Symmetric semi-discrete optimal transport for mesh interpolation

Agathe Herrou

THESE de DOCTORAT DE

L'UNIVERSITE CLAUDE BERNARD LYON 1

Ecole Doctorale N° 512
Informatique et Mathématique de Lyon

Discipline : Informatique

Soutenue publiquement le 20/10/2022, par :
Agathe HERROU

Symmetric semi-discrete optimal transport for mesh interpolation

Devant le jury composé de :

Mme DELON Julie	Professeure des Universités, Université Paris Cité	Rapportrice
M. THIBERT Boris	Maître de Conférences, Université Grenoble Alpes	Rapporteur
M. SANTAMBROGIO Filippo	Professeur des Universités, Université Lyon 1	Président du jury
Mme ATTALI Dominique	Directrice de recherches, CNRS Grenoble	Examinatrice
M. BONNEEL Nicolas	Chargé de recherches, CNRS Lyon	Directeur de thèse
Mme DIGNE Julie	Chargée de recherches, CNRS Lyon	Co-directrice de thèse
M. LÉVY Bruno	INRIA Nancy Grand-Est	Invité

Laboratoire LIRIS
Adresse
Bâtiment Nautibus
Campus de la Doua
25 avenue Pierre de Coubertin
69622 Villeurbanne Cedex

École Doctorale InfoMaths
7, avenue Jean Capelle
69621 VILLEURBANNE Cedex

Titre Transport optimal semi-discret symétrique pour l'interpolation de maillages

Résumé Cette thèse a pour but de développer des méthodes géométriques pour approximer l'interpolation de déplacement, issue du transport optimal. Le transport optimal est une théorie mathématique modélisant des déplacements de matière sous une contrainte de minimisation de coût, avec de nombreuses applications en physique, en informatique graphique et en géométrie. Le coût minimal du déplacement entre deux distributions définit une distance, qui elle-même est à l'origine de l'interpolation de déplacement. Ces interpolations peuvent sous certaines conditions présenter des discontinuités, que les approximations discrétisées du transport optimal n'arrivent pas toujours à bien capturer. Le travail de cette thèse vise à développer une approximation qui capture bien ces discontinuités.

Notre méthode s'appuie sur le transport optimal semi-discret, où seul l'une des distributions est discrétisée, capturant ainsi avec précision les discontinuités de la distribution restée continue. Les plans de transport ainsi obtenus partitionnent la distribution continue en cellules associées aux échantillons de la discrétisation. On peut donc assimiler un plan de transport optimal semi-discret à un diagramme de puissance composé de ces cellules. Cette variante du transport optimal a cependant l'inconvénient de briser la symétrie entre les deux distributions. Nous commençons par formaliser notre problème comme la recherche de plans de transport couplés par le biais des barycentres de leurs cellules. Nous présentons ensuite un premier algorithme pour le calcul de ces plans de transport couplés. Il repose sur un schéma classique d'algorithme alterné, calculant successivement des plans de transport et les barycentres de leur cellules jusqu'à convergence. Les résultats obtenus à partir de cet algorithme permettent d'interpoler entre les distributions initiales en conservant une précision satisfaisante, en particulier au niveau des discontinuités, et y compris lorsque la discrétisation des distributions est faits avec relativement peu de points. Nous présentons ensuite notre exploration de méthodes d'optimisation pour résoudre le même problème. Ces méthodes expriment les contraintes de notre problème comme un point critique d'une fonctionnelle, et cherchent à atteindre ces points à l'aide d'algorithmes tels que l'algorithme de Newton. Cette approche n'a cependant pas donné de résultats concluants, les fonctions considérées étant trop bruitées pour se prêter à des algorithmes d'optimisation.

Mots clés Transport optimal, Interpolation, Optimisation, Géométrie algorithmique

Title Symmetric semi-discrete optimal transport for mesh interpolation

Abstract This thesis aims to develop geometric methods to approximate displacement interpolation, derived from optimal transport. Optimal transport is a mathematical theory modeling movements of matter under a cost minimization constraint, with many applications in physics, computer graphics and geometry. The minimum displacement cost between two distributions defines a distance, which itself is at the origin of displacement interpolation. This interpolation may under certain conditions present discontinuities, that the discretized approximations of the optimal transport do not always successfully capture. The work of this thesis aims to develop an approximation that captures these discontinuities well.

Our method relies on semi-discrete optimal transport, where only one of the distributions is discretized, thus accurately capturing the discontinuities of the distribution that remains continuous. The transport plans thus obtained partition the continuous distribution into cells associated with the samples of the discretization. A semi-discrete optimal transport plan can thus be assimilated to a power diagram made up of these cells. This variant of optimal transport however has the disadvantage of breaking the symmetry between the two distributions. We start by formalizing our problem as the search for a pair of transport plans coupled through the barycenters of their cells. We then present an algorithm for calculating these coupled transport plans. This first algorithm is based on a classical alternating algorithm scheme, successively computing the transport plans and the barycenters of their cells until convergence. The results obtained from this algorithm allow to interpolate between the initial distributions while maintaining a satisfactory precision, in particular when it comes to discontinuities, including when the discretization of the distributions is done with relatively few points. We then present our exploration of optimization methods for solving the same problem. These methods express the constraints of our problem as a critical point of a functional, and aim to reach these points using algorithms such as Newton's method. However, this approach did not yield conclusive results, as the functions involved were too noisy to lend themselves well to optimization algorithms.

Keywords Optimal transport, Interpolation, Optimization, Algorithmic geometry

Contents

Résumé en français

Cette thèse a pour but de développer des méthodes géométriques pour l'approximation de l'interpolation de déplacement. L'interpolation de déplacement est une interpolation basée sur la théorie du transport optimal, qui modélise des déplacements de matière sous des contraintes de coût. Le coût minimal du déplacement de la matière entre deux distributions définit une distance sur l'ensemble des distributions, et l'interpolation associée est l'interpolation de déplacement, qui représente le trajet de la matière d'une distribution à une autre. La fonction représentant le déplacement de la matière d'un point de la distribution de départ à la distribution d'arrivée est appelée un plan de transport.

La théorie du transport optimal et l'interpolation de déplacement trouvent de nombreuses applications dans des domaines variés. On peut par exemple citer le transfert de couleurs, où le transport s'effectue dans l'espace des couleurs entre deux histogrammes ; les algorithmes d'apprentissage, où cette notion de distance permet de mesurer l'adéquation des résultats générés aux données d'entrée ; ou encore la simulation de fluides, les distributions représentant l'état du fluides à différents pas de temps.

Les algorithmes développés dans cette thèse utilisent une version du transport optimal appelée le transport semi-discret. Dans ce cadre, la distribution d'arrivée du transport est discrétisée par un nuage de points (appelés sites), tandis que la distribution de départ reste continue. Les plans de transport ainsi produits partitionnent l'espace de départ en un ensemble de polyèdres convexes, en correspondance avec les points de la distribution discrète. Ces polyèdres forment une variante des diagrammes de Voronoi appelée diagrammes de puissances, où l'aire de chaque cellule est contrôlée par un paramètre scalaire, appelé son poids. Les poids définissant un plan de transport optimal peuvent être calculés en minimisant une fonctionnelle convexe définie sur l'ensemble des vecteurs de poids. Il existe plusieurs algorithmes efficaces pour optimiser cette fonctionnelle.

L'inconvénient principal du transport semi-discret est son manque de

symétrie : l'une des distributions étant discrétisée et l'autre continue, elles ne peuvent pas jouer des rôles interchangeables. Ainsi, dans le but de calculer une approximation de l'interpolation de déplacement entre deux distributions, nous contournons ce problème en définissant non pas un mais deux plans de transport, chacun transportant de l'une des distributions vers une discrétisation de l'autre. Afin de capturer les caractéristiques du plan de transport, en particulier ses discontinuités, ces plans de transport sont couplés : on impose que les points de la discrétisation d'un domaine soient situés aux barycentres des cellules le transportant vers l'autre domaine. En conséquence, les deux discrétisations doivent comporter le même nombre d'échantillons.

Nous présentons un premier algorithme pour calculer ces plans de transport couplés. Cet algorithme s'inspire d'algorithmes classiques tels que l'algorithme de Lloyd. Appliqué à deux distributions μ et ν, il consiste à alternativement calculer le plan de transport de sa distribution de départ (par exemple μ) vers la discrétisation de sa distribution d'arrivée (ici ν), puis repositionner les échantillons de la discrétisation de μ sur les barycentres des cellules de μ. La deuxième partie de cette itération s'effectue en réalisant la même opération en inversant les rôles de μ et ν. Cet algorithme converge en pratique en une centaine d'itérations. L'interpolation linéaire des nuages de points discrétisant μ et ν produit une première approximation de l'interpolation de déplacement entre μ et ν.

Nous étudions ensuite la possibilité d'atteindre le même objectif par des méthodes d'optimisation itératives, telles que la descente de gradient ou l'algorithme de Newton. Ces optimisations portent sur des fonctions objectif constituées d'une combinaison de la fonctionnelle du transport et de fonctions exprimant les contraintes barycentriques. Au contraire de l'optimisation calculant le transport, ces optimisations portent sur les poids du transport et les positions des sites. La fonctionnelle de transport, considérée comme une fonction de ces deux variables, n'est ni convexe, ni concave, mais comporte un point selle incompatible avec l'objectif poursuivi dans cette thèse. L'utilisation d'une simple descente de gradient a donc été abandonnée pour cette raison. Nous présentons ensuite une variante de la méthode de Newton visant à minimiser une fonction sous des contraintes d'égalité. Cette méthode cherche à annuler un vecteur exprimant toutes les contraintes du problème (optimalité des plans de transport et recentrage des sites). Bien que cette méthode semble théoriquement appropriée à la résolution de notre problème, le caractère très sensible des plans de transport aux perturbations la rend concrètement impraticable.

Enfin, nous exposons comment utiliser le résultat de cet algorithme pour

calculer une approximation fidèle de l'interpolation entre μ et ν. Nous nous appuyons pour cela sur une propriété remarquable de nos plans de transport couplés, dans lesquels les relations de voisinages entre cellules sont préservées d'un diagramme à l'autre (i.e., si deux cellules sont voisines dans le diagramme de μ, leurs cellules correspondantes seront aussi voisines dans le diagramme de ν). En conséquence, la géométrie des cellules est également préservée, hormis dans le cas où le plan de transport comporte des discontinuités. Dans ce dernier cas, on observe empiriquement que les frontières des cellules s'alignent avec ces discontinuités. Cela permet d'obtenir, en interpolant linéairement entre les sommets des diagrammes de puissance, une approximation de l'interpolation de déplacement qui couvre la totalité des distributions initiales et capture correctement ses discontinuités. Une évaluation quantitative de cet algorithme, en comparant ces résultats à ceux d'une approximation précédente, considérée comme réalité terrain, montre qu'il donne de meilleurs résultats, y compris en utilisant bien moins de points pour ses discrétisations.

Remerciements

Je remercie mes directeurs de thèse, Nicolas Bonneel et Julie Digne, de m'avoir donné l'opportunité de me lancer dans cette aventure de recherche et de m'avoir guidée tout au long de ce parcours malgré les nombreuses difficultés que j'ai pu rencontrer. Je remercie également Bruno Lévy, qui a donné l'impulsion scientifique de cette thèse, et dont les conseils ont permis d'éclaircir de nombreux points obscurs de ce sujet.

Je remercie les autres membres de mon équipe, permanents, doctorants, stagiaires, avec qui j'ai pu avoir des discussions scientifiques enrichissantes, en tête à tête ou en réunion d'équipe. Je remercie en particulier Vincent Nivoliers, qui m'a permis de poser un œil nouveau sur mon sujet de recherche. Je remercie tous les gens qui ont pu m'aider avec les aspects techniques de ma thèse, en particulier la prise en main de ce mastodonte de code qu'est Graphite. Je remercie également les personnels administratifs du labo pour leur aide précieuse et leur efficacité, qui ont rendu cette thèse un peu plus facile.

Je remercie les professeurs pour qui j'ai enseigné au cours de ces quatre années (enfin, surtout la dernière). Ces enseignements n'ont pas toujours été de tout repos (surtout ceux du lundi 8h), mais ce fut une expérience enrichissante, qui m'aura permis d'un peu démystifier le métier d'enseignant pour peut-être, qui sait, m'y engager définitivement un jour. Je remercie également la cellule médiation pour les activités de transmission au près de plus jeunes élèves auxquelles j'ai pu participer.

Je remercie chaleureusement toute la bande des doctorants, dont la composition aura bien évolué du début à la fin de ma thèse (eh oui c'est ça de tous soutenir pendant le covid). J'espère que nous aurons à nouveau l'occasion de faire des pub quizz et de boire des coups ensemble, bien que le temps semble inéluctablement nous transformer en trentenaires à apéros dînatoires.

Je remercie les musiciens avec qui j'ai pu jouer pendant ces quatre dernières années, de la fanfare à la scène live-coding lyonnaise. Jouer de la musique aura été pour moi une bouffée d'oxygène salutaire pendant ma thèse, et sans

elle je ne serais probablement pas arrivée au bout de ce travail.

Je remercie les professeurs qui, du lycée à la fin de mes années d'ENS, m'ont donné le goût des mathématiques, de l'informatique et de leurs intersections. Je n'en serais pas arrivée là sans vous, et j'espère me montrer digne de l'enseignement que vous m'avez donné. Aux professeurs qui m'ont aussi donné le goût des langues et des lettres, je suis désolée de ne pas avoir suivi votre voie académiquement, mais je continuerai à faire de mon mieux pour la cultiver en autodidacte.

Je remercie enfin mes parents, dont l'éducation qu'ils m'ont donné depuis mon enfance m'a permis d'atteindre ce niveau avancé dans un domaine scientifique aussi pointu, et ce alors qu'il est très éloigné de leurs compétences à eux. Je remercie également mes sœurs, qui ne m'ont pas éduquée mais dont la joyeuse présence aura contribué à mon enfance heureuse. Last but not least, je remercie enfin mon compagnon, qui a été un soutien indéfectible et un conseiller précieux à mes côtés pendant toutes ces années. J'espère que nous pourrons continuer à nos épauler mutuellement dans la suite de nos carrières respectives.

