 5 we showcase numerical approximations of cloud data by segments.

## 1 Setting

### 1.1 Reminders on 2-Wasserstein distance

The 2-Wasserstein distance is a special instance of the optimal transport distance between two probabilities measures. It is defined as follow:

**Given  $\Omega \subset \mathbb{R}^d$ ,  $\mu \in \mathcal{P}(\Omega)$  and  $\nu \in \mathcal{P}(\Omega)$ , the 2-Wasserstein distance between  $\mu$  and  $\nu$ ,  $W_2(\nu, \mu)$  is given by:**

$$W_2^2(\nu, \mu) = \inf_{\gamma \in \Pi(\nu, \mu)} \int_{\Omega^2} \|x - y\|_2^2 \, d\gamma(x, y), \quad (\text{MK})$$

where  $\Pi(\nu, \mu)$  is the set of coupling between  $\mu$  and  $\nu$ , that is, the set of measures whose marginals are  $\mu$  and  $\nu$  :

$$\gamma \in \Pi(\nu, \mu) \Leftrightarrow \begin{cases} \int_{\Omega} \psi(x) d\nu(x) = \int_{\Omega^2} \psi(x) d\gamma(x, y) & \forall \psi \in L^1(\nu) \\ \text{and} \\ \int_{\Omega} \phi(y) d\mu(y) = \int_{\Omega^2} \phi(y) d\gamma(x, y) & \forall \phi \in L^1(\mu) \end{cases} \quad (1)$$

If both  $(\Omega, \mu)$ ,  $(\Omega, \nu)$  are Polish spaces, an elegant way [25] to solve (MK) is via its dual

$$\begin{aligned} & \sup_{\psi \in L^1(\nu), \phi \in L^1(\mu)} \int_{\Omega} \psi d\nu + \int_{\Omega} \phi d\mu \\ \text{s.t } & \forall (x, y) \in \Omega^2 : \quad \psi(x) + \phi(y) \leq \|x - y\|_2^2, \end{aligned} \quad (2)$$

where  $\phi$  and  $\psi$  are the Lagrange multipliers for (1), the marginals constraints of (MK). Introducing the  $c$ -transform of  $\phi$  as :

$$\phi^c(x) = \inf_{y \in \Omega} \|x - y\|_2^2 - \phi(y).$$

The problem (2) can be rewritten as :

$$\sup_{\phi \in \Phi^c(\Omega)} \int_{\Omega} \phi^c d\nu + \int_{\Omega} \phi d\mu, \quad (3)$$

where  $\Phi^c(\Omega)$  is the set of  $c$ -concave function on  $\Omega$ , see [25]. Consider now the case where the measure  $\mu$  is atomic :  $\mu(\mathbf{x}) = \sum_{i=1}^n m_i \delta_{\mathbf{x}_i}$ ,  $\mathbf{x}_i \in \mathbb{R}^d$ ,  $m_i \in \mathbb{R}$ . In this case  $\phi$  belongs to  $\mathbb{R}^n$ , and  $\phi^c(x) = \min_{i \in \llbracket 1, n \rrbracket} \|x - \mathbf{x}_i\|_2^2 - \phi_i$ . This naturally leads to the definition of the  $i$ -th Laguerre cell [1] :

$$\mathcal{L}_i(\mathbf{x}, \phi) = \{x \in \Omega \text{ such that } \|x - \mathbf{x}_i\|_2^2 - \phi_i \leq \|x - \mathbf{x}_j\|_2^2 - \phi_j \quad \forall j \in \llbracket 1, n \rrbracket\}. \quad (4)$$

Provided that  $\nu(\mathcal{L}_i \cap \mathcal{L}_j) = 0$  for every  $i \neq j$ , the final problem states as :

$$W_2^2(\mu, \nu) = \sup_{\phi \in \mathbb{R}^n} g(\phi, \mathbf{x}), \text{ with } g(\phi, \mathbf{x}) = \sum_i \int_{\mathcal{L}_i(\mathbf{x}, \phi)} (\|x - \mathbf{x}_i\|_2^2 - \phi_i) d\nu + \sum_i m_i \phi_i. \quad (5)$$

Denoting  $\phi^*$  as the solution of maximization problem (5) the physical interpretation of the Laguerre cell  $\mathcal{L}_i(\mathbf{x}, \phi^*)$  is that the Dirac mass located at  $\mathbf{x}_i$  is transported to  $\text{supp}(\nu) \cap \mathcal{L}_i$ . There is equality between problem (MK) and problem (5) since the quadratic cost is continuous and  $(\Omega, \nu), (\Omega, \mu)$  are two Polish spaces [25].

## 1.2 Setting

In this paper  $\mu$  will denote a  $n$ -atomic probability measure on  $\mathbb{R}^d$  :

$$\mu(\mathbf{x}) = \sum_{i=1}^n m_i \delta_{\mathbf{x}_i}, \text{ s.t. } \sum_i m_i = 1 \text{ with } \mathbf{x}_i \in \mathbb{R}^d, m_i \in \mathbb{R}^+, \text{ for all } 1 \leq i \leq n.$$

By contrast,  $\nu$  will be a probability measure supported by a polyline,

$$\nu = \sum_{\alpha=1}^p \rho_{\alpha} l_{\#}^{\alpha} \lambda_{[0,1]} \quad \text{s.t.} \quad l^{\alpha} : \begin{array}{ccc} [0, 1] & \rightarrow & \mathbb{R}^d \\ t & \mapsto & (1-t)P_{\alpha} + tP_{\alpha+1} \end{array}, \rho_{\alpha} \in \mathbb{R}^+, P_{\alpha} \in \mathbb{R}^d, \quad (6)$$

where the measure  $l_{\#}^{\alpha} \lambda_{[0,1]}$  is the push-forward through  $l^{\alpha}$  of  $\lambda_{[0,1]}$ , the  $[0, 1]$ -Lebesgue measure. The measure  $l_{\#}^{\alpha} \lambda_{[0,1]}$  is defined as :

$$l_{\#}^{\alpha} \lambda_{[0,1]}(A) := \lambda_{[0,1]}((l^{\alpha})^{-1}(A)) \text{ for every Borelian set } A.$$

The fact that  $\nu$  is a probability measure translates into  $\sum_{\alpha} \rho_{\alpha} = 1$ . Notice that  $\rho_{\alpha} = 0$  is admissible and in this case the support of the resulting probability  $\nu$  will be composed of disjoint polylines.

In order to ensure the regularity of function  $g$  defined in (5) we enforce hypothesis (H) throughout the paper.

$$\forall \alpha \in \llbracket 1, p \rrbracket, \quad \forall (i, j) \in \llbracket 1, n \rrbracket^2, \quad i \neq j : \quad \langle P_{\alpha+1} - P_{\alpha}, \mathbf{x}_i - \mathbf{x}_j \rangle \neq 0. \quad (\text{H})$$

Under (H), for all  $\phi$  we have  $\nu(\mathcal{L}_i(\mathbf{x}, \phi) \cap \mathcal{L}_j(\mathbf{x}, \phi)) = 0$ . Indeed if  $\mathcal{L}_i$  and  $\mathcal{L}_j$  are two Laguerre cells with a common boundary  $\mathcal{L}_i \cap \mathcal{L}_j$ , the boundary has to be orthogonal to  $\mathbf{x}_i - \mathbf{x}_j$ . Hypothesis (H) prevents tiny perturbation of  $\phi$  to harshly shift the affectation of the segment's mass from one Dirac mass to another, see Figure 1. Under this assumption  $g$  is a  $\mathcal{C}^1$  function of  $\phi$ , see [9].

Let  $\mathcal{I}$  be the set of indices  $(i, \alpha)$  such that the  $i^{\text{th}}$  Laguerre cell meets the  $\alpha^{\text{th}}$  segment, that is:

$$\mathcal{I} = \{(i, \alpha) \text{ s.t. } \mathcal{L}_i(\mathbf{x}, \phi) \cap [P_{\alpha}, P_{\alpha+1}] \neq \emptyset\}.$$



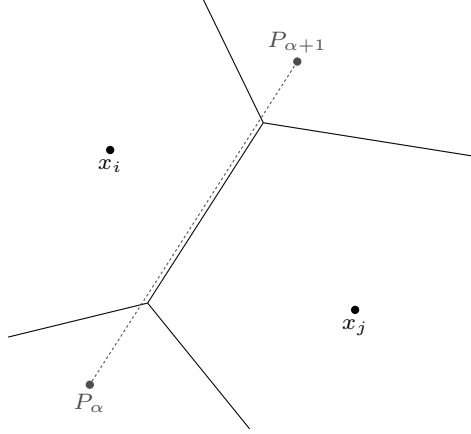


Figure 1: When the hypothesis (H) is violated one can cook up a  $\phi$  such that the functional  $g$  in (5) is no longer  $\mathcal{C}^1$  with respect to  $\phi$ . Here a small perturbation on  $\phi_i$  or  $\phi_j$  induces a discontinuity on  $\nabla_\phi g$ .