Introduction

Interpolating between distributions is a mathematical operation with a large range of applications in computer graphics and computer vision. This operation can be used whenever one wants to compute new values defined as intermediates between two or more sets of input data. It can be used to generate objects such as BRDFs or color palettes [BvdPPH11], to compute intermediate frames in animation [Ree81] or to reconstruct distributions from noisy measurements [DCSA+14], to only cite a few applications.

The simplest notion of interpolation is probably linear interpolation, where the intermediate point between any two input points is obtained by computing their weighted barycenter. However, while it can be sufficient for some applications, this interpolation quickly shows some limitations, in particular when one aims to model physical deformations. Indeed, the result of linear interpolation is closer to a cross-fade with no physical interpretation. Well-named displacement interpolation, based on optimal transport, fills this gap, and yields intermediate distributions on the basis of what it would cost to move one to the other (see Fig. 1). It is thus better-suited to model physical phenomena, in particular those involving deformations.

The theory of optimal transport was initially introduced to model displacements of matter under a cost minimization constraint. From these very general principles, several physical models can be derived, making optimal transport a very versatile tool with several applications.

Optimal transport has seen numerous developments in the last decades, on a theoretical level, with applications in geometry, probably theory and PDEs, as one can read about in Cédric Villani's monograph [Vil09], as well as in its applications, as developed in the works of Filippo Santambrogio [San15] and of Gabriel Peyré and Marco Cuturi [PC+17].

Computational optimal transport The classical way of computing optimal transport between two distributions is to discretize them and to use a linear solver to solve the associated linear program. However, this is very

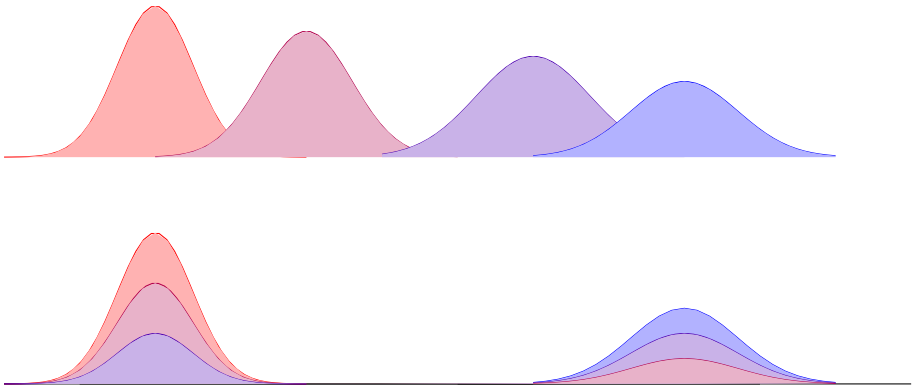


Figure 1: Displacement interpolation (up) models a translation and has physical meaning, while linear interpolation (down) suggest "teleportation" of matter. Initial measure is in red, final measure is in blue and interpolation steps are indicated by intermediate colors.

computationally costly, especially when the discretizations involve a lot of points. Several less costly variants have been introduced to counter that.

Such variants include entropic regularization [Cut13] and sliced optimal transport [RPDB11]. This thesis focuses on semi-discrete optimal transport, which has interesting geometric properties that we will take advantage of. Semi-discrete optimal transport is a discretization of optimal transport that only discretizes the source distribution. While seemingly counter-intuitive as a simplification of the problem, this gives it a geometric interpretation as a power diagram [AHA98], which is fully described by a finite-dimension vector of weights. Computing a semi-discrete transport plan then amounts to computing the right value of the weights, which is at the minimum of a convex function. The problem of convex optimization has been abundantly studied, and thus a wealth of efficient algorithms exist to compute semi-discrete optimal transport plans.

The question of finding discontinuities in optimal transport plans in general is a difficult problem, and a challenge for good approximations of optimal transport is to accurately capture these discontinuities. Some partial results exist about them: (continuous) optimal transport maps that transport to a convex shape equipped with a uniform measure are necessarily continuous [Caf92], and, conversely, optimal transport maps that transport from a convex shape to a shape whose boundaries presents regions of sufficiently negative total curvature will result in a transport plan with discontinuities [CJL$^+$15].

These discontinuities in the transport plan can be important when optimal transport is used to model physical phenomena: they usually represent points where matter separates. However, existing approximations of (continuous) optimal transport through semi-discrete optimal transport usually do not represent well these discontinuities.

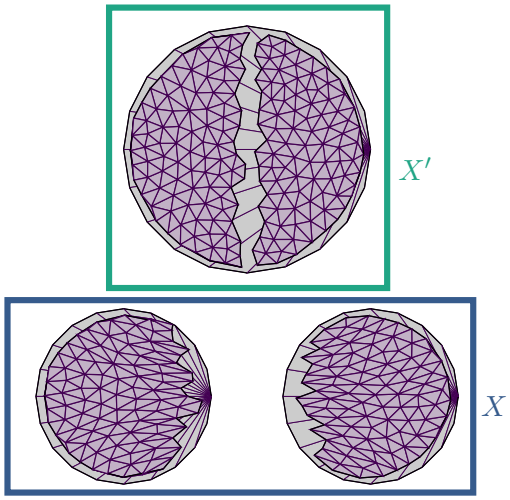


Figure 2: The initial and final positions of the interpolating mesh computed with Lévy's algorithm [Lé14] superimposed with the two meshes X and X'. We observe that the interpolations erodes the original meshes and that it presents jagged boundaries on discontinuities.

Figure 2 showcases the result of the semi-discrete approximation proposed by Bruno Lévy [Lé14], when approximating transport from a 2-d disk to two smaller disks (all equipped with a uniform measure). This transport plan presents a discontinuity due to the changes in topology from the source to the target. This approximation lacks precision in capturing this discontinuity.

The goal of this thesis is to develop an approximation of optimal transport, using semi-discrete optimal transport, that properly represent these discontinuities.

Applications of optimal transport Optimal transport, as a theory arising from a concrete engineering problem, unsurprisingly finds many applications in applied mathematics and in computer graphics.

The evolution of physical systems can be cast as optimal transport problems, an impressive example being the reconstruction of the early stages of the universe from present-day observations [LMvH21]. Problems from

computer graphics can be modeled by optimal transport as well, such as color transfer [RDG10], where one aims to transform the color histogram of an image into that of another, while minimizing the amount of modifications performed, in order to preserve the initial characteristics of the image.

The distance underlying displacement interpolation, called the earth mover distance or Wasserstein distance, models the amount of work needed to displace a distribution to another. It can thus be used to compare distributions in terms of how much one has to transform one to turn it into another. For example, one can use it to measure how close two images are in terms of features [RTG04], or to measure how accurate the reconstruction [DCSA$^+$14] of a surface from a point cloud is. Such applications can demand a specialized formulation of optimal transport, such as optimal transport on the circle [DSS10] for histograms. The barycenters induced by the earth mover distance can be approximated using semi-discrete optimal transport, as is done by Claici et al. [CCS18].

As a metric, earth mover distance can prove more interesting than concurrent metrics for comparing distributions, either because of its strong physical meaning or because of its topological properties. This can make the earth mover distance a good candidate in generative models [GPC18] or GANs [ACB17], where it is used to measure the fitness of the generated output to the model's input.

Goal of this thesis The aim of this thesis is to develop methods to approximate displacement interpolation between meshes, using the geometrical properties of semi-discrete optimal transport. In particular, we want our approximations to faithfully represent the discontinuities that might arise in the transport plan. Semi-discrete optimal transport, in its classical form, only takes properly into account the discontinuities arising from the discretized mesh, which poses a problem when, for example, both meshes are disconnected. To overcome this limitation, we propose a symmetrized version of semi-discrete optimal transport, in which both meshes are discretized and two coupled semi-discrete transport plans are computed. We show that the power diagrams representing these transport plans can be linearly interpolated, and that this resulting interpolation does, in practice, properly take into account possible discontinuities.

Notations

We present here the naming conventions that will be used accross this manuscript.

- μ, ν denote positive, continuous measures, typically admitting a density,
- X, Y denotes the continuous spaces (typically an open subset of $\mathbb{R}^d$) over which these measures are defined,
- $c : X \times Y \to \mathbb{R}$ denotes a cost function over these spaces,
- x_i, y_i denote the samples discretizing continuous densities (also called sites in the context of semi-discrete transport plans),
- N is the number of such samples, identical for both samplings,
- p_i, q_i denote the weights assigned to the sites x_i, y_i,
- ϕ_i, ψ_i denote the weights defining transport plans,
- $\mathcal{X} = (x_i, \phi_i, y_i, \psi_i)$ denotes a variable gathering all the parameters defining two coupled semi-discrete transport plans,
- Φ (or variants Φ_x, Φ_y) denote the transport functional to optimize over these weights.

Chapter 1

State of the art

In this chapter, we will start by recalling the bases of the theory of optimal transport and the distance it induces on measures, as well as the associated interpolation. We will then delve more precisely into semi-discrete optimal transport and optimization algorithms to compute them. We review classical algorithms using alternating strategies to compute partitions or samplings of continuous domains, which serve as inspiration for one of the algorithms presented in this thesis. At last, we review applications of optimal transport (semi-discrete or not) in various fields.

1.1 General optimal transport

1.1.1 Monge's formulation

The first formulation of optimal transport is due to French mathematician and engineer Gaspard Monge, which he introduced in 1781 in his Mémoire sur la théorie des déblais et des remblais [Mon81]. The concrete problem that he was initially aiming to solve, and that is traditionally given as the canonical example to illustrate optimal transport, is the one of moving sand from a given initial position to a final one, say, to fill a hole or to build a fortification, while spending as little energy as possible. The expected result is a function mapping each initial position of a sand particle to the final position this particle should occupy at the end of the motion.

This concrete problem gives an example of optimal transport where one aims to transport an uncountable quantity, here the sand, which can be considered with infinitesimally small particles. This setting is called continuous optimal transport. However, optimal transport can also take place between discrete objects, the archetypal example for this variant being the problem

of mines and factories. Given a set of mines producing iron ore, and a set of factories using this iron ore to manufacture their products, one wishes to assign the production of a mine to a factory for its consumption, while minimizing the effort of moving the iron ore, computed as the mass of ore displaced times the distance traveled.

We can formalize both these problems under the following common framework. We consider an initial measure μ over a set X, representing either the initial distribution of sand or the production of each mine, and a final measure ν over a set Y, representing the desired distribution of sand or the ore consumption of each factory. We consider with a cost function $c : X \times Y \to \mathbb{R}$, representing the cost of displacing matter between a point of X and a point of Y. In practice, we take c to be the squared L_2 distance, also called quadratic cost: $c(x, y) = \|x - y\|^2$, for which optimal transport plans have remarkable geometric properties, which will be developed in Section 1.2.

Solving the optimal transport problem then consists in finding a function $T : X \to Y$, called a transport map, minimizing the total cost of displacement in a measure-preserving way.

We wish for this transport map to preserve the mass during the transportation operation, expressed by the following constraint

$$\nu = T_{\#}\mu$$

which models mass conservation from the initial measure to the final one.

The notation $f_{\#}\mu$, where $f : X \to Y$ is a measurable function, denotes a measure called the pushforward measure of the measure μ by the function f. It is defined as $f_{\#}\mu(A) = \mu(f^{-1}(A))$ for each measurable set $A \subseteq Y$. Intuitively, this means that no matter is lost (or created) during transportation: the pushforward measure of every measurable set A is equal to the total measure of matter transported to A.

Given this constraint, we wish to minimize the cost of displacement, which is given by the following formula:

$$\int_{x \in X} c(x, T(x)) d\mu(x)$$

where, for each x in the initial set X, we sum the cost of displacing x to its target location $T(x)$, weighted by the mass $\mu(x)$ of x.

The formulas we have presented are more adapted to describing the case where measures μ and ν are continuous, called continuous optimal transport. In the case where X is a discrete set $\{x_i \mid 1 \leq i \leq N\}$, the cost function becomes

$$\sum_{i=1}^{N} c(x_i, T(x_i))\mu_i$$

and the pushforward constraint is $\forall j, \nu_j = \sum_{i, T(x_i)=y_j} \mu_i$. This version of optimal transport is called discrete optimal transport.

1.1.2 Kantorovich's formulation

The main weakness of Monge's formulation comes from the fact that it represents the optimal transport map as a function $X \to Y$ instead of a general relation between X and Y. This introduces a fundamental dissymmetry, which for example translates to the fact that one can merge matter (map two different source points to the same target) but not split it (map a point to two different targets).

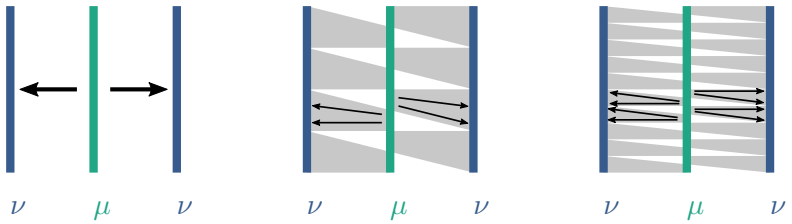


Figure 1.1: Example of an optimal transport problem that has no solution under Monge's formalism: there is no function representing an optimal transport plan from source measure μ, composed of one line segment, to target measure ν, composed of two parallel segments. Figure inspired by [Vil09] and [Lé14].

Figure 1.1 illustrates a situation where there is no solution Monge's formulation could model. In this problem, we try to find a transport map that transports a single segment to two identical segments. We can build a sequence of mappings from μ to ν where every step improves the cost of the transport, but where the infimum is never realized by a function.

More generally, such problems arise when μ concentrates matter on sets with Lebesgue measure equal to zero, which cannot be properly split using a function. This shows this formulation needs to be relaxed.

Instead of representing the optimal transport from (X, μ) to (Y, ν) as a function from X to Y, we model it as a measure γ over the product space $X \times Y$, where $\gamma(x, y)$ represents the amount of matter to move from x to

y. The measure formalism allows us to move only a fraction of the matter located on x to y.

The quantity to be minimized thus becomes

$$\int_{X\times Y} c(x,y)d\gamma(x,y)$$

and the pushforward constraint becomes a constraint on γ's marginals: if we denote π_X and π_Y the projections on X and Y respectively, we demand that $\mu = \pi_{X\#}\gamma$ and $\nu = \pi_{Y\#}\gamma$. In other terms, this means that for every measurable set $A \subset X$ (resp. Y), $\mu(A) = \gamma(A\times Y)$ (resp. $\nu(A) = \gamma(X\times A)$).

Such a γ is called a transport plan, and we denote by $\Gamma(\mu,\nu)$ the set of such measures γ.

Theorem 1. *A solution to the Kantorovich problem exists when the costs function is the quadratic cost and X, Y are compact sets.*

This is a consequence of a more general theorem stating the existence of a solution when c is lower semi-continuous. A proof can be found in Ambrosio et al. [AGS05], chapter 6.

Ultimately, these two formulations deal with the same problem: given a solution that solves Monge's problem, we can adapt it into a solution to Kantorovich's problem, as explicited by Proposition 1. However, as we just showed, the converse is not necessarily true.

Proposition 1. *If $(Id\times T)_{\#}\mu \in \Gamma(\mu,\nu)$, then T pushes μ forward to ν.*

A proof can for example be found in Lévy's article [Lé14].

1.1.3 Dual formulation of Kantorovich's problem

Kantorovich's formulation shows that the optimal transport problem can be interpreted as a linear program:

$$\min_{\gamma\in\Gamma(\mu,\nu)} \int_{(x,y)\in X\times X} c(x,y)d\gamma(x,y)$$

This appears more clearly when considering the discrete setting, where the expression to minimize is written $\sum_{i,j} c_{i,j}\gamma_{i,j}$ and the constraints are $\sum_j \gamma_{i,j} = \mu_i$, $\sum_i \gamma_{i,j} = \nu_j$ and $\gamma_{i,j} \geq 0$.

As a linear program, this problem has a dual formulation. Instead of minimizing the total cost, one will maximize the sum of two dual variables ϕ and ψ:

$$\max_{\psi,\phi}(\int_X \psi d\mu + \int_Y \phi d\nu) \tag{1.1}$$

where $\psi : X \to \mathbb{R}$ and $\phi : Y \to \mathbb{R}$ are bounded functions, under the constraint that $\forall x \in X, y \in Y, \psi(x) + \phi(y) \leq c(x, y)$.

When ψ and ϕ are functions that satisfy this constraint, the c-conjugate of ϕ, defined by $\phi^c(x) = \inf_{y \in Y} c(x, y) - \phi(y)$, is also a suitable function and improves the value that formula 1.1 aims to maximize. This yields a new formulation of the problem:

$$\max_{\phi} \int_{x \in X} (\inf_{y \in Y} c(x, y) - \phi(y)) d\mu(x) + \int_{y \in Y} \phi(y) d\nu(y) \tag{1.2}$$

The canonical interpretation of the dual formulation stems from the mines and factories problem. Instead of directly transporting the iron ore from the source to the targets, one hires a transporter that only charges for loading the ore coming out of the mines and unloading it at the factories, these respective costs being the dual variables ϕ and ψ we just introduced. Since the transporter wants to maximize its profit, its goal is to maximize the sum of those variables. However, it also needs to be a credible alternative to the mines and factories operator simply doing the transportation themselves, thus the cost $\phi(x) + \psi(y)$ of loading and unloading at a given (x, y) mine-factory pair cannot be greater that the cost $c(x, y)$ of transporting between them.

1.1.4 Displacement interpolation

The smallest transport cost defines a distance over measures, called the earth mover's distance or Wasserstein metric. When equipping the set of measures over a given set X with this distance, it turns it into a Riemannian manifold, on which one can carry interpolations, called displacement interpolation.

The p-th Wasserstein distance between two probability measures μ and ν is given by

$$W_p(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{X \times X} d(x, y)^p d\gamma(x, y) \right)^{1/p}$$

In the most common case, $p = 2$. One can prove that, whenever d is a distance function, W_p is also a distance over the space of probability measures over X. McCann[McC97] introduced the displacement interpolation of two measures, and Agueh and Carlier [AC11] generalized it to more than two measures with barycenters in the Wasserstein space.

There is an explicit formulation of Wasserstein barycenters between two measures.

Proposition 2. *If Monge's formulation yields an optimal transport map T, then geodesics between μ and $T_{\#}\mu$ in the Wasserstein space are given by*

$$\mu_t = ((1-t)Id + tT)_{\#}\mu$$

where $t \in [0;1]$.

A proof can be found in Villani's monograph [Vil09].

This is an interpolation between measures that can be more relevant than linear interpolation for a range of applications.

1.2 Semi-discrete optimal transport

Asides from continuous optimal transport, where the source and target measures have a density and are typically supported by an open subset of $\mathbb{R}^n$, and discrete optimal transport, where, on the contrary, both source and target measures are finitely supported by a sum of diracs, a third case exists, where the source measure is continuously supported and the target measure is finitely supported: this is the so-called semi-discrete optimal transport.

The canonical practical example to illustrate semi-discrete optimal transport, as presented in [KMT16], is the one of a city endowed with a population density, which has to share a set of bakeries, placed on the map at discrete locations. We know the production capacity of each bakery, which correspond to the measure of its dirac, and the transport constraint imposes that the total needs of the population equal the total production of the bakeries. To the difference of the mines and factories setting, the source is represented by a continuous density instead of a sparse set of locations. The choice of each bakery by customers will be determined by a combination of their distance to the bakery and of its prices relative to its competitors. Solving the optimal transport problem here means to find the set of prices that will exactly match the production capacity of a bakery to the demand of its customer base.

In the remainder of this document, we will consider a source measure with density ρ: $d\mu(x) = \rho(x)dx$ and a target measure supported by a sum of N diracs: $\nu = \sum_{i=1}^{N} p_i \delta_{x_i}$. We will call sites the dirac locations x_i.

Since the target Y is finite, the function $\phi : Y \to \mathbb{R}$ can be finitely represented as a vector of $\mathbb{R}^N$ indexed by Y, and computing an optimal transport plan amounts to maximizing a functional over $\mathbb{R}^N$.

We can adapt the c-transformed version of Kantorovich's dual formulation (equation 1.2) to the semi-discrete case, by adopting the discrete notation $(\phi_i)_{1\leq i\leq N}$ for the dual variable ϕ. If the cost function c verifies the so-called twist condition, which states that the differential of c with respect to its first variable is injective (see for example [KMT16] for more details), the infimum over a finite set can be simplified into a minimum $\min_{1\leq i\leq N}(c(x,x_i)-\phi_i)$ and partition X into cells $Cell_i = \{x \mid \forall j, c(x,x_i)-\phi_i \leq c(x,x_j)-\phi_j\}$:

$$\max_{\phi} \sum_{i=1}^{N} \int_{x\in Cell_i} (c(x,x_i)-\phi_i)\, d\mu(x) + \sum_{i=1}^{N} \phi_i p_i \tag{1.3}$$

In terms of bakeries, this means that the city is divided along the territories of the bakeries. If every bakery has identical products and implements identical prices, clients simply go to the nearest bakery: the territories map is a Voronoi diagram. If the prices are different from one bakery to another, clients make a trade-off between prices and distance by minimizing $c(x,x_i)-\phi_i$.

We will discuss next the properties of such partitions.

1.2.1 Transport plans as power diagrams

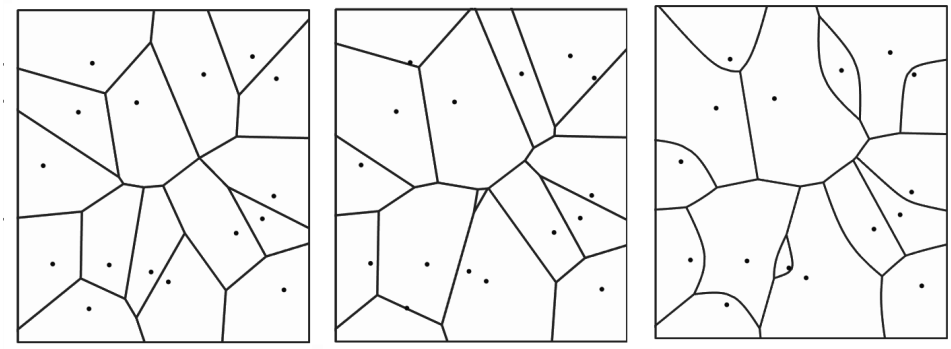


Figure 1.2: From left to right: a Voronoi diagram, a Power diagram, and a Laguerre diagram for cost $c(x,y) = \|x-y\|$ (also known as an Apollonius diagram), all for the same set of sites. Image from [Sca15].

Let us now introduce some necessary definitions. We consider a domain $X \subset \mathbb{R}^d$, a distance (or cost) function c over X, and a set of sites $\{x_i \mid 1 \leq i \leq N\} \subset X$. Here are some common ways to partition X according to the (x_i)s.

Definition 1 (Voronoi Diagram)**.** *A Voronoi diagram is a partition of the domain X according to sites (x_i) such that the cell of the partition associated to x_i is made up of the points of X that are closer to x_i than to any other x_j. We denote by $Vor(x_i)$ this Voronoi cell.*

$$\forall i, Vor(x_i) = \{x \in X \mid \forall j, c(x, x_i) \leq c(x, x_j)\}$$

Laguerre diagrams generalize Voronoi diagrams: instead of associating to each sites simply the closest points, a new degree of freedom is added in the form of a weight associated to each site, which will modulate its "influence" and determine which portion of space it can gather.

Definition 2 (Laguerre Diagram)**.** *The Laguerre cell $Lag^{\phi}(x_i)$ of site x_i with respect to weights (ϕ_i) consists in the points $x \in X$ for which $c(x, x_j) - \phi_j$ is minimal for $j = i$.*

$$\forall i, Lag^{\phi}(x_i) = \{x \in X \mid \forall j, c(x, x_i) - \phi_i \leq c(x, x_j) - \phi_j\}$$

Another convention exists for defining Laguerre diagrams, where the minimized expression is $c(x, x_i) + \phi_i$. For example, Kitagawa et al. [KMT16] or Lévy [Lé14] use this convention for their algorithms computing semi-discrete optimal transport. Both are equivalent, but the $-$ convention can be considered more intuitive, in that increasing the weight of a cell increases its area. Besides, we use the convention that is coherent with the c-transform introduced in Section 1.1.3.

Power diagrams are a restriction of Laguerre diagrams where the cost c is the squared euclidean distance.

Definition 3 (Power diagrams)**.**

$$\forall i, Pow^{\phi}(x_i) = \left\{x \in X \mid \forall j, \|x - x_i\|^2 - \phi_i \leq \|x - x_j\|^2 - \phi_j\right\}$$

By definition, a semi-discrete transport plan (optimal or not) defined by a given value of Kantorovich's dual variable $(\phi)_i$, with euclidean cost $c = \|.\|^2$, partitions X into a power diagram. The dual variable $(\phi)_i$ plays the role of the power diagram weights.

We can thus rewrite Equation 1.3 for $c = \|.\|^2$ using the definition of power diagrams. Let us define the functional Φ:

$$\Phi(\phi_i) \;=\; \sum_{i=1}^{N} \int_{x \in Pow^{\phi}(x_i)} \|x - x_i\|^2 \, d\mu(x) + \sum_{i=1}^{N} \phi_i \cdot (p_i - \mu(Pow^{\phi}(x_i))) \tag{1.4}$$

The optimization problem becomes:

$$\max_{\phi_i \in \mathbb{R}^N} \Phi(\phi_i) \tag{1.5}$$

Computing a semi-discrete optimal transport plan for the quadratic cost thus amounts to finding the weights of a power diagram minimizing the functional Φ.

From now on, we will only consider transport problems for the quadratic cost.

Lagrangian interpretation One can interpret Φ as the Lagrangian function associated to a constrained optimization problem. The first sum $\sum_{i=1}^{N} \int_{Pow^{\phi}(x_i)} \|x - x_i\|^2 \, d\mu(x)$ corresponds to the transport cost associated with the power diagram, which we aim to minimize. In the second sum, the ϕ_is play a role akin to Lagrange multipliers, enforcing the constraint $\mu(Pow^{\phi}(x_i)) = p_i$: the actual measures of the power cells are equal to the Dirac's masses, i.e. their prescribed measures. We get back to the basic definition of the optimal transport problem: minimizing the cost of transport while preserving a measure. Finding a constrained minimizer using the Lagrangian boils down to finding a stationary point of the Lagrangian function Φ, which does not necessarily have to be a minimum. It has been shown above that, in this case, the stationary point we are looking for is a maximum.

de Goes et al. [DGBOD12] show that the interpretation of the power weights as Lagrange multipliers is legitimate. This is due to the fact that the sum $\sum_{i=1}^{N} \int_{x \in Pow^{\phi}(x_i)} \|x - x_i\|^2 \, d\mu(x)$ is actually constant with regards to the weights.

Converse implication Besides, Aurenhammer et al. [AHA98] show that the implication that any semi-discrete optimal transport plan is a power

diagram is actually an equivalence. This means that, given a set of weights $(\phi)_i$, the power diagram that is defined by these weights represents a semi-discrete optimal transport plan. This plan associates X, equipped with a measure μ, to (x_i), equipped with a discrete measure ν, such that $\nu(x_i) = \mu(Pow^{\phi}(x_i))$.

This implication is due to the fact that that a power diagram minimizes the cost of transport under constraints of capacity.

1.2.2 Numerical algorithms

Most numerical algorithms for computing semi-discrete optimal transport plans amount to maximizing the functional given in Equation 1.5, with different optimization strategies.