If the space  $\Omega$  is convex then the Laguerre cells defined in (4) are convex and  $\mathcal{L}_i \cap [P_\alpha, P_{\alpha+1}]$  is a segment. For every  $(i, \alpha) \in \mathcal{I}$  denote  $t_s^{i\alpha}, t_e^{i\alpha}$  the starting (respectively ending) time of the  $\alpha^{\text{th}}$  segment  $[P_\alpha, P_{\alpha+1}]$  in the  $i^{\text{th}}$  Laguerre cell :

$$0 \leq t_s^{i\alpha} \leq t_e^{i\alpha} \leq 1 \text{ and } \mathcal{L}_i \cap [P_\alpha, P_{\alpha+1}] = [l^\alpha(t_s^{i\alpha}), l^\alpha(t_e^{i\alpha})]. \quad (7)$$

Then the cost function  $g(\phi, \mathbf{x})$  defined in (5) can be re-written as :

$$g(\phi, \mathbf{x}) = \sum_{(i, \alpha) \in \mathcal{I}} \int_{t_s^{i\alpha}}^{t_e^{i\alpha}} r^{i\alpha}(t) dt + \sum_{i=1}^n \phi_i m_i \quad (8)$$

with  $r^{i\alpha}(t) = (\|l^\alpha(t) - \mathbf{x}_i\|_2^2 - \phi_i) \rho_\alpha$

## 2 Derivatives of the cost function

### 2.1 First order derivative with respect to $\phi$

Hypothesis (H) ensures that  $\nu(\mathcal{L}_i(\mathbf{x}, \phi) \cap \mathcal{L}_j(\mathbf{x}, \phi)) = 0, \forall i \neq j$ . A direct application of [9] shows that  $g$  is differentiable with respect to  $\phi$  and :

$$\frac{\partial g}{\partial \phi_i}(\phi, \mathbf{x}) = m_i - \int_{\mathcal{L}_i(\mathbf{x}, \phi)} d\nu. \quad (9)$$

### 2.2 Computation of the second order derivative with respect to $\phi$

If there exists a segment  $[P_\alpha, P_{\alpha+1}]$  that passes through the intersection of at least three Laguerre cells then the functional  $g$  fails to be twice differentiable. On the one hand it is impossible to design an hypothesis in the spirit of hypothesis (H) that can prevent such a pathological case to happen during the optimization in  $\phi$ . On the other hand such a baneful case almost surely never happens. Hence in this section the Hessian is computed without proof of existence by following a cumbersome calculus.

Denote  $(e_i)_i$  the canonical basis. Following the computation of the first derivative given in (9), the second order derivative is given by :

$$\frac{\partial^2 g}{\partial \phi_i \partial \phi_j} = - \lim_{\varepsilon \rightarrow 0} \frac{\int_{\mathcal{L}_i(\mathbf{x}, \phi + \varepsilon e_j)} d\nu - \int_{\mathcal{L}_i(\mathbf{x}, \phi)} d\nu}{\varepsilon}. \quad (10)$$

Denote  $n_{il}$  the outer normal of  $\mathcal{L}_i$  on the facet  $\mathcal{L}_i \cap \mathcal{L}_l$ , that is  $n_{il} = \frac{x_l - x_i}{\|x_l - x_i\|}$ . Denote by  $\delta_{il}$  the first order approximation of the evolution of the facet  $\mathcal{L}_i \cap \mathcal{L}_l$  in the normal direction  $n_{il}$  when we change the Lagrange multiplier of the  $j$ -th cell, that is, when we change  $\phi$  into  $\phi + \varepsilon e_j$ , see Figure 2 (left). If both  $i$  and  $l$  are different from  $j$  then  $\delta_{il} = 0$  otherwise  $\delta_{ij}$  is given by :

$$\begin{aligned} y + \varepsilon \delta_{ij} n_{ij} + o(\varepsilon) &\in (\mathcal{L}_i \cap \mathcal{L}_j)(\mathbf{x}, \phi + \varepsilon e_j) \\ \Leftrightarrow \|y + \varepsilon \delta_{ij} n_{ij} - \mathbf{x}_j\|_2^2 - \phi_j - \varepsilon + o(\varepsilon) &= \|y + \varepsilon \delta_{ij} n_{ij} - \mathbf{x}_i\|_2^2 - \phi_i \\ \Rightarrow \delta_{ij} &= \frac{1}{2\|\mathbf{x}_j - \mathbf{x}_i\|}. \end{aligned} \quad (11)$$

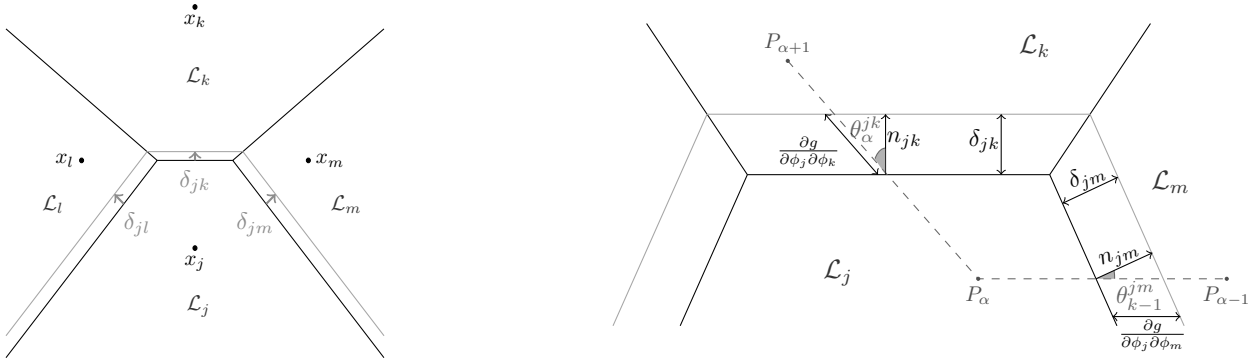


Figure 2: (Left) Normal displacements ( $\delta_{il}, \delta_{ik}, \delta_{im}$ ) of the boundary of the Laguerre cell when  $\phi_i$  increases and  $(\phi_l, \phi_k, \phi_m)$  remain constant. (Right) geometrical interpretation of the Hessian

Denote  $\Theta_{ij}$  the indices of the segments that intersect the facet  $\mathcal{L}_i \cap \mathcal{L}_j$ , it is given by

$$\Theta_{ij} = \{\alpha \text{ such that, } (i, \alpha) \in \mathcal{I} \text{ and } (j, \alpha) \in \mathcal{I}\}.$$

For every  $\alpha$  in  $\Theta_{ij}$  denote  $\theta_\alpha^{ij}$  the angle between the segment  $[P_\alpha, P_{\alpha+1}]$  and the outer normal  $n_{ij}$ . See Figure 2 (right), simple geometrical consideration shows that the Hessian formula boils down to :

$$\frac{\partial g}{\partial \phi_i \partial \phi_j}(\phi, \mathbf{x}) = \begin{cases} - \sum_{\theta \in \Theta_{ij}} \frac{\rho_\alpha}{2\|\mathbf{x}_j - \mathbf{x}_i\| \cos(\theta_\alpha^{ij})} & \text{if } i \neq j, \\ - \sum_{k \neq i} \frac{\partial g}{\partial \phi_i \partial \phi_k} & \text{otherwise.} \end{cases} \quad (12)$$

### 2.3 Computation of the first order derivative with respect to $P$ and $\rho$

In this section we compute the derivatives of the 2-Wasserstein distance with respect to the parameters of the measure  $\nu$ . We denote generically as  $\partial_\nu$  the derivative with respect to one of the parameters that define  $\nu$ , that is either one of the positions  $P_\alpha$  or one of the densities  $\rho_\alpha$ . Let  $\phi^*$  be the optimal Lagrange multiplier in (5)

$$G(\nu) := g(\phi^*, \nu) = W_2^2(\mu, \nu)$$

The derivative of  $g(\phi^*, \nu)$  with respect to the parameters of  $\nu$  are given using the chain rule formula.