The gradient of Φ is

$$\nabla_{\phi_i}\Phi((\phi_i)_i) = p_i - \mu(Pow^{\phi}(x_i))$$

and the coefficients of its hessian $\nabla^2\Phi$, used in the second-order algorithm presented in Section 1.2.2, are given by

$$\begin{aligned} \frac{\partial^2\Phi(X)}{\partial\phi_i^2} &= -\sum_{k=1}^{K_i} \frac{1}{2\left\|x_i - x_{i_k}\right\|} \int_{x\in E_{i,i_k}} d\mu(x) \\ \frac{\partial^2\Phi(X)}{\partial\phi_i\,\partial\phi_j} &= \frac{1}{2\left\|x_i - x_j\right\|} \int_{x\in E_{i,j}} d\mu(x) \end{aligned} \tag{1.6}$$

for x_j neighbor of x_i, where $E_{i,j}$ is the frontier between cells $Pow^{\phi}(x_i)$ and $Pow^{\phi}(x_j)$, and i_k is the index of the k-th neighbor of $Pow^{\phi}(x_i)$. The interested reader can for example refer to [KMT16] for a complete computation of these expressions.

First-order algorithms

One can show, as in [AHA98], that the functional Φ is the lower envelope of a family of hyperplanes and is thus, by application of the envelope theorem [MS02], concave and smooth. Consequently, it lends itself well to a gradient descent algorithm to maximize it (or, more accurately, a gradient ascent).

Since the expression of $\nabla_{\phi_i}\Phi$ corresponds to the (signed) gap between the mass $\mu(Pow^{\phi}(x_i))$ of a power cell and its prescribed mass p_i, a gradient descent algorithm amounts to increasing the weight of cells with too little mass

and decreasing that of cells with too much mass. Aurenhammer et al. also show that this gradient descent algorithm has a complexity of $O(N \log(N))$ for computing each descent step.

Several improvements of this gradient descent algorithm have been proposed. One of them is the multiscale algorithm of Mérigot [Mér11]. The idea of this algorithm stems from the importance of the choice of a starting point when computing optimizations, and in particular semi-discrete transport plans: a poorly chosen starting point can greatly slow down the convergence. This algorithm relies on the iterative simplification of the target measure, by constructing a coarsening sequence of discrete measures. In practice, the supports of these measures are constructed using Lloyd's algorithm. The mass of a site in a simplified measure is the sum of the masses of the previous sites belonging to its Voronoi cell constructed with Lloyd's algorithm: mass is preserved in a compatible way over the decomposition. A sequence of semi-discrete transport plans is then computed from the continuous measure to each of the discrete simplified measures. The algorithm starts with the coarsest measure, at each step using the optimal power diagram weights computed for a discrete measure to initialize the weights of the transport to the next finer one. More precisely, the weight of a site in the finer measure is set to the weight of its associated site in the coarser one. One advantage of this method is that it is not dependent on the algorithm used to compute semi-discrete transport plans: a new improved method for transport computation would result in an improvement of the multilevel algorithm. In practice, the speedup with regards to the classical method can reach up to an order of magnitude. The article also presents a theoretical guarantee that the sequence of intermediate power diagram weights converges to the weights of the actual semi-discrete transport plan.

Lévy [Lé14] discusses how to concretely implement such a multiscale algorithm, using a quasi-Newton method to compute the transport plans. The main practical aspect of this implementation is the computation of integrals over a power diagram intersected with a simplicial (here, tetrahedral) mesh. This is done by iterating over all cell-tetrahedron intersections. One starts with an arbitrary intersection, and maintains a stack of such intersections that is updated at each step with the neighboring cells and tetrahedra of the current intersection. This algorithm can be further sped up by spatially sorting the points. One can thus iteratively compute the contributions of each intersection to the transport objective function and its gradient, which can then be used by a quasi-Newton algorithm to compute the semi-discrete transport plan.

KMT algorithm

Kitagawa, Mérigot and Thibert [KMT16] present a Newton algorithm with proven convergence for computing semi-discrete transport. This algorithm involves computing the inverse (or at least the pseudo-inverse) of the hessian (Equation 1.6) of the transport functional Φ in order to compute a descent direction. Enforcing a lower bound on the areas of the cells allows to ensure sufficient regularity of Φ.

Informally, this algorithm consists in computing a descent direction using the hessian, then evaluating the maximum stepsize that will decrease the norm of the gradient without creating cells of measure below a certain threshold.

Algorithm 1 KMT algorithm for the computation of semi-discrete optimal transport plans.

Input: A tolerance η and an initial set of weights ϕ^0 such that

$$\varepsilon_0 = \frac{1}{2}\min\left[\min_i \mu(\mathrm{Pow}^{\phi^0}(x_i)), \min_i p_i\right] > 0$$

1: **while** $\left\|\nabla_{\phi_i^k}\right\| \geq \eta$ **do**
2: Compute descent direction $d_k = \left(\nabla^2\Phi(\phi^k)\right)^{-1}(\nabla_\phi\Phi(\phi^k))$
3: Compute step of form 2^{-l} such that $\phi_l^k := \phi^k + 2^{-l}d_k$ satisfies

$$\begin{cases} \min_{1\leq i\leq N} \mu(Pow^\phi(x_i)) \geq \varepsilon_0 \\ \left\|(\nabla_\phi\Phi(\phi_l^k))\right\| \leq (1-2^{-(l+1)})\left\|\nabla_\phi\Phi(\phi^k)\right\| \end{cases}$$

4: $\phi_i += d * 2^{-l}$
5: $k++$
6: **end while**

As we can see in Algorithm 1, this algorithm requires that at each step, the measures of the power cells do not fall below a certain threshold. In order for the KMT algorithm to find a step size such that cells are sufficiently large, it requires to be at least initialized with cells respecting this condition. This is not trivial, in particular when sites x_i are not all located on the support of μ. In this case, there is a significant risk of their Voronoi diagram displaying cells with zero mass. We next see approaches that aim at addressing this problem through an initialization step.

Initialization of KMT algorithm Meyron [Mey19] introduces three different initialization methods to start the KMT algorithm without empty cells.

The first is a perturbation method, where one slightly modifies the weights of the cells until none is empty. This amounts to performing a gradient descent with a very loose stopping criterion, and, in terms of complexity, can defeat the purpose of using an efficient algorithm such as KMT.

This is similar to the two-stage algorithm proposed by Bansil [Ban19], where the execution of the KMT algorithm is prefaced with a regularized gradient descent, which allows the overall algorithm to run in polynomial time on disconnected source measures.

The second method takes advantage of the fact that we know explicit formulas, given in Proposition 3, for translating and rescaling the cells of a power diagram by applying a transformation to the weights defining it. It consists in tweaking these weights until all cells intersect with the support of the source measure. However, as is, this method can prove counter-productive when the source measure support is not connected, since it can only relocate the cells to a single connected component.

Proposition 3. *Given a point set (x_i), a scaling factor $\lambda > 0$ and a translation vector t, if we define the point set (x'_i) with $x'_i = \lambda x_i + t$, then there is the following relation between the* power *cells of the two sets*

$$\mathrm{Pow}^{\phi'}(x'_i) = \mathrm{Pow}^{\phi}(x_i)$$

where $\phi_i = \frac{\phi'_i}{\lambda} + 2x'_i \cdot t + (\lambda - 1)\left\|x'_i\right\|^2$ (from [Mey19]).

At last, the third method is an interpolation one: it consists in choosing a superset of $Supp(\mu) \cup \{x_i\}_i$, equipping it with the Lebesgue measure, and iteratively computing the optimal weights for a sequence of measures linearly interpolating this uniform superset to the final measure μ. This allows to use the KMT algorithm for each weight computation, since the computed diagrams never present empty cells.

Performances of the KMT algorithm KMT algorithm is very efficient in practice when used to transport from a connected domain to a point cloud, typically converging in a dozen iterations. In fact, it is mathematically proven that it converges globally with linear rate. In the case of a non-connected domain, while this algorithm still manages to compute a transport plan, it needs to use a smaller step size to "pass" power cells from one connected

component to another without creating empty cells. This significantly slows down the convergence of the algorithm.

Figure 1.3 compares the evolution of the power cells during the course of the algorithm on two different source domains with same mass. One of these domains is connected, the other is disconnected, and the transport is computed to the same set of sites. With the connected domain, the convergence is prompt and smooth, while it slows down on the disconnected domain as soon as cells need to be transferred from one connected component to the other. The applications developed in this thesis are typically interested in disconnected domains, which makes this algorithm ill-suited to compute transport plans in this context.

Transport between a simplex soup and a point cloud Mérigot et al. [MMT17a] present an adaptation of the KMT algorithm in the case where the source measure is supported by an union of simplices, with typically some of these simplices being of lower dimension than the ambient space. The measure over the simplex soup is assumed to be the sum of density measures over its simplices, and the point cloud is assumed to fulfill a generic condition. Under these conditions, they prove that the damped Newton algorithm converges with linear speed.

Other algorithms for the computation of semi-discrete optimal transport

Balzer [Bal09] presents a different way to compute a power diagram. The weights are set one at a time to the value canceling their error function (given by gap between the current cell size and its capacity constraint). The originality of the method lies in the fact that, instead of computing the canceling value using a descent algorithm, it uses the false position method to compute the root of the error function. Although this method converges in practice, it does not have any convergence guarantees.

1.3 Alternating algorithms in semi-discrete optimal transport

A number of problems in computer science and applied mathematics can be cast either as optimal quantizing or optimal clustering problems. These problems can be expressed as a combination of some sort of transport cost minimization constraints (either with capacity constraints or not) and of

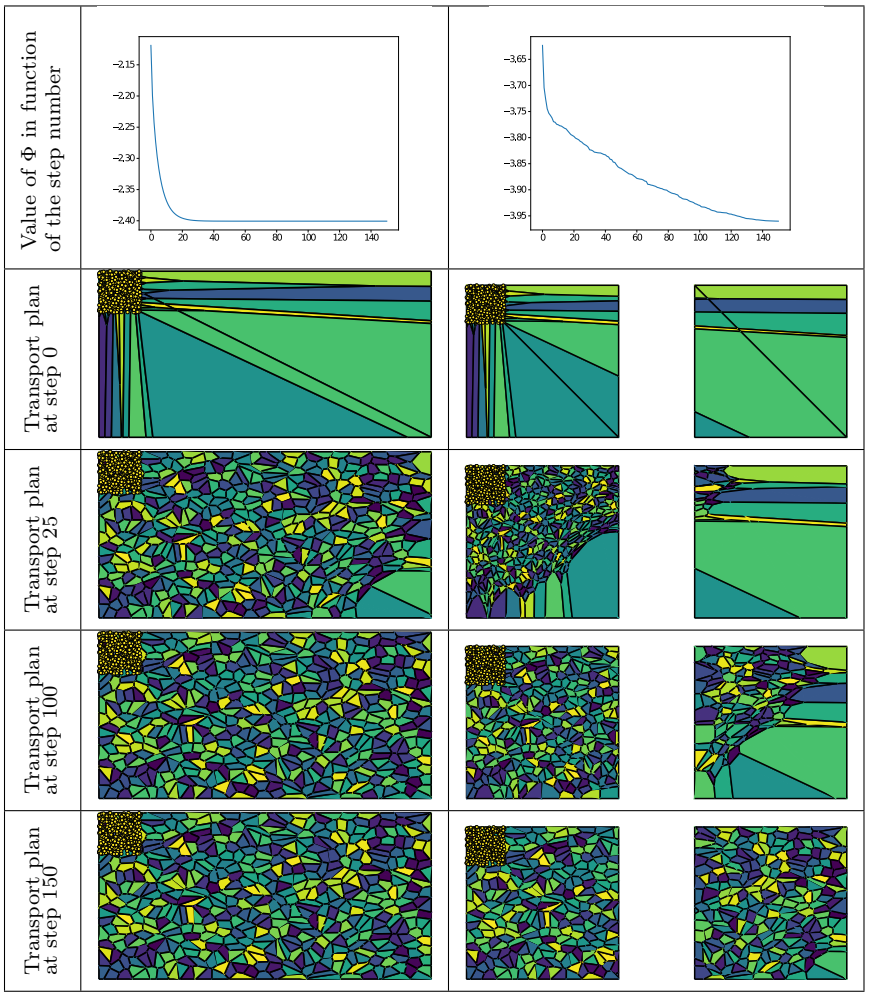


Figure 1.3: In the left column, the KMT algorithm computes an optimal semi-discrete transport plan from a connected rectangle to a sampling of its top left corner, while in the right column, the source mesh is made up of two disconnected squares, totaling to the same area. Both transport plans are initialized as Voronoi diagrams. At the 25^{th} iteration, the algorithm is almost converged in the connected case, while in the non-connected one most of the cells are still located on the same connected component as they initially were. This slow convergence can also be seen in the plotting of the objective function on the top row, where the convergence is much faster with the connected source.

barycenter constraints. The transport constraints can be fulfilled either by Voronoi diagrams (in case there are no capacity constraints) or power diagrams (in case there are). Examples of such problems without capacity constraints, such as the optimal sampling of a color space or approximation of integrals, are developed in [DFG99].

A classical solution framework for solving these problems is to alternatively enforce the transport constraints and the barycenter constraints until some convergence criterion is reached. These alternating algorithms serve as an inspiration for the algorithm we present in Chapter 3.1.1.

1.3.1 Lloyd's algorithm

Centroidal Voronoi Tessellations (CVTs) are a special class of Voronoi diagrams where the location of the sites coincides with the weighted barycenters of their Voronoi cells. They can be defined over a discrete as well as a continuous underlying domain, the continuous case being of more interest to us for its analogy with semi-discrete transport. The barycenters positions in a CVT yield samples distributed according to the underlying density.

These properties make CVTs particularly suited to applications in varied scientific fields, as reviewed in [DFG99].

Algorithms

The best-known algorithm for computing CVTs is Lloyd's algorithm [Llo82], an iterative algorithm where a step of computation of the Voronoi diagram alternates with a recentering step.

Algorithm 2 Lloyd's algorithm for the computation of Centroidal Voronoi Tessellations.

Input: A set of sites x_i
  while Convergence criterion is not met **do**
    Compute the Voronoi cells $Vor(x_i)$ of sites x_i
    Relocate each site x_i at the weighted barycenter of its cell $Vor(x_i)$
  end while

Lloyd's algorithm can be interpreted as a fixed point iteration for the function mapping a set of sites to the barycenters of the associated Voronoi cells, called the Lloyd map.

A probabilistic variant of Lloyd is presented by Linde et al. [LBG80].

Du and al. [DFG99] propose a k-means algorithm for the discrete setting, with both a randomized and a deterministic variant, which iteratively inserts

new points in the Voronoi cells and updates the positions of the barycenters. Contrarily to Lloyd's algorithm in its classical form, this algorithm does not involve explicitly computing the Voronoi diagram at each step, as it is continuously updated by an insertion at each step.

Convergence

Du et al. [DFG99] show that a CVT can be interpreted as the minimizer of the objective function: the fixed points of Lloyd's map are its fixed points.

$$(x_i)_i \mapsto \sum_{i=1}^{N} \int_{x \in Vor(x_i)} \|x - x_i\|^2 \, d\mu(x) \tag{1.7}$$

The value of this function diminishes at each step of Lloyd's algorithm, as shown in Lemma 2.2 of [DEJ06], from which we can deduce that the algorithm converges.

Function 1.7 is $\mathcal{C}^2$, which points in the direction of gradient descent and Newton's method. Such optimization algorithms are described in [DFG99] and [LWL+09].

Properties

The cells of a CVT tend to become regular hexagons [Tó01] when the underlying density is uniform. We will encounter a variation of this fact in our results.

1.3.2 Capacity-constrained Centroidal Power Diagrams

In the same way that power diagrams are an extension of Voronoi diagrams, centroidal power diagrams generalize centroidal Voronoi diagrams. To make up for the extra degree of freedom introduced by the weights of the power diagram, the problem data includes a set of capacity constraints. The resulting objects are called Capacity-Constrained Centroidal Power Diagrams (CCCPDs).

Balzer et al. [BSD09] give a method similar to the k-means method for computing CVTs on a discrete grid, where one builds the power cells of the capacity-constrained power diagram pixel by pixel. This method has a higher per-step complexity than Lloyd's, which makes it unsatisfactory to compute CCCPDs.

A better algorithm, inspired by Lloyd's, is presented in [DGBOD12], where a step of computation of the power diagram fulfilling the capacity constraints

(considered as a semi-discrete transport problem) alternates with a step of sites recentering. This algorithm has linear convergence, with its main weak point being the computation of the barycenters (as pointed out in [XLC+16]). The sites generating a capacity-constrained centroidal power diagram have blue noise properties. de Goes et al. [DGBOD12] take advantage of this to compute high-quality samplings of gray scale images.

An improved method is the optimization presented by Xin et al. [XLC+16]. It builds on the notion that computing a CCCPD amounts to maximizing the transport functional with respects to the weights and minimizing it with respects to the sites positions. Observing that, given the sites' positions, the weight are uniquely defined (up to an additive constant), one can make the transport functional a function of only the sites positions. One can then use L-BFGS [Noc80] to optimize this functional.

Balzer [Bal09] studies a similar object: constrained centroidal Voronoi diagram. These are Voronoi diagrams such that the area of the cells is determined by a constraint. Since Voronoi diagrams do not enjoy the extra degree of freedom provided by the power weights, one has to optimize over the sites' positions in order to fulfill these constraints. Because optimizing over all the sites would be too computationally demanding, the presented algorithm optimizes one site at a time over the course of an iteration. The cell for which the difference between its actual area and its capacity constraint is the lower is selected, and its site is moved in the direction of its most oversized (or least undersized) neighbor.

1.4 Applications of semi-discrete optimal transport

1.4.1 Displacement interpolation approximation

Bruno Lévy [Lé14] proposes an algorithm based on semi-discrete optimal transport to approximate the displacement interpolation between simplicial (triangular or tetrahedral) meshes.

The algorithm consists in sampling one of the meshes and transporting the other to the samples. The vertices of the interpolated mesh evolve between these samples and the centroids of the power diagram arising from the transport. The edges and simplices are the ones of the Delaunay triangulation (or tetrahedrization) of the samples.

This algorithm computes an interpolation between two subsets of the initial

Algorithm 3 Algorithm approximating displacement interpolation through semi-discrete optimal transport (taken from [Lé14]).

Input: Two simplicial meshes X and X' and a number of samples N

Output: A simplicial mesh with N vertices and a pair of points x_i and c_i attached to each vertex.

(1) Sample X' with set of points $\{x_i \mid 1 \leq i \leq N\}$

(2) Compute the weight vector $(\phi_i)_i$ that realizes the optimal transport between X and $(x_i)_i$

(3) Set the barycenters $c_i \leftarrow centroid(Pow^{\phi}(x_i) \cap X')$

(4) Compute the Delaunay triangulation of the sampling $(x_i)_i$ and mesh (c_i) using the same triangulation

(5) Remove the triangles that are not included in X at $t = 0$

(5') Remove the triangles adjacent at $t = 1$ to a site x_i such that $Pow^{\phi}(x_i)$ spans several connected components

(6) Interpolation X_t has vertices $t \cdot x_i + (1 - t) \cdot c_i$, $t \in [0, 1]$

meshes. It has the advantage of allowing interpolation between different total masses. Indeed, the total mass affected to the sampling of X' doesn't have to match with the actual mass of X' but with that of X.

The biggest limitation of this algorithm is that, by constructing the interpolation from a sampling, it necessarily erodes the initial meshes. In particular, a consequence is that the potential discontinuities in X are not taken into account, since the sampling is done on X'. This usually results in jagged boundaries where the discontinuities occur — see Figure 2.4 for an illustration. This will be developed more in Chapter 3.

1.4.2 Area preserving mappings

Zhao et al. [ZSG+13] use optimal transport to construct area-preserving mappings between a topological surface and a planar surface. When mapping a surface to the plane, it is usually difficult to preserve both angles and areas: one has to make a tradeoff. This article aims to find mappings destined to applications where area preservation is crucial, such as a number of medical applications.

This method builds upon a previous algorithm by Dominitz and Tannenbaum [DT10] that used discrete optimal transport to transform angle-preserving mappings into area-preserving mappings. The originality of the new approach lies in the use of semi-discrete optimal transport, which reduces computational cost, as already discussed in this chapter.

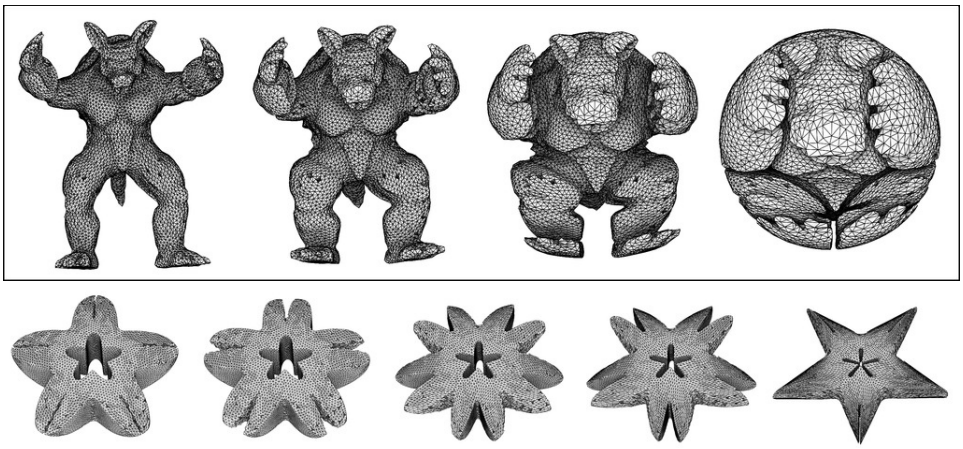


Figure 1.4: Interpolation between a sphere and an armadillo, and between two stars. Image taken from [Lé14].

Given an input surface represented by a mesh, one computes an angle-preserving mapping from it to the unit euclidean disk using an already-established method. A semi-discrete transport map is then computed between the uniform unit disk and the vertices of the mesh each endowed with the area of their neighborhood in the image of the initial mapping. This step allows to quantify the area distortion induced by the angle-preserving mapping.

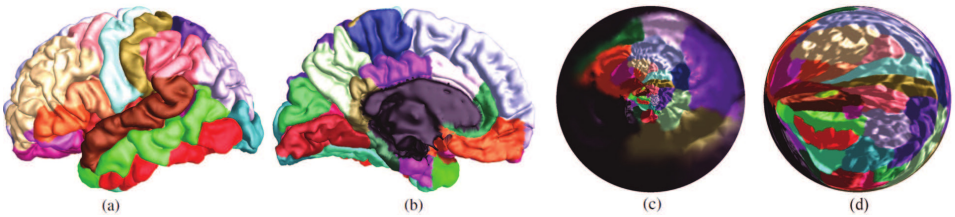


Figure 1.5: Angle-preserving mapping (c) of the surface of the brain (a) and (b), compared with the area-preserving mapping (c). Image from Zhao et al. [ZSG+13]

The final mapping is made up of the composition of the initial angle-preserving mapping, and of the semi-discrete transport map that corrects it for area preservation.

1.4.3 Fluids simulation

Semi-discrete transport can be used to simulate the behavior of continuous physical systems such as fluids or galaxies at a very large scale.

Fluids with free boundaries

In order to simulate fluids, Bruno Levy [Lé22] decomposes them into power cells through partial semi-discrete optimal transport. Modeling the fluid elements with power cells yields a Lagrangian representation of the fluid. The fixed masses of the cells model the incompressibility of the fluid: while a fluid element can be deformed and split at will, its total volume has to stay constant. Partial optimal transport differs from classical in that the sum of the discrete masses is lower than the total mass of the source: the power cells only cover part of the domain. This models the fact that the fluid has a free surface in contact with the ambient air.

In order to make up for the fact that the source and target total measures do not match, an extra target element is added, in the form of an uniform sampling of the whole domain. All the samples are equipped with the same weight, forcing their part of the diagram to effectively be a Voronoi diagram, except at the interface with the fluid where they may be truncated by the fluid cells. This models the fact that it does not "cost" anything for the fluid to displace the air.

One then can make this uniform sampling tend to a continuous uniform distribution, which is equivalent to removing the aforementioned Voronoi cells and instead adding the constraint $\|x - x_i\|^2 < \phi_i$ to the cell's definition, making it the intersection of a classical power cell and of a ball.

This problem can be solved using a variant of the KMT algorithm. The main differences with its classical version come from the new definition of the cells, which does not modify the essence of the gradient expression (it remains the gap between the prescribed mass of a cell and its actual mass) but does add a term for the Hessian. The rest of the algorithm, from the stepsize computation to the stopping criterion, remains unchanged.

An example showing the effects of surface tension and the handling of topology changes can be seen in Figure 1.6, which shows the evolution of a pool of fluid in which a drop falls.

Early Universe Reconstruction

The behavior of galaxy clusters at a very large scale under the influence of gravitation can be interpreted as the behavior of a fluid. Knowing that, it is

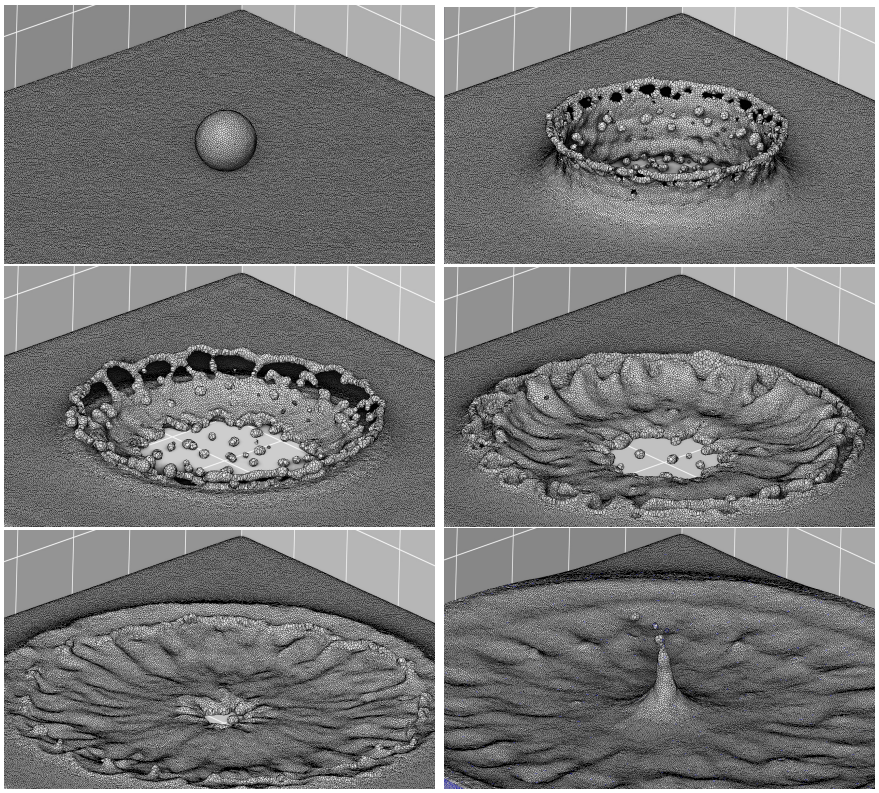


Figure 1.6: Simulation of a drop falling into a pool of fluid using partial optimal transport. Image from [Lé22].

no surprise that optimal transport has been used to simulate the evolution of their positions from the early times of the universe to present day.

Lévy et al. [LMvH21] propose a new take on this method by designing a semi-discrete optimal transport algorithm to reconstruct the early stages of the universe.

Just after the Big Bang, light and matter were too densely packed for light to circulate. Both light and matter were exerting pressure over each other, with small heterogeneities in their distribution. After the universe expanded enough to allow light to travel (photon decoupling), these heterogeneities persisted and self-maintained, the densest regions having higher gravitational potential and in turn attracting more matter. These heterogeneities can be observed in two different structures today: first, in the cosmic microwave background, emitted at the time of photon decoupling, and second in the structure of the repartition of galaxy clusters today.

Thus, there is a direct evolution path from the early repartition of matter to what it is in present day. Lévy et al. pose the equations governing the evolution of matter, and show that these can be formulated in terms of optimal transport. This stems from the application of the least action principle, which corresponds to the optimization aspect of optimal transport, under matter conservation constraints, which are present in optimal transport as the pushforward constraint.