$$\frac{dG}{d\nu} = \frac{\partial g}{\partial \phi} \frac{\partial \phi}{\partial \nu} + \frac{\partial g}{\partial \nu} \quad (13)$$

Since  $\phi^*$  is a solution of (5), the derivative of  $g$  with respect to  $\phi$  is zero at  $\phi^*$ . Hence computing the differential of  $G$  consists in differentiating the cost function  $g$  while keeping  $\phi$  fixed at  $\phi^*$ . Differentiating (8) with respect to  $\nu$  we obtain :

$$\partial_\nu G = \sum_{(i,\alpha) \in \mathcal{I}} (\partial_\nu t_e^{i\alpha}) r^{i\alpha}(t_e^{i\alpha}) - (\partial_\nu t_s^{i\alpha}) r^{i\alpha}(t_s^{i\alpha}) + \int_{t_s^{i\alpha}}^{t_e^{i\alpha}} \partial_\nu r^{i\alpha}(t) dt$$

When considering  $\partial_\nu t^{i\alpha}$  only three cases can occur :

1. If  $P_\alpha$  belongs to the interior of the  $i$ -th Laguerre cell then,  $t_s^{i\alpha} = 0$  and  $\partial_\nu t_s^{i\alpha} = 0$ .
2. If  $P_{\alpha+1}$  belongs to the interior of the  $i$ -th Laguerre cell then,  $t_e^{i\alpha} = 1$  and  $\partial_\nu t_e^{i\alpha} = 0$ .
3. For all  $i$  there exists exactly one  $j$  such that  $t_s^{i\alpha} = t_e^{j\alpha}$ . The segment  $[P_\alpha, P_{\alpha+1}]$  intersects  $\mathcal{L}_i \cap \mathcal{L}_j$  at point  $l^\alpha(t_s^{i\alpha})$ . For all  $x$  in  $\mathcal{L}_i \cap \mathcal{L}_j$ , by definition of the Laguerre cell (4), the following equality holds  $\|x - \mathbf{x}_i\|_2^2 - \phi_i = \|x - \mathbf{x}_j\|_2^2 - \phi_j$  then  $r^{i\alpha}(t_s^{i\alpha}) = r^{j\alpha}(t_s^{j\alpha})$ .

It follows that

$$\partial_\nu G = \sum_{(i,\alpha) \in \mathcal{I}} \int_{t_s^{i\alpha}}^{t_e^{i\alpha}} \partial_\nu r^{i\alpha}(t) dt \quad (14)$$

### 2.3.1 Derivative with respect to $P$

We are first interested in differentiating the Wasserstein distance with respect to the position of the endpoints  $(P_\alpha)_{\alpha \in \llbracket 1, p \rrbracket}$  of the polyline. A direct application of (14) yields :

$$\frac{\partial G}{\partial P_\alpha} = \int_{t_s^{i(\alpha-1)}}^{t_e^{i(\alpha-1)}} \partial_{P_\alpha} r^{i(\alpha-1)}(t) dt + \int_{t_s^{i\alpha}}^{t_e^{i\alpha}} \partial_{P_\alpha} r^{i\alpha}(t) dt$$

The differential of  $r^{i\alpha}(t)$  with respect to  $P_\alpha$  amounts to differentiate  $\|l^\alpha(t) - x_i\|^2$ , we recall that :

$$\begin{cases} \frac{\partial \|l^\alpha(t) - x_i\|^2}{\partial P_\alpha} &= 2(1-t)(l^\alpha(t) - x_i) \\ \frac{\partial \|l^{(\alpha-1)}(t) - x_i\|^2}{\partial P_\alpha} &= 2t(l^{(\alpha-1)}(t) - x_i). \end{cases}$$

Let  $\mathcal{K}(\alpha)$  the set of indices  $i$  such that the  $i$ -th Laguerre cell has a non-empty intersection with the  $\alpha$ -th segment:

$$\mathcal{K}(\alpha) = \{i \text{ s.t. } \mathcal{L}_i(x, \phi) \cap [P_\alpha, P_{\alpha+1}] \neq \emptyset\} = \{i \text{ such that } (i, \alpha) \in \mathcal{I}\}.$$

A straightforward computation yields

$$\begin{aligned} \frac{\partial G}{\partial P_\alpha} &= \left( P_\alpha + \frac{P_{\alpha-1} - P_\alpha}{3} - \sum_{i \in \mathcal{K}(\alpha-1)} \int_{t_s^{i(\alpha-1)}}^{t_e^{i(\alpha-1)}} t \mathbf{x}_i dt \right) \rho_{\alpha-1} \\ &\quad + \left( P_\alpha + \frac{P_{\alpha+1} - P_\alpha}{3} - \sum_{i \in \mathcal{K}(\alpha)} \int_{t_s^{i\alpha}}^{\dagger t_e^{i\alpha}} t \mathbf{x}_i dt \right) \rho_\alpha \end{aligned} \quad (15)$$

where  ${}^\dagger t$  is the reverse parameterization of the segment  $[P_\alpha, P_{\alpha+1}]$ , that is

$${}^\dagger t_s^{i\alpha} = 1 - t_e^{i\alpha} \quad \text{and} \quad {}^\dagger t_e^{i\alpha} = 1 - t_s^{i\alpha}.$$

We may interpret (15) as the sum of the torques of the surrounding segments  $[P_{\alpha-1}, P_\alpha]$  and  $[P_\alpha, P_{\alpha+1}]$  around the point  $P_\alpha$ .

### 2.3.2 Derivative with respect to $\rho$

The functional  $g$  (8) is linear in  $\rho_\alpha$ , and its derivative is therefore trivial. Throughout this paper, we consider that the density is constant by segment. As a consequence

$$\rho_\alpha = \frac{\|P_\alpha - P_{\alpha+1}\|}{\sum_\alpha \|P_\alpha - P_{\alpha+1}\|}, \quad (16)$$

hence the derivative of  $\rho$  with respect to  $P$  is easily computed.

### 2.3.3 Optimization algorithm

The aim of this section is to describe the algorithm which optimizes the Wasserstein distance  $G$  with respect to the position of the endpoints  $P_\alpha$  of the polyline. The proposed algorithm is a gradient descent method with a metric  $\Sigma$ , we recall that the gradient of  $G$  with respect to  $P$  is given by :

$$\nabla_P G = \frac{\partial G}{\partial P} + \frac{\partial G}{\partial \rho} \frac{\partial \rho}{\partial P}, \quad (17)$$

The differential of  $G$  with respect to  $P$  and  $\rho$  are discussed in Section 2.3.1 and Section 2.3.2 respectively. The main goal of this section is to discuss the choice of the metric  $\Sigma$ .

Let us consider an isolated segment  $[P_\alpha, P_{\alpha+1}]$ , that is  $\rho_{\alpha-1} = \rho_{\alpha+1} = 0$ . The average of the gradient for the segment  $[P_\alpha, P_{\alpha+1}]$  is given by :

$$\frac{1}{2} \left( \frac{\partial G}{\partial P_\alpha} + \frac{\partial G}{\partial P_{\alpha+1}} \right) = \left( \frac{1}{2} P_\alpha + \frac{1}{2} P_{\alpha+1} - \sum_{i \in \mathcal{K}_\alpha} \int_{t_s^{i\alpha}}^{t_e^{i\alpha}} \mathbf{x}_i dt \right) \rho_\alpha = \rho_\alpha (c_\alpha - \bar{\mathbf{x}}_\alpha) \quad (18)$$

where  $c_\alpha$  is the center of the  $\alpha$ -th segment and  $\bar{\mathbf{x}}_\alpha$  is the weighted average of points  $\mathbf{x}_i$  seen by the  $\alpha$ -th segment :

$$c_\alpha = \frac{P_\alpha + P_{\alpha+1}}{2} \quad \text{and} \quad \bar{\mathbf{x}}_\alpha = \sum_{i \in \mathcal{K}_\alpha} \int_{t_s^{i\alpha}}^{t_e^{i\alpha}} \mathbf{x}_i dt.$$

Equation (18) is reminiscent of the formula of the derivative of  $G$  in the semi-discrete setting, see [18, 8, 19]. In this setting,  $\mu$  is a measure with density and the approximating measure  $\nu$  is a sum of Diracs :

$$\nu = \sum_{\alpha=1}^p m_\alpha \delta_{\mathbf{y}_\alpha}$$

In this case, the derivative of  $G$  with respect to the position of the points is given by :

$$\frac{\partial G}{\partial \mathbf{y}_\alpha} = m_\alpha (\mathbf{y}_\alpha - \mathbf{b}_\alpha), \quad (19)$$

where  $\mathbf{b}_\alpha = \frac{1}{m_\alpha} \int_{\mathcal{L}_\alpha} x d\mu$  is the barycenter of the  $\alpha$ th Laguerre cell.