In the case of this article, the positions of galaxy clusters are discretized and considered the target distribution, while the continuous source distribution is considered uniform. The objective is to compute the transport plan from the latter to the former, modeling the evolution of the system, which one can then compare to a ground truth stemming from cosmic microwave background.

This method uses the KMT algorithm in order to compute its semi-discrete transport plan. The computational efficiency of semi-discrete transport in general and of the KMT algorithm in particular allows to compute discretizations of a million points in a few minutes on an ordinary computer, where the same level of discretization would have taken months with a fully discrete formulation.

In addition, the computed power diagram weights hold an interesting physical interpretation as the gravitational potential of the associated point.

1.4.4 Optics

Light in Power [MMT17b] discusses how semi-discrete optimal transport can be used to model inverse problems in optics. More specifically, given a light source, either punctual or collimated, and a light intensity distribution, one wishes to design lenses or mirrors whose illumination corresponds to this distribution. This is an inverse problem to the direct problem of computing the illumination yielded by a light source passing through a lens or reflecting over a mirror.

The key to modeling such a problem with optimal transport is to express it in terms of conservation of the light energy, which finds a natural translation into the mass conservation constraint of optimal transport.

In this framework, the light source is considered continuous, assimilated either to $\mathbb{S}^2$ (in the case of a punctual source) or to $\mathbb{R}^2$ (in the case of a collimated source), and equipped with the adequate density. The final illumination is discretized. Semi-discrete transport is computed between the source and the target illumination, and the resulting power cells are used to compute a parameterization of the lens or mirror surface. In the

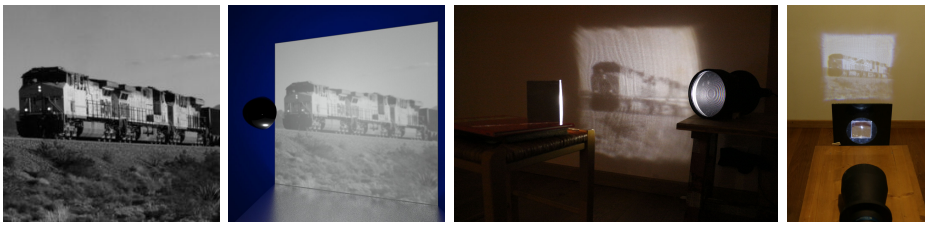


Figure 1.7: From left to right: target image; simulation of the illumination from a mirror reflecting a punctual source (located inside the mirror); illumination from a fabricated mirror reflecting a collimated source; illumination from a fabricated lens refracting a collimated light source. Image taken from [MMT17b].

case of a collimated source, the mirror and the lens both have a piecewise linear surface; in the case of a punctual source, the mirror is made up of paraboloid sections, while the lens is made up of ellipsoids. In all four cases, the parameters of these surfaces are computed from the weights defining the semi-discrete transport.

1.4.5 Texture enrichment

Galerne et al. [GLR17] use semi-discrete optimal transport in patch space to enrich generated textures.

Their method builds upon Gaussian texture generation, where an example texture is represented as a Gaussian vector, and the generated texture is made up of patches drawn from that distribution, by computing a convolution between the original patch and white noise.

This technique alone generates textures that have the same local statistical properties as the example texture, but do not preserve larger structures.

In order to solve this issue, Galerne et al. rework the generated texture. First, an optimal transport plan from the space of generated patches to the actual patches of the example images (or, more realistically, a sampling of them) is computed. Then, for each pixel of the generated texture, each patch containing said pixel is projected onto an example patch according to the transport plan, and the pixel is assigned a new value computed as the average of its values in these projections.

This technique enjoys better spatial stability than its concurrents, including a similar method that uses simple nearest-neighbors instead of optimal transport. Long ranges correlations are better respected, while the local statistical properties are still similar to the example's. Examples of results

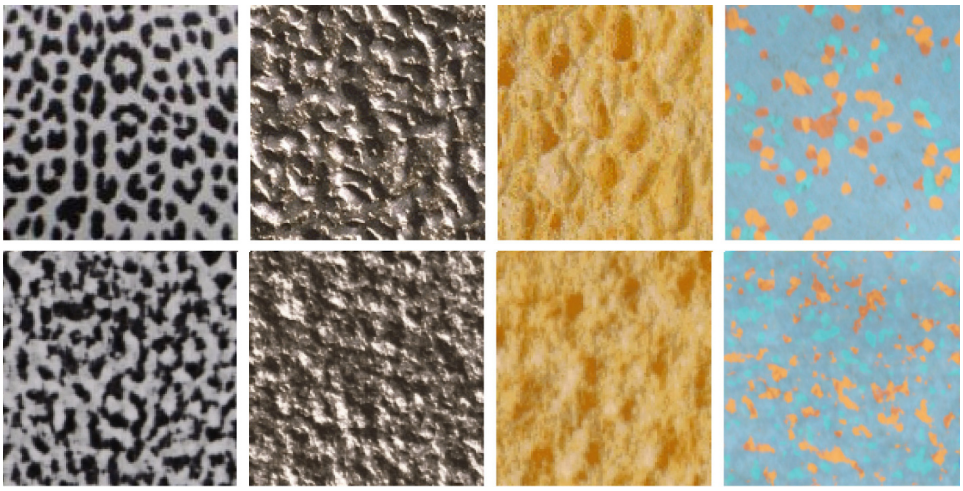


Figure 1.8: Exemplar texture (first line) and generated texture (second line). Image from [GLR17].

using this technique can be seen in Figure 1.8, with the source textures and the generated textures presented side-to-side. This method has better performance on stochastic and near-stochastic textures than on regular textures, even when increasing the patch size.

1.5 Other applications of optimal transport

The earth mover's distance is used by Rubner et al. [RTG04] as a metric to compute perceptually accurate distances between images. Image retrieval often uses characteristics of images represented as histograms, such as brightness (one-dimensional histogram), color (three-dimensional histogram) or texture (frequency content), to compare the contents of images. Rubner et al. show that earth mover's distance yields better results in comparing images from this point of view. In particular, this distance allows to regroup histogram bins and thus similar features, that are not perceived as fundamentally different. The earth mover's distance is also well-adapted to partial matching, which is particularly useful when matching only parts of an image.

Genevay et al. [GPC18] use optimal transport to evaluate how a generative model fits its training data. The Wasserstein distance is a particularly well adapted metric to compare distributions in this context. In order to make the computation of transport distances tractable, they use a regularized version

of the Wasserstein distance (see [Cut13]) for their loss, under the name of Sinkhorn loss. They present a numerical scheme involving autodifferentiation of the loss to learn under this metric. Applications include all sorts of generative models, in particular image generation.

In a similar approach, Arjovsky et al. [ACB17] use the earth mover's distance as a metric for GANs to learn probability densities. Compared with other usual metrics used for GANs, the earth mover's distance allow more sequences to converge, which in turn improves the convergence of the GANs.

Rabin et al. [RDG10] use OT for color transfer. Given a style image and its extracted color palette, given as a histogram, one is looking for a function operating over images that transforms a source image so that its color histogram matches that of the style. This problem can be cast as an optimal transport problem: the transformation function has to turn a distribution into another while minimizing the amount of actual changes. Naive solutions to this problem often result in artifacts, due to unwanted decrease or enhancement of contrast (this last point can be particularly problematic in the case where the enhanced features are compression artifacts or noise). Their paper proposes a method that suppresses these artifacts through a regularization operator that is applied to the optimal transport map.

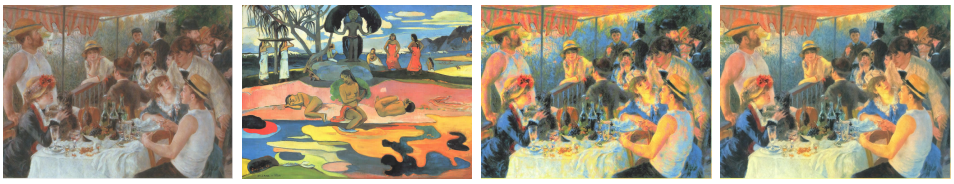


Figure 1.9: Example result for OT-powered color transfer with Rabin et al. [RDG10]'s method. From left to right: source image (Auguste Renoir, *Le déjeuner des canotiers*, 1881), style image (Paul Gauguin, *Mahana no atua — Le jour de Dieu*, 1894), simple color transfer, color transfer with regularization.

Delon et al. [DSS10] present an algorithm to compute optimal transport between measures defined on the unit circle. Their algorithm takes advantage of the fact that one can lift the measures on the circle to periodic measures on $\mathbb{R}$, and extend the cost function as well. This type of transport also has applications in image processing: measures on the circle can represent histograms, which are of use in color transfer, where color is represented in the Hue, Saturation and Lightness model, or in image retrieval, where distributions characterizing images can be defined on the circle.

Digne et al. [DCSA+14] use optimal transport as a guide for measuring the fitness of a mesh reconstruction to a given point cloud. Their method starts from the Delaunay triangulation of the point cloud in question, which is then iteratively simplified. At each step, the simplification operation is chosen to minimize the increase in the value of the transport from the point cloud to the candidate reconstructed mesh. This method is robust to noise and outliers, and is very efficient at recovering sharp features.

1.6 Conclusion

Semi-discrete optimal transport is a formulation of optimal transport with strong ties to geometry, through the interpretation of transport plans as power diagrams. Transport plans can thus be encoded as the vector of the weights defining the power diagram. The weights for which the transport cost is minimal realized the optimum of a convex objective function, which allowed for the development of several efficient optimization algorithms to compute them.

However, in spite of its numerous qualities, semi-discrete optimal transport has the substantial drawback of being fundamentally non-symmetrical. Next chapter will propose a way to bring back symmetry while retaining its advantages.

Furthermore, in this chapter, we presented alternating algorithms for computing optimal samplings and partitions of continuous domains, which we will use as inspiration for an algorithm for computing symmetrized semi-discrete transport. We will also look into extending the optimization algorithms for semi-discrete transport to its symmetrized formulation.

Chapter 2

Problem statement

It is known that optimal transport maps transporting an uniform measure supported by any bounded shape to an uniform measure supported by a convex shape are continuous [Caf92], and that, conversely, optimally transporting from a convex shape to a shape whose boundary presents regions of sufficiently negative total curvature will result in a transport plan with discontinuities [CJL$^+$15], as illustrated in Figure 2.1. If we want to propose an approximation of displacement interpolation, an important point is that it should capture well such discontinuities.

We consider two measures μ et ν whose support is represented by simplicial meshes. Our goal is to compute an accurate approximation of the displacement interpolation between μ and ν, that, in particular, properly captures potential discontinuities. Unlike most approximations of optimal transport, our approach does not simply discretize both meshes and compute a discrete transport plan between sample sets, but instead takes advantage of the nice properties of semi-discrete optimal transport. Indeed, as de-

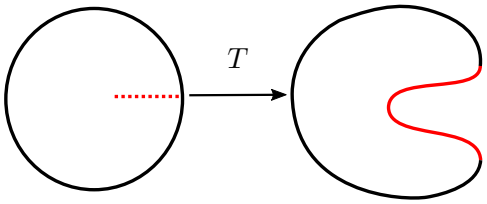


Figure 2.1: The optimal transport plan T between 2-d shapes maps the red dotted line on the source onto the red line on the target, which has total curvature below a threshold value. This mapping is discontinuous precisely on this line.

veloped in Section 1.2.2, very efficient algorithms exist to compute exact semi-discrete optimal transport, whereas discrete OT demands either an algorithm of cubic-time complexity (if exactness is required), or an algorithm that sacrifices exactness by using regularization.

We propose an algorithm that lies in the continuity of other semi-discrete approximations of displacement interpolation, where one of the continuous domains is transported to the barycenters of an optimal partition of the other, similar to Mérigot [Mér11] or Lévy [Lé14].

2.1 Non-symmetrized transport

The algorithm of Lévy [Lé14] can be used to compute approximations of the displacement interpolation between two meshes X and X'. To compute this approximation, mesh X' is sampled. An optimal transport plan from X to the sampling of X' is then computed, and is represented by a power diagram. The interpolation is done between the samples of X' and the barycenters of the power diagram, located on X. See Figure 2.2 for an illustration.

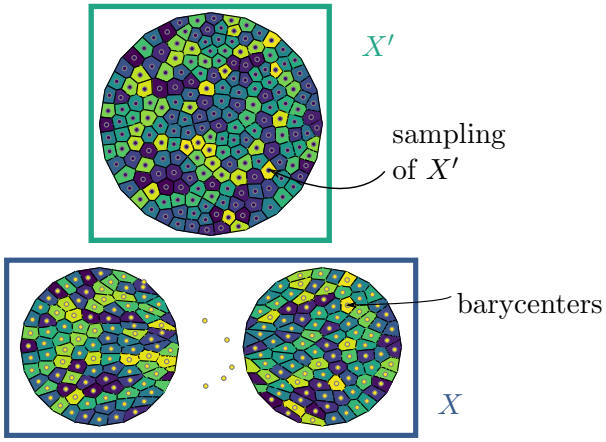


Figure 2.2: X is optimally transported to a sampling of X', and this sampling is interpolated to the barycenters of the transport plan.

The simplices of the interpolated mesh are retrieved from the Delaunay triangulation (or tetrahedrization, in the 3d case) of the sampling of X'.

The main weakness of this approach lies in the difficulty to capture all the properties of the input meshes it aims to interpolate, in particular when it comes to their discontinuities.

Algorithm 4 Algorithm approximating displacement interpolation through semi-discrete optimal transport (taken from [Lé14]).

Input: Two simplicial meshes X and X' and a number of samples N

Output: A simplicial mesh X_t with N vertices and a pair of points x_i and c_i attached to each vertex. Transport is parameterized by time $t \in [0; 1]$

(1) Sample X' with set of points $\{x_i \mid 1 \leq i \leq N\}$

(2) Compute the weight vector $(\phi_i)_i$ that realizes the optimal transport between X and $(x_i)_i$

(3) Set the barycenters $c_i \leftarrow centroid(Pow^{\phi}(x_i) \cap X')$

(4) Compute the Delaunay triangulation of the sampling $(x_i)_i$ and mesh (c_i) using the same triangulation

(5) Remove the triangles that are not included in X at $t = 0$

(5') Remove the triangles adjacent at $t = 1$ to a site x_i such that $Pow^{\phi}(x_i)$ spans several connected components

(6) Interpolation X_t has vertices $t \cdot x_i + (1 - t) \cdot c_i$, $t \in [0, 1]$

Since the original mesh X' is discretized through a random sampling (x_i), a triangulation of (x_i) will, in most cases, not cover all of X' — see figure 2.4 for an illustration. In the case where X' is disconnected, this triangulation contains triangles that cross the discontinuity. These triangles are filtered out in step (3) of algorithm 4. Filtering these triangles also typically results in loss of matter from X'. This results in the erosion of X' in the interpolation built from the triangulation of (x_i).

When mesh X is not connected, the algorithm does not take into account these discontinuities when sampling X'. Consequently, the locus of these discontinuities on the interpolation not only suffers from the erosion due to the sampling, but also presents jagged boundaries due to the filtering of overlapping simplices, as illustrated in Figure 2.3. These issues can be mitigated by using a CVT to sample X', in order to cover the mesh more uniformly, but it is not enough to solve them.

This is a consequence of the non-symmetric roles played by the meshes X and X' in this approach, since one is approximated by a set of samples and the other is not. This results in at most one of the two meshes' features being well approximated during the interpolation.

These erosion and jagged boundaries issues are illustrated in Figure 2.4, where the interpolating mesh at its initial and final positions are superimposed with the input meshes that it aims to approximate. See Section 5.3.1 for more examples of this interpolation.

To overcome these issues, we present in this chapter an algorithm sym-

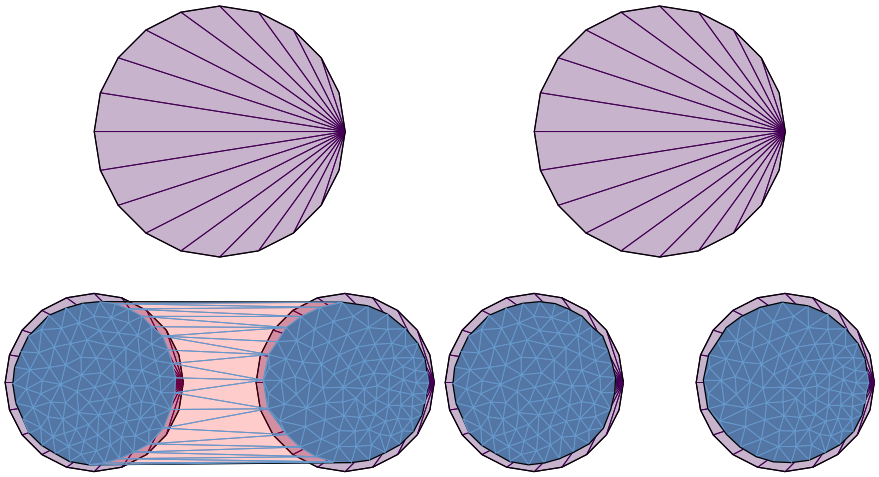


Figure 2.3: Red triangles are filtered out of the triangulation as they overlap the connected components.

metrizing Lévy's algorithm.

2.2 Coupled transport plans

In addition to the measures μ and ν, we consider two sets of samples $(x_i)_i$ and $(y_i)_i$, equipped with the discrete measures $\sum_{i=1}^{N} p_i \delta_{x_i}$ and $\sum_{i=1}^{N} q_i \delta_{y_i}$, where p_i (resp. q_i) represents the amount of mass ascribed to the sample x_i (resp. y_i) and thus held by the cell $Pow^{\phi}(x_i)$ (resp. $Pow^{\psi}(y_i)$) when the power diagram represents an optimal transport plan. These mass prescriptions should not be confused with the power diagram weights ϕ_i (resp. ψ_i) that encode the optimal transport plans from μ to $(x_i)_i$ (resp. ν to $(y_i)_i$).

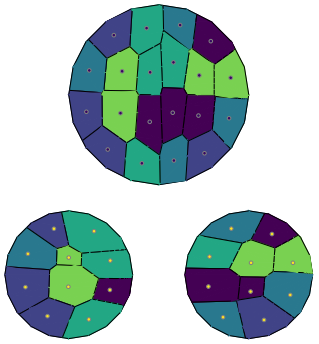


Figure 2.5: Example of the situation we wish to attain, computed with the method developed in 3.1.1

In that context, we aim at finding samples positions x_i and y_i, and weights ϕ_i and ψ_i such that x_i is the barycenter of $Pow^{\psi}(y_i)$ and y_i is the barycenter of $Pow^{\phi}(x_i)$ for all i, and power diagrams $Pow^{\phi}(x_i)$ and $Pow^{\psi}(y_i)$ respectively describe optimal trans-

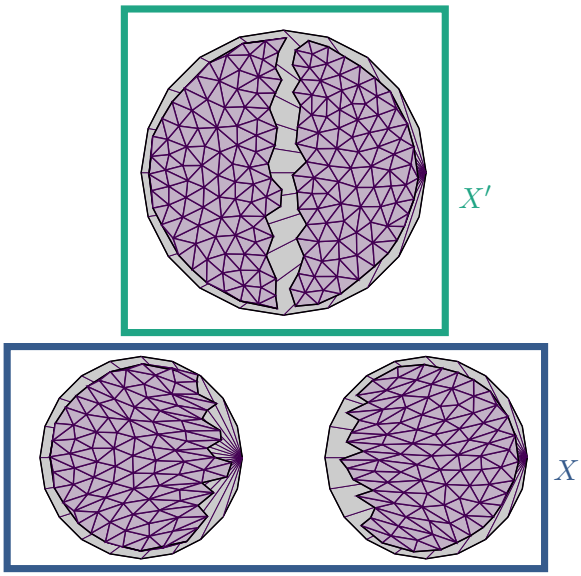


Figure 2.4: The initial and final positions of the interpolating mesh computed with Lévy's algorithm superimposed with the two meshes X and X'. We observe that the interpolations erodes the original meshes and that it presents jagged boundaries on discontinuities.

port maps from μ to $\sum_{i=1}^{N} p_i \delta_{x_i}$ and from ν to $\sum_{i=1}^{N} q_i \delta_{y_i}$, both sets of samples being equipped with the discrete uniform measure. We denote these transport maps T_ϕ and T_ψ.

In practice, we use uniform discrete measures: $\forall i, p_i = q_i = \frac{1}{N}$. Enforcing $p_i = q_i$ allows the same amount of mass to be gathered in $Pow^\phi(x_i)$ and $Pow^\psi(y_i)$. We chose this value to be uniformly equal to $\frac{1}{N}$ so that the underlying measure densities associated to μ and ν are properly captured: regions of higher density carry smaller power cells and thus are more densely sampled.

An example of such a situation is given in Figure 2.5. Note that these constraints locate samples (x_i) on the support of ν, despite the fact that they are associated to μ through the transport plan T_ϕ.

2.3 Constrained optimization

This problem can be formalized as a constrained optimization problem. The objective function is a combination of transport functionals that is maximal

when power diagrams correspond to optimal transport maps. Constraints encode the fact that the sites (x_i) and (y_i) are located at the barycenters of the respective cells $Pow^{\phi}(x_i)$ and $Pow^{\psi}(y_i)$.

We formulate it as follows:

$$\begin{aligned}\max_{(x_i,\phi_i,y_i,\psi_i)} &\sum_{i=1}^{N}\int_{x\in Pow^{\phi}(x_i)}(\|x-x_i\|^2-\phi_i)d\mu(x)+\sum_{i=1}^{N}p_i\phi_i\\ &+\sum_{i=1}^{N}\int_{y\in Pow^{\psi}(y_i)}(\|y-y_i\|^2-\psi_i)d\nu(x)+\sum_{i=1}^{N}q_i\psi_i\\ \text{s.t. }&\int_{x\in Pow^{\phi}(x_i)}(x-y_i)d\mu(x)=0\\ \text{and }&\int_{y\in Pow^{\psi}(y_i)}(y-x_i)d\nu(y)=0\end{aligned} \tag{2.1}$$

2.3.1 Intersection of sets

This problem could alternatively be cast as searching for the intersection of four subsets of the parameter space, each corresponding to one of our constraints:

$$\left\{(x_i,\phi_i,y_i,\psi_i)\ \mid\ \forall i,\mu(\mathrm{Pow}^{\phi}(x_i))=p_i\right\} \tag{2.2}$$

$$\left\{(x_i,\phi_i,y_i,\psi_i)\ \mid\ \forall i,\int_{x\in Pow^{\phi}(x_i)}(x-y_i)d\mu(x)=0\right\} \tag{2.3}$$

$$\left\{(x_i,\phi_i,y_i,\psi_i)\ \mid\ \forall i,\mu(\mathrm{Pow}^{\psi}(y_i))=q_i\right\} \tag{2.4}$$

$$\left\{(x_i,\phi_i,y_i,\psi_i)\ \mid\ \forall i,\int_{y\in Pow^{\psi}(y_i)}(y-x_i)d\nu(y)=0\right\} \tag{2.5}$$

Our problem would then amount to finding a point (x_i,ϕ_i,y_i,ψ_i) at the intersection of sets 2.2, 2.3, 2.4 and 2.5.

Computing the intersection of two or more sets can be done through an alternating projection strategy [BB96]. This method is guaranteed to converge if the sets are convex and closed, however, it can give good practical results even when these conditions are not met.

Chapter 3

Alternating algorithm

The constraints presented in Section 2.2 are very reminiscent of the ones characterizing CCCPDs, with a significant difference in that our setting includes two different domains, with cross-domains barycentric constraints. By analogy with the Lloyd-like algorithms used to compute CCCPDs, we propose a fixed-point iteration to solve this constrained problem, by alternatively optimizing the transport functional and enforcing the constraints.

3.1 Alternating algorithm

3.1.1 Algorithm

As input, our algorithm takes two measures μ and ν whose supports $Sp(\mu)$ and $Sp(\nu)$ are domains meshed with triangles (in 2-d) or tetrahedra (in 3-d). Measures μ and ν are defined with piecewise linear density functions, which are entirely given by their values on the mesh vertices and linearly interpolated over triangles or tetrahedra.

Our algorithm starts by uniformly sampling the supports of both measures μ and ν following the method described by Levy and Bonneel [LB12] that samples each simplex proportionally to its area. We then repeat until convergence the following operations.

First, we optimize the weights of $Pow^{\phi}(x_i)$, the power diagram restricted to the support of μ, describing the semi-discrete transport map between μ and $(x_i)_i$ with capacities p_i using standard semi-discrete optimal transport techniques [Lé14].

We then move each sample y_i to the barycenter $\bar{y}_i$ of the newly computed power cell $Pow^{\phi}(x_i)$, accounting for measure μ: $\bar{y}_i = \frac{\int_{x \in Pow^{\phi}(x_i)} x d\mu(x)}{\int_{x \in Pow^{\phi}(x_i)} d\mu(x)}$.

Then, we repeat the same operation by inverting the roles of the samples and measures — computing the transport map between ν and $(y_i)_i$ with capacities q_i, and centering the samples $(x_i)_i$ at the barycenter of $Pow^{\psi}(y_i)$ with respect to measure ν.

Algorithm 5 Alternating algorithm for computing symmetrized optimal transport maps

1: $(x_i) :=$ random sampling of $Sp(\nu)$
2: $(y_i) :=$ random sampling of $Sp(\mu)$
3: **while** not converged **do**
4: $\quad (\phi_i) :=$ semi-discrete optimal transport weights from μ to $\sum_{i=1}^{N} p_i \delta_{x_i}$
5: $\quad (y_i) :=$ centroids of $Pow^{\phi}(x_i)$
6: $\quad (\psi_i) :=$ semi-discrete optimal transport weights from ν to $\sum_{i=1}^{N} q_i \delta_{y_i}$
7: $\quad (x_i) :=$ centroids of $Pow^{\psi}(y_i)$
8: **end while**

We will show in Chapter 5 that even this seemingly simple algorithm that symmetrizes the notion of semi-discrete optimal transport produces displacement interpolation results that capture well discontinuous behavior in the transport maps. A sample run of Algorithm 5 on a 2-d example is shown in Fig. 3.1.

3.1.2 Implementation

Each semi-discrete optimal transport computation results in an optimization, typically performed using an iterative solver (L-BFGS in our case). For the two optimal transport optimizations performed at the first (outer) iteration, in practice we initialize transport weights with a constant value, resulting in Voronoi diagrams. For the remaining iterations we employ a warm restart strategy: the optimized values of $(\phi_i)_i$ and $(\psi_i)_i$ from the previous iteration are reused as initial guesses. We repeat these iterations a fixed number of times. Figure 3.2 shows the effect of this warm restart on the transport plans computation times.

We found that 50 iterations were enough in practice to reach convergence in all our examples. We also have found that ending the iterations of our algorithm with a transport plan computation step, rather than ending it with a recentering step, tends to yield better results with regards to the interpolation, even when the algorithm was not yet completely converged.