In the semi-discrete setting, the most commonly used algorithm when minimizing  $G$  with respect to  $\mathbf{y}$  is to update the point position  $\mathbf{y}_\alpha$  to their barycenters  $\mathbf{b}_\alpha$ . This procedure is known as Lloyd's algorithm [8, 19, 18]. In view of the formula of the gradient (19), this method is a gradient descent with metric  $\Sigma = \text{diag}(m_\alpha)$ . In our case it is natural to consider a metric  $\Sigma$  defined via  $\rho_\alpha$  as an analogy to the semi-discrete setting. We define :

$$\Sigma = \text{diag}\left(\frac{\rho_{\alpha-1} + \rho_\alpha}{2}\right). \quad (20)$$

The main algorithm that minimizes  $G$  with respect to  $P$  is given by algorithm 1 below

---

**Algorithm 1** Optimization polyline position

---

```
1: procedure OPTIMIZATION IN P
2:    $\nabla_P G \leftarrow \mathbf{0}$ 
3:   do
4:      $P \leftarrow P - \Sigma^{-1} \cdot \nabla_P G$  ▷ with  $\Sigma$  defined in (20),  $\nabla_P G$  defined in (17)
5:      $\rho \leftarrow \text{normalizedDensity}(P)$  ▷ as in (16)
6:      $\phi^*, \nabla_P G \leftarrow \text{computeOptimalTransport}(\mathbf{x}, m, P, \rho)$  ▷ see Algorithm 2
7:   while  $\|\nabla P\|_\infty \geq 10^{-3}$ 
8: end procedure
```

---

### 3 Numerical implementation

The Laguerre cells are computed using the computational geometry library CGAL [6]. In 2D the algorithm is fast and scalable, the average complexity for  $n$  sites randomly drawn is linear in time and memory. For 3D triangulation the worst case complexity is quadratic, but for random point configurations the complexity is observed to be almost linear [12]. In this section we discuss the computation of  $g$  and its parallelization.

#### 3.1 Integration computation

The main issue when computing  $g$  and its derivative is to evaluate the intersections (7) between the polylines and the Laguerre cells. The intersections are computed by following each segment  $l^\alpha(t)$  with  $t$  increasing. We first focus our attention on computing the exit time of the Laguerre cell  $j$  knowing the starting time  $t_s^\alpha$ . To clarify things suppose that at the known time  $t_e^\alpha = t_s^\alpha$  the segment exits the Laguerre cell  $\mathcal{L}_i$  and enters the Laguerre cell  $\mathcal{L}_j$ . The objective is to compute the time  $t_e^\alpha = t_s^\alpha$  and the index  $k$  such that the segment exits the Laguerre cell  $\mathcal{L}_j$  at time  $t_e^\alpha$  and enters the Laguerre cell  $\mathcal{L}_k$ . Such a time is computed by solving the following minimization problem

$$k = \underset{m \text{ s.t. } m \neq i, t_s^\alpha < t_m \leq 1}{\operatorname{argmin}} t_m, \quad \text{with} \quad t_m = \frac{2\langle P_\alpha, \mathbf{x}_j - \mathbf{x}_m \rangle + \|\mathbf{x}_m\|^2 - \phi_m + \phi_j - \|\mathbf{x}_j\|_2^2}{2\langle P_\alpha - P_{\alpha+1}, \mathbf{x}_j - \mathbf{x}_m \rangle}. \quad (21)$$

The exit time  $t_e^\alpha$  is now equal to  $t_k$ . Note that the choice of  $m$  can be restricted to the indexes of adjacent Laguerre cells, which is a small set in practice (a dozen of indexes). If the set  $\{m \neq i, t_s^\alpha < t_m \leq 1\}$  is empty, the segment ends in the Laguerre cell  $\mathcal{L}_j$  and we set  $t_e^\alpha = 1$  and we stop the procedure for the segment.

In the case when  $t_s^\alpha = 0$ , that is the segment starts in the Laguerre cell  $\mathcal{L}_j$ , we consider the same minimization problem as (21) where the constraint set is replaced with  $\{m, 0 \leq t_m \leq 1\}$ .

We still have to compute the index of the Laguerre cell where the segment  $[P_\alpha, P_{\alpha+1}]$  begins. If  $\alpha > 0$ , it is obviously the index of the ending Laguerre cell of  $[P_{\alpha-1}, P_\alpha]$ . For the case  $\alpha = 0$ , we add a dummy segment  $[\mathbf{x}_l, P_\alpha]$ , where  $\mathbf{x}_l$  is the position of the  $l$ -th Dirac mass corresponding to the largest multiplier  $\phi_l = \max_k \phi_k$ . By definition of the Laguerre diagram (4) the point  $\mathbf{x}_l$  belongs to  $\mathcal{L}_l$ .

Given the starting and ending times, the computation of  $g(\phi, \mathbf{x})$  and its derivative abridge to integrate polynomials within  $[t_s^\alpha, t_e^\alpha]$  using Gaussian quadrature.

There might exist several solutions  $k$  to the minimization problem (21). In this case, the segment encounters a corner, the intersection of at least 3 Laguerre cells. In practice this case never occurs thanks to floating point arithmetic but the rounding error can elect a non-suitable candidate. In this case, the algorithm assigns a segment of negligible length to the candidate, see Figure 3.

#### 3.2 Parallelism

The evaluation of  $g(\phi, \mathbf{x})$  and its derivative with respect to  $\phi$  are discussed in Section 3.1. For each segment of the polyline the intersection times are computed sequentially by (21) and the integration is performed on the fly. The

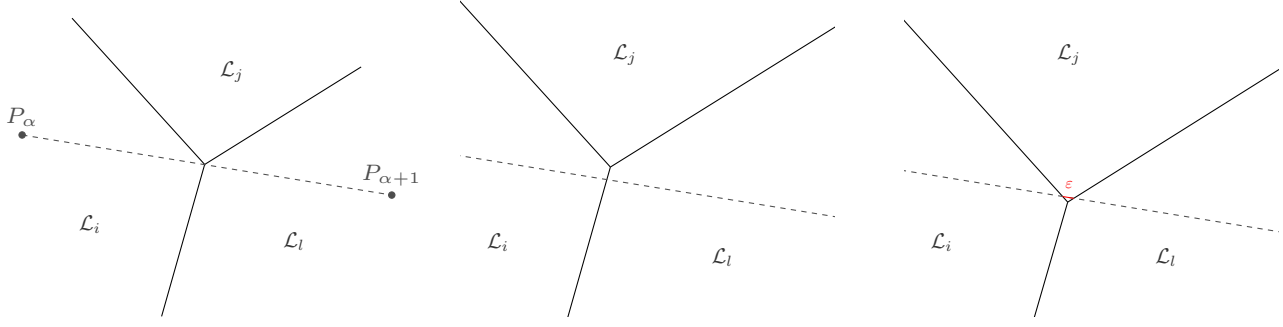


Figure 3: Example of non-uniqueness to the minimization problem (21). At the left the polyline  $[P_\alpha, P_{\alpha+1}]$  intersects the junction of 3 Laguerre cells  $(i, j, l)$ . In the center  $k = l$  chosen as expected. At the right  $k = j$  chosen and  $t_e^{j\alpha} - t_s^{j\alpha} = \varepsilon$ , a small error occurs assigning a bit of the segment to the  $j$ th Laguerre cell.

parallelism is unfurled at the highest level of the algorithm for the segments. Remember that the computation of the intersection of the polyline and the Laguerre tessellation requires for each segment the knowledge of the cell of its starting point. This cell can be inferred from the calculation of the previous segment. Hence, we slice the polyline into contiguous chunks of segments with equivalent size, each worker dealing with one chunk. For each of these chunks the sequential integration procedure is launched and then the results are merged. We carry out both OpenMP and `c++11` thread class implementation. For a standard chunk size and for a regular processor the `c++11` performance overtakes the OpenMP's one. We suppose that this difference of performance (up to a factor two) is due to OpenMP overhead.

In Figure 4, the evolution of the computation time of the cost function  $g$  and its gradient with respect to  $\phi$  is benchmarked. The speedup unit is defined as the ratio of the execution time of the non-parallelized task over the parallelized task. OpenMP and `c++11` thread class are both implementations of shared memory parallelization, the memory is simultaneously accessible for every thread. The performance of a process depends intrinsically on how close the data is. If the data is on processor cache the latency is a dozens of CPU cycles, if the data is located on RAM, the latency is  $40ns$ . In our experiment, for large number of threads, 30% of the data was not located on cache. This explains why performance drops as the number of threads increases.

## 4 Computation of the optimal transport

The goal of this section is to compare the available methods to optimize  $g$  with respect to  $\phi$  (5). The functional  $g$  is concave with respect to  $\phi$  since it is a dual formulation of the problem (MK) see [25]. Under the hypothesis (H) the functional  $g$  is  $\mathcal{C}^1$  with respect to  $\phi$ . Note that a higher level of regularity cannot be established as argued in Figure 5. This lack of regularity precludes the use of the convergence framework for second order methods established by Mériçot *et al.* [17].

However, we may apply classical convergence results of concave analysis to functional the  $g$ . To trigger those arguments, the Lipschitz constant of the gradient  $L$  has to be bounded. Under Hypothesis (H), using the notation of Section 2.2 summarized in Figure 2 and the expression of the Hessian matrix (12), we establish a finite upper bound for  $L$  using the Gershgorin circle theorem :

$$|L| \leq 2 \max_i \sum_{j \neq i} \left| \frac{\partial^2 g}{\partial \phi_i \partial \phi_j} \right| \leq \frac{p^2 \max_\alpha (\rho_\alpha)}{\min_{l \neq m} (\|\mathbf{x}_l - \mathbf{x}_m\|_2) \min_{i, j, \alpha, i \neq j} |\cos((\theta_\alpha^{ij}))|}, \quad (22)$$

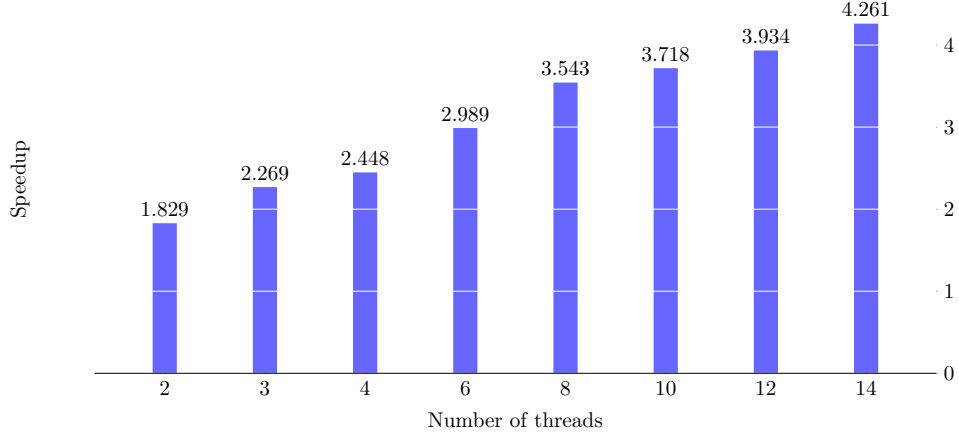


Figure 4: Speedup for the computation of the cost function  $g$  with `c++11` thread class. The parallelism is placed over the segment integration for  $n = 200K$  points and  $p = 80K$  segments for an increasing number of threads with a super-calculator equipped with Intel Xeon<sup>®</sup> E5-2680

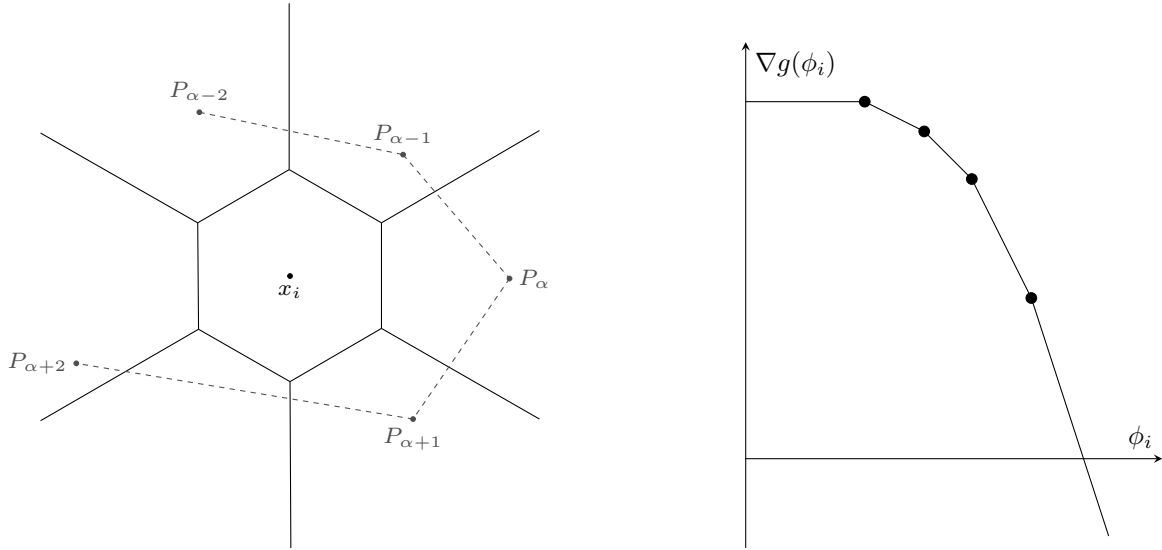


Figure 5: Counter-example to the smoothness of  $\nabla g$ . Consider the a Laguerre tessellation and a set of segment as displayed in the left and increase  $\phi_i$ . The Laguerre cell  $\mathcal{L}_i$  increases and the gradient of  $g$  with respect to  $\phi$  exhibits kinks. Those kinks happen each time a Laguerre cell meets a segment for the first time. Note that under Hypothesis (H), the boundary of the Laguerre cell can only cross transversally the segment. If Hypothesis (H) is not met  $\nabla g$  may fail to exist.

where  $p$  is the number of segments composing the polyline,  $\rho_\alpha$  is the density associated to each segment and  $\theta_\alpha^{ij}$  is the angle between the vector  $P_{\alpha+1} - P_\alpha$  and  $(\mathbf{x}_j - \mathbf{x}_i)$ . For fixed measures  $\nu$  and  $\mu$  the evaluation of the bound (22) is costly as it involves combinatorial quantities and is not evaluated in practice.

The gradient Lipschitz condition (22) is sufficient to ensure convergence of ascent methods. Note that  $g$  is bounded from above as  $\mu$  and  $\nu$  are compactly supported measures. The gradient method with step  $s < \frac{2}{L}$  converges to a stationary point  $\phi^*$  see [21, 4, 20]. The same holds for variant step-size method with line search such as Global

Brazalai Borwein algorithm [22, 13].

Because of the counter-example in Figure 5, and the lack of regularity of the gradient of  $g$ , quadratic convergence cannot be guaranteed for second order methods. Note also that in a generic setting, the Newton method is impractical since the Hessian matrix defined in equation (12) fails to be invertible, see Section 4.1.3.

## 4.1 Choice of optimization method

The different methods are tested against the same benchmark. It consists in drawing uniformly 10K points and 500 segments in 2D. Several realization of the optimization procedure are performed and plotted in the corresponding figures. The methods are benchmarked in Table 1.

### 4.1.1 First order method

We first implement a gradient ascent method. The generic convergence history is displayed in Figure 6. As explained in Section 4, the Lipschitz constant of the gradient cannot be satisfactorily computed, so the step-size is chosen according to strong Wolfe conditions. In practice the algorithm settles for a constant step of 0.05. As it can be observed in Figure 6 the rate of convergence of the gradient towards zero is too slow for this method to be used in practice.

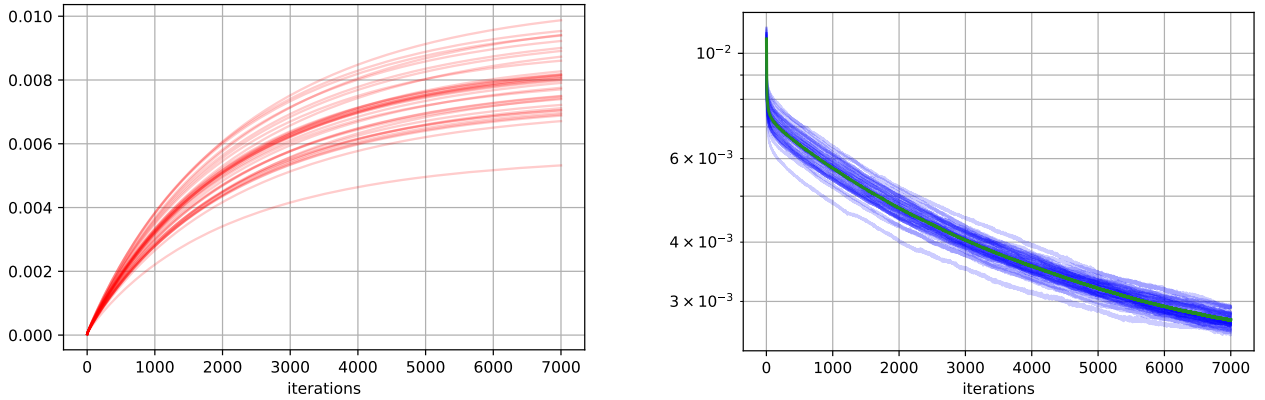


Figure 6: Gradient ascent method : cost function history (left) and norm of the gradient (right). Method stopped by the max iteration criterion.

We have tried other first order methods including Polak-Ribire, Fletcher-Reeves, Barzalai Borwein and Nesterov acceleration. Only Nesterov acceleration has behaved differently from the other methods and it is displayed in Figure 7. Note however, that Nesterov acceleration requires an estimation of  $L$ , the Lipschitz constant of the gradient. Figure 7 was obtained with an estimation of  $L$  that promotes quick decreases of gradient norm at the beginning of the algorithm. Note however that the estimation of  $L$  seems too optimistic since the Nesterov algorithm does not converge. A more pessimistic choice of  $L$  leads to a convergence rate similar to the gradient algorithm, at least during the first 7K iterations.

### 4.1.2 Quasi-Newton Method

The main idea behind quasi-Newton methods is to build at the iteration  $k$  an approximation of the Hessian matrix of  $g$ . We choose the limited memory BFGS method, which only stores a limited amount of vectors determined by the user. The result is displayed in Figure 8. This method converges faster than first order method, the L-BFGS



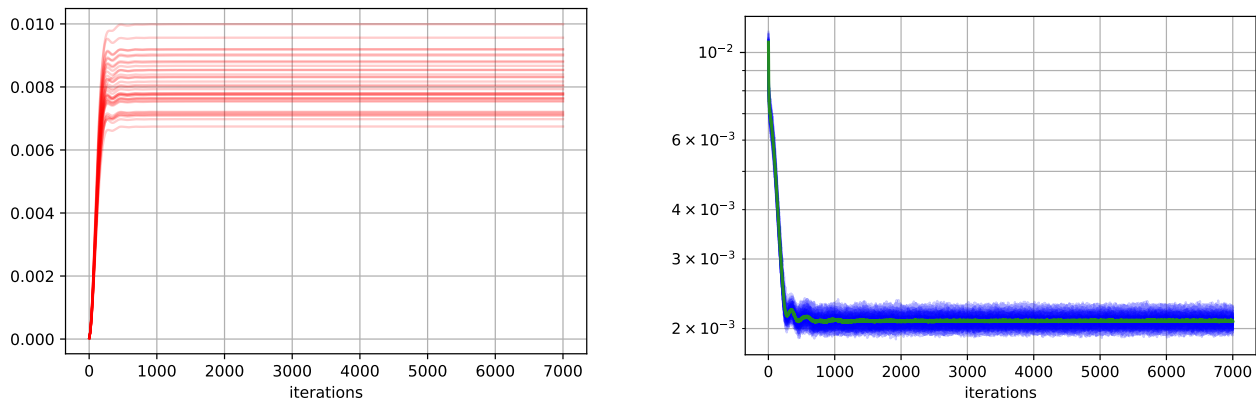


Figure 7: Nesterov acceleration of gradient ascent : cost function history (left) and norm of the gradient (right). Method stopped by the max iteration criterion.

algorithm reaches the desired gradient tolerance  $5.10^{-5}$  within 1500 iterations. By contrast first order methods failed to converge in 7000 iterations. However when the method is close to a critical point the convergence speed tends to decrease.

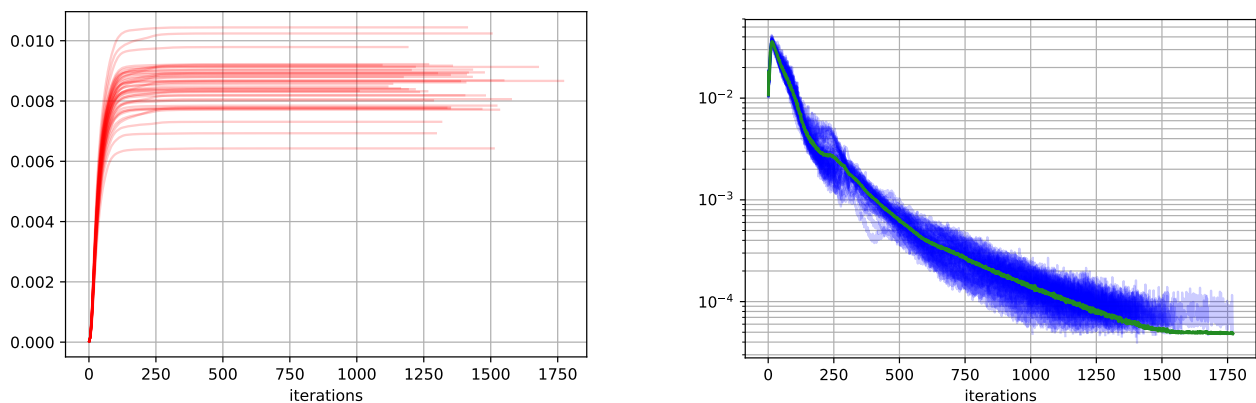


Figure 8: L-BFGS method : cost function history (left) and norm of the gradient (right). Method converges to the targeted gradient norm  $5.10^{-5}$  in a average of 1500 iterations.

#### 4.1.3 Newton and quasi-Newton hybridization

In the light of the previous section, the flaw of the quasi-Newton method is its lack of briskness when it reaches the vicinity of a maximizer. Second-order methods are known to converge quadratically in the basin of attraction of a maximizer. The idea of the hybrid algorithm is to start with a quasi-Newton method, and then switch to a second order Newton algorithm when the basin of attraction is reached. The quandary is to determine when to start the Newton method in preference to the L-BFGS method. Indeed far from the basin of attraction the direction given by

the Newton method is poor and the natural step 1 is truncated by the line search which causes extra computation time.

We propose to switch from the L-BFGS algorithm to the Newton algorithm as soon as there is no empty Laguerre cell. A Laguerre cell  $\mathcal{L}_i$  is called empty as long as there is no mass assigned to its centroid  $\mathbf{x}_i$  that is  $\nu(\mathcal{L}_i) = 0$ . This criterion is informally motivated by the fact that the second order information relates the competition and the connectivity between Laguerre cells, hence if one of them is empty its information is inconsequential. From a practical point of view, this condition ensures the Hessian on being a full rank matrix and hence invertible. This procedure is described in Algorithm 2 and the result are displayed in Figure 9.

---

**Algorithm 2** Computation of optimal transport

---

**Input:** Dirac positions  $\mathbf{x}$  and masses  $m$ .

**Input:** Nodes  $P$  and density  $\rho$  of the polyline.

**Require:**  $\phi_{\text{init}}$  a starting  $\phi$  for the computation of  $g$

**Require:** gradTol L2 tolerance on the gradient norm

**Require:** outerMax maximum number of iterations

**Require:** wolfeProcedure standard line search with strong Wolfe condition and initial step  $s = 1$ . [4]

```

1: function COMPUTEOPTIMALTRANSPORT( $\mathbf{x}, m, P, \rho$ )
2:   bf  $\leftarrow$  LBFGS(memSize) ▷ Initialization of L-BFGS
3:    $\phi \leftarrow \phi_{\text{init}}$ 
4:    $\nabla\phi, \text{cost}, \text{hiddenNumber} \leftarrow \text{computeIntegration}(\phi)$ 
5:    $i \leftarrow 0$ 
6:   while  $i < \text{outerMax}$  &  $\text{gradTol} < \|\nabla\phi\|_2$  do
7:     if  $\text{hiddenNumber} \neq 0$  then
8:        $d \leftarrow \text{bf.findDirection}(\nabla\phi)$ 
9:     else
10:       $\mathcal{H} \leftarrow \text{computeHessian}(\phi)$ 
11:       $d \leftarrow -\mathcal{H}^{-1}\nabla\phi$  ▷ The Hessian is definite so Newton direction is taken
12:    end if
13:     $s \leftarrow \text{wolfeProcedure}(\text{cost}, \phi, s, d)$ 
14:     $\nabla\phi_{\text{old}} \leftarrow \nabla\phi$ 
15:     $\nabla\phi, \text{cost}, \text{hiddenNumber} \leftarrow \text{computeIntegration}(\phi + sd)$ 
16:    bf.addDirection( $\nabla\phi_{\text{old}} - \nabla\phi, sd$ ) ▷ Actualize memory of the L-BFGS algorithm
17:     $\phi \leftarrow \phi + sd$ 
18:  end while
19: end function

```

---

In the numerical tests, the switch between the two Newton method happens approximatively 30 iterations before termination of the algorithm. During the Newton phase of the hybrid algorithm, the algorithm undergoes two stages. During the twenty first iteration of the Newton method the algorithm stabilizes around the maximum and the Wolfe line-search prevents picking  $s = 1$ . During the second stage, quadratic convergence is achieved in few iterations.

Levenberg-Marquardt's method was also implemented with a regularization parameter that tends to zero when approaching the basin of convergence, however in large scale optimization problems the lightweight hybrid method is preferable since it requires the inversion of the Hessian only in the last iterations.

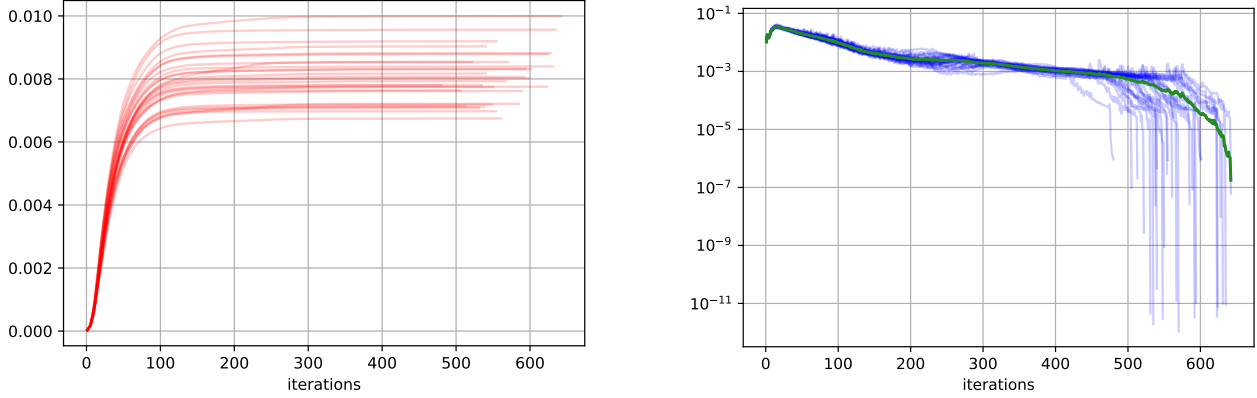


Figure 9: Hybrid Newton method : cost function history (left) and norm of the gradient (right). The method converges to the targeted gradient norm  $10^{-6}$  in 600 iterations on average.

	BBG	Nesterov	BFGS	BFGS/Newt	LM
Time/iteration	36.4ms	36.7ms	49.4ms	186ms	649ms
gradient norm	$5.5 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$	$1.5 \cdot 10^{-4}$	$10^{-15}$	$5.4 \cdot 10^{-5}$

Table 1: Comparison of algorithms for solving the optimal transport problem for  $10^4$  points and 500 lines. Time required for an iteration in millisecond and the gradient norm after 1000 iterations. The BFGS/Newton method converges up to numerical error.

## 5 Numerical examples

### 5.1 Representation of picture

The approximation of a measure by a curve has multiple applications of which some are described in [11]. In this section we discuss the representation of a picture by a polyline. The first step is the discretization of the picture, here a landscape, by a sum of Dirac masses of weights  $m$ . Several approaches are possible but the most intuitive is to take the Dirac positions  $\mathbf{x}$  on a Cartesian grid and their weights equal to the pixels intensities. The polyline is then initialized randomly, and Algorithm 1 is launched. With no speed or curvature constraints, the gradient method in  $P$  described in Algorithm 1 empirically gives a stationary point. This solution is highly non-smooth, the length of the segments and the angle between them are disparate see Figure 10 top right.

This scribbled solution is, in practice, difficult to carry out with laser engravers. Indeed for the same number of segments  $p$ , the engraving time of uncontrolled trajectories can be up to five times longer. In order to get around this problem, we follow the method described in [11], that is projecting the polyline after the gradient step of Algorithm 1 on a set of kinematic constraints. The constraints imposed on the speed and the acceleration polyline read as :

$$\mathcal{K}(K_1, K_2) = \left\{ (P_\alpha)_{\alpha \in \llbracket 1, p \rrbracket} \text{ s.t } \begin{cases} \|P_{\alpha+1} - P_\alpha\| \leq K_1, & \forall \alpha \in \llbracket 1, p-1 \rrbracket \\ \|2P_\alpha - P_{\alpha-1} - P_{\alpha+1}\| \leq K_2, & \forall \alpha \in \llbracket 2, p-1 \rrbracket \end{cases} \right\}, \quad (23)$$

with  $K_1$  the constant controlling the speed of the polyline and  $K_2$  its acceleration. The projection on the set  $\mathcal{K}(K_1, K_2)$  is performed using the Alternating Direction Method of Multipliers (ADMM) [14].

The authors wish to thank Andrew Gibson for letting them use the original picture and Alban Gossard for his priceless help in realizing the wood engraving.

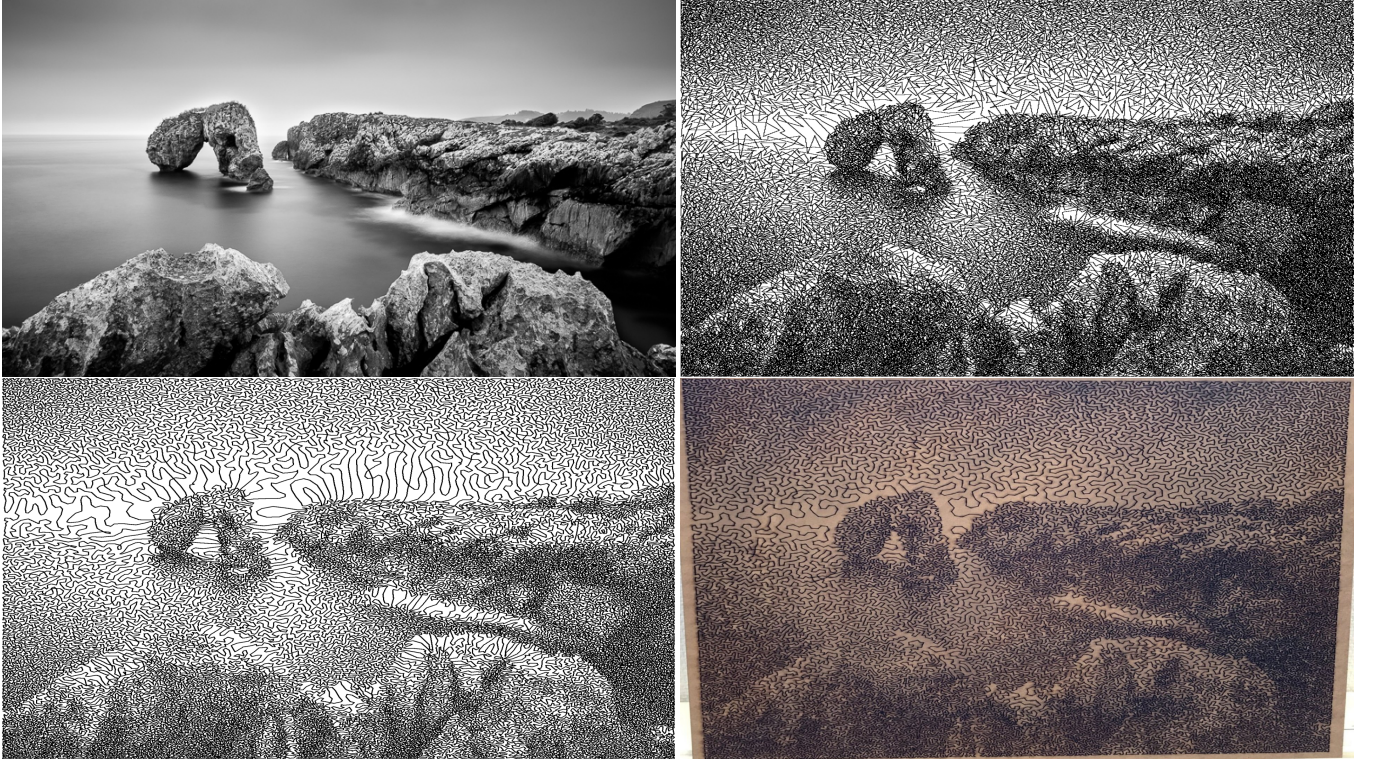


Figure 10: Curvling by  $\frac{3}{4}$ -discrete optimal transport. Original image (top left), approximation by a polyline with 400K segments without any constraints (top right), approximation by a polyline with kinematic constraints and 250K segments (bottom left), final rendering after wood engraving, the polyline is composed of 80K segments (bottom right). In all tests measure  $\mu$  is represented by 320K Dirac masses.

### 5.1.1 Galaxy filaments

Galaxies are known to cluster along filaments and other low dimensional structures [23, 5]. However mathematical extraction of these filaments is a challenging task. We might try to apply refinements of our algorithm to find those filaments. As a naive proof of concept, we have applied our method while setting  $\rho_{2k+1} = 0$  for all  $k$ , so that the lines are disjoint. We use the data of [24] available at : <https://github.com/etempel/bisous>. The galaxies are represented by Dirac masses. Note that in our tests their masses  $m$  are arbitrarily set to  $\frac{1}{n}$  but the code enables other values. In Figure 11, computations are performed for  $n = 180K$  galaxies and a decreasing number of filaments 8K,1K,500.



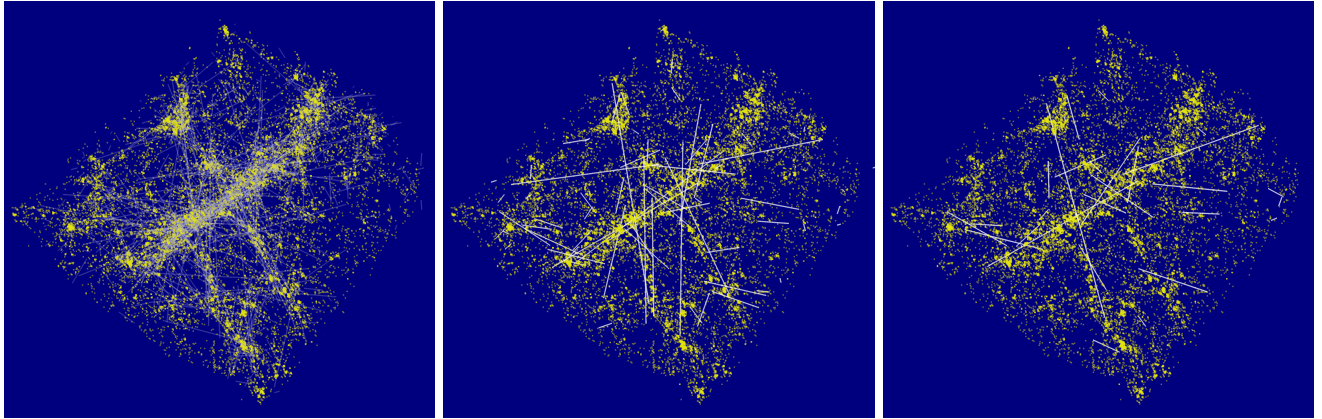


Figure 11: A thin slice of the global solution for locating Bisous filaments (white), and galaxies (yellow).

## References

- [1] AURENHAMMER, F. Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing* 16, 1 (1987), 78–96.
- [2] BENAMOU, J.-D., AND BRENIER, Y. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik* 84, 3 (2000), 375–393.
- [3] BENAMOU, J.-D., CARLIER, G., CUTURI, M., NENNA, L., AND PEYRÉ, G. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing* 37, 2 (2015), A1111–A1138.
- [4] BERTSEKAS, D. P. *Nonlinear programming*. Athena scientific Belmont, 1999.
- [5] BEYGU, B., KRECKEL, K., VAN DE WEYGAERT, R., VAN DER HULST, J., AND VAN GORKOM, J. An interacting galaxy system along a filament in a void. *The Astronomical Journal* 145, 5 (2013), 120.
- [6] CGAL, Computational Geometry Algorithms Library. <http://www.cgal.org>.
- [7] CUTURI, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems* (2013), pp. 2292–2300.
- [8] DE GOES, F., BREEDEN, K., OSTROMOUKHOV, V., AND DESBRUN, M. Blue noise through optimal transport. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 171.
- [9] DE GOURNAY, F., KAHN, J., AND LEBRAT, L. Differentiation and regularity of semi-discrete optimal transport with respect to the parameters of the discrete measure. *arXiv preprint arXiv:1803.00827* (2018).
- [10] DE GOURNAY, F., KAHN, J., LEBRAT, L., AND PIERRE, W. Approches variationnelles pour le stippling: distances l2 ou transport optimal ? In *GRETSI 2017 XXVI* (Sept. 2017).
- [11] DE GOURNAY, F., KAHN, J., LEBRAT, L., AND PIERRE, W. Optimal transport approximation of measures. *arXiv preprint arXiv:1804.08356* (2018).
- [12] DWYER, R. A. Higher-dimensional voronoi diagrams in linear expected time. *Discrete & Computational Geometry* 6, 3 (1991), 343–367.

- [13] FLETCHER, R. On the barzilai-borwein method. In *Optimization and control with applications*. Springer, 2005, pp. 235–256.
- [14] GLOWINSKI, R. On alternating direction methods of multipliers: a historical perspective. In *Modeling, simulation and optimization for science and technology*. Springer, 2014, pp. 59–82.
- [15] HILLER, S., HELLWIG, H., AND DEUSSEN, O. Beyond stippling methods for distributing objects on the plane. In *Computer Graphics Forum* (2003), vol. 22, Wiley Online Library, pp. 515–522.
- [16] HOFF III, K. E., KEYSER, J., LIN, M., MANOCHA, D., AND CULVER, T. Fast computation of generalized voronoi diagrams using graphics hardware. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 277–286.
- [17] KITAGAWA, J., MÉRIGOT, Q., AND THIBERT, B. Convergence of a newton algorithm for semi-discrete optimal transport. *arXiv preprint arXiv:1603.05579* (2016).
- [18] LÉVY, B. A numerical algorithm for l2 semi-discrete optimal transport in 3d. *ESAIM: Mathematical Modelling and Numerical Analysis* 49, 6 (2015), 1693–1715.
- [19] MÉRIGOT, Q. A multiscale approach to optimal transport. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1583–1592.
- [20] NESTEROV, Y. *Introductory lectures on convex optimization: A basic course*, vol. 87. Springer Science & Business Media, 2013.
- [21] POLYAK, B. T. Introduction to optimization. translations series in mathematics and engineering. *Optimization Software* (1987).
- [22] RAYDAN, M. The barzilai and borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization* 7, 1 (1997), 26–33.
- [23] TEMPEL, E., KIPPER, R., SAAR, E., BUSSOV, M., HEKTOR, A., AND PELT, J. Galaxy filaments as pearl necklaces. *Astronomy & Astrophysics* 572 (2014), A8.
- [24] TEMPEL, E., STOICA, R. S., KIPPER, R., AND SAAR, E. Bisous model detecting filamentary patterns in point processes. *Astronomy and Computing* 16 (2016), 17–25.
- [25] VILLANI, C. *Optimal transport: old and new*, vol. 338. Springer Science & Business Media, 2008.