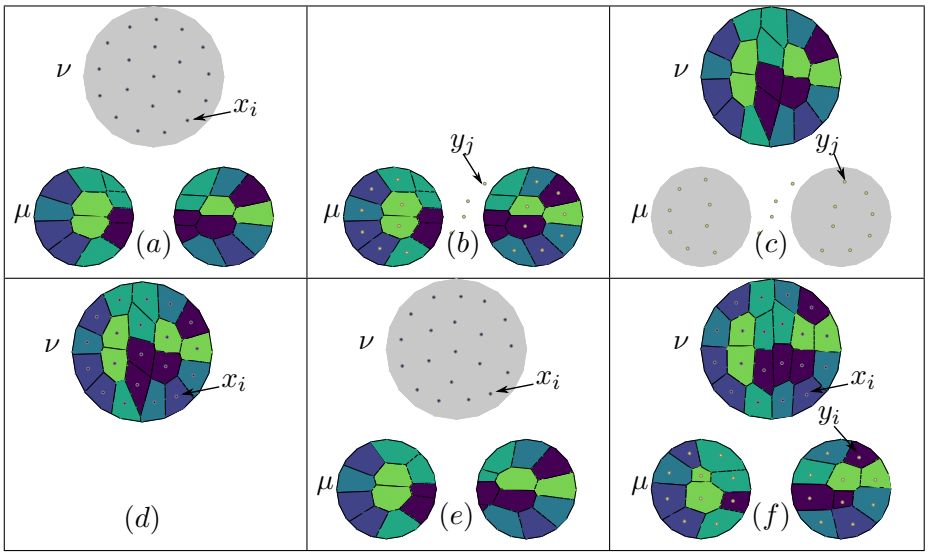


Figure 3.1: First steps of the algorithm and final setting. Steps $(a-d)$ correspond to a single iteration of Alg. 5. (a) Measure μ is transported to samples (x_i), located on the support of ν. (b) Samples (y_i) are relocated to the barycenters of the new power cells. (c) Measure ν is transported to samples (y_i), located on the support of μ. (d) Samples (x_i) are relocated to the barycenters of the new power cells. (e) Transport from μ to (x_i) for the second iteration. (f) Transport maps and samples at convergence after several iterations.

Computation of the transport maps is done using the `HLBFGS` library [Liu10], and the algorithm has been implemented within the Graphite library [Lé19]. In the particular case of computing transport plans from a mesh with non-convex connected components, our implementation using L-BFGS was not converging in practice, which forced us to use a less efficient but more robust gradient descent algorithm to compute our results. We will investigate it in the future.

Runtime analysis The overall complexity of our algorithm is dominated by the optimal transport computation, since the recentering complexity is negligible. Requiring multiple calls to optimal transport optimizations makes the overall procedure relatively costly — though of performance similar to iterative semi-discrete optimal transport computations of fluid dynamics [GM18] that perform similar iterations. Due to our warm restart,

transport map computations are typically much faster after the first iteration. This is illustrated by Figure 3.2, where statistical properties (mean, median and quartiles) of the transport computation times are presented, computed over the course of 50 experiments. Each experiment is done with a different initial placement of the sites, and the weights are always all initialized to 1 (i.e. the transport plans start out as Voronoi diagrams). Due to the potentially high values of outliers, we present these values in logarithmic scale. While some transport plan computations beyond the two initial ones can occasionally take more time, in the typical run they are much faster. In practice, in all our 2-d and 3-d examples, the entire process takes approximately 7–8 minutes for 200 samples and 27–37 minutes for 10k samples, on an Intel Xeon E5–1650 6-core machine at 3.5GHz.

3.2 Preliminary interpolations

The alternating algorithm gives as an output two point clouds in which samples are in one-to-one correspondence with each other. We can linearly interpolate between samples in correspondence, which gives us intermediate point clouds that approximate displacement interpolation.

We present in Figure 3.3 the results of such pointwise interpolation. The first four rows present the results of the algorithm on meshes that either have multiple non-connected components (first row) or locally present sufficiently negative curvature on their boundaries for displacement interpolation between them to present discontinuities. Our method seems appropriate to capture discontinuities, as the resulting interpolated points align well with the locations where we expect discontinuities in the result. The last row interpolates between two mass distributions that share the same support, but have different non-uniform densities (in this case, the sum of two gaussians placed in the opposite corners of a square). In this case as well, linearly interpolating the point clouds seems to approximate well displacement interpolation.

However good these preliminary results are, point clouds remain quite a coarse approximation of meshes. One option would be to follow the approach of Bruno Lévy [Lé14] and consider the regular triangulation of these points associated to the power diagrams computed by the algorithm. However, the properties of the coupled power diagrams allow for a better meshing strategy, which we present in Chapter 5.

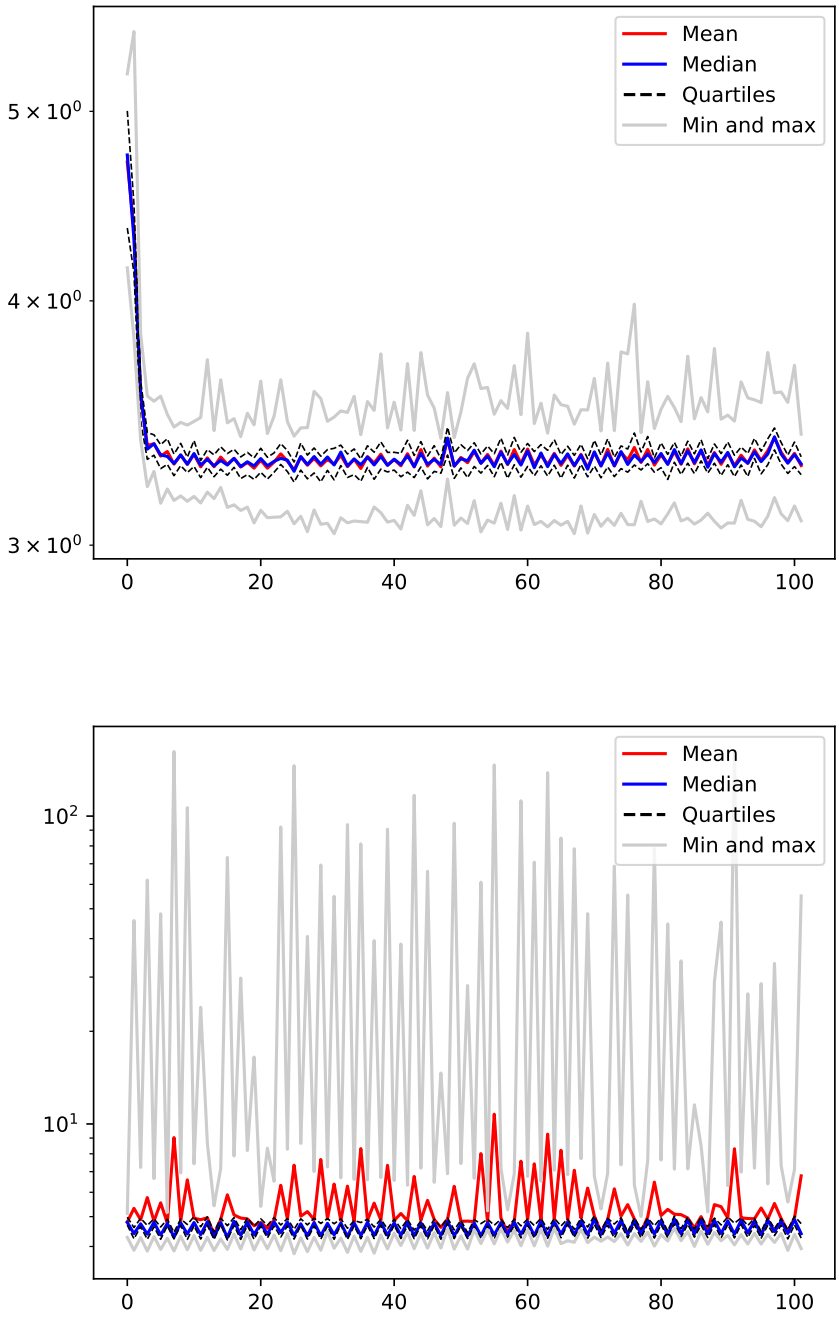


Figure 3.2: Average CPU time used for computing the transport plans in the alternating algorithm, with 50 alternating iterations (meaning 100 transport plan computations). First chart shows times with warm restart, and the second without. These statistics were computed over the course of 50 algorithm runs, when executing the algorithm for one disk transporting to two disks, as in Figure 5.6, with $N = 5000$ points.

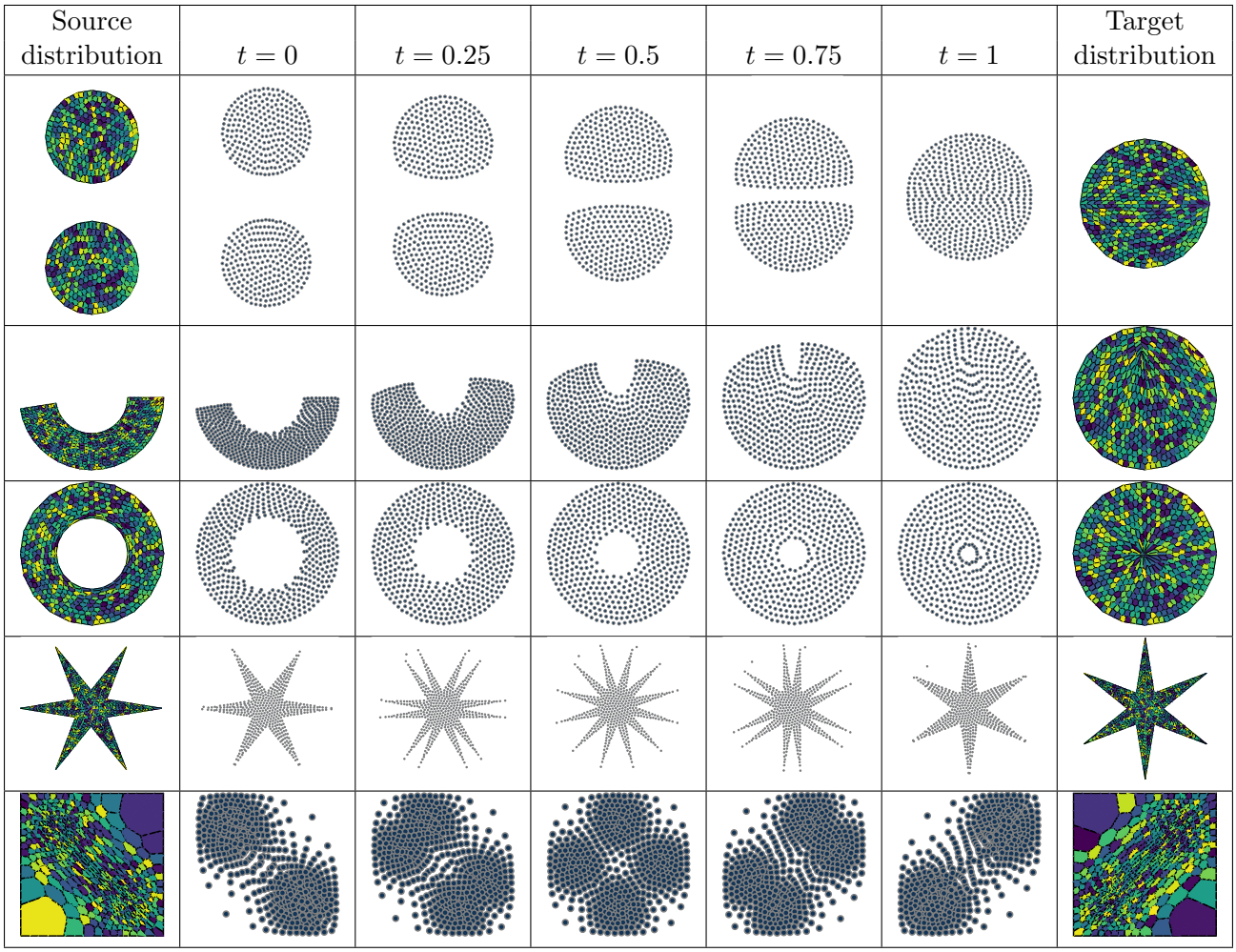


Figure 3.3: Linear interpolations between the sites positions obtained through the alternating algorithm, with $N = 500$ samples. The initial and final positions of the sites approximate well the input meshes, and the interpolated point clouds approximate the displacement interpolation. The first four rows present the results on uniform measures with discontinuities in the displacement interpolation, and the last one interpolate between two identical square domains with non-uniform densities.

Chapter 4

Optimization approaches

We have presented in Chapter 4 an alternating algorithm for solving the constrained optimization problem presented in Chapter 3. Algorithm 4 involves computing two optimal transport plans per iteration, which are usually computed using convex optimization techniques. This means that this method will make a number of calls to a convex optimizer that is linear in the number of iterations. There exist more classical methods for solving constrained optimization problems, which would solve the problem in only one optimizer call. This opens the possibility for using acceleration schemes, and getting better performances.

In this chapter, we explore first and second-order methods for carrying out this constrained optimization, and discuss their results.

4.1 Problem formulation

We recall the constrained optimization formulation given in Chapter 3:

$$\begin{aligned}
\max_{(x_i,\phi_i,y_i,\psi_i)} & \sum_{i=1}^{N} \int_{x\in Pow^{\phi}(x_i)} (\|x-x_i\|^2 - \phi_i) d\mu(x) + \sum_{i=1}^{N} p_i\phi_i \\
& + \sum_{i=1}^{N} \int_{y\in Pow^{\psi}(y_i)} (\|y-y_i\|^2 - \psi_i) d\nu(x) + \sum_{i=1}^{N} q_i\psi_i \\
\text{s.t. } & \int_{x\in Pow^{\phi}(x_i)} (x-y_i) d\mu(x) = 0 \\
\text{and } & \int_{y\in Pow^{\psi}(y_i)} (y-x_i) d\nu(y) = 0
\end{aligned}$$

The objective function is a combination of transport functionals that is maximal when power diagrams correspond to optimal transport maps. Constraints encode the fact that the sites (x_i) and (y_i) are located at the barycenters of the respective cells $Pow^{\phi}(x_i)$ and $Pow^{\psi}(y_i)$.

Notations To lighten our formulas, we will use $\mathcal{X}$ as a shorthand for the parameters vector $(x_i, \phi_i, y_i, \psi_i)_{1 \leq i \leq N}$ and introduce the following notations. The transport functionals that we wish to maximize are denoted:

$$\Phi_x(\mathcal{X}) = \sum_{i=1}^{N} \int_{x \in Pow^{\phi}(x_i)} (\|x - x_i\|^2 - \phi_i) d\mu(x) + \sum_{i=1}^{N} p_i \phi_i$$
$$\Phi_y(\mathcal{X}) = \sum_{i=1}^{N} \int_{y \in Pow^{\psi}(y_i)} (\|y - y_i\|^2 - \psi_i) d\nu(\mathrm{y}) + \sum_{i=1}^{N} q_i \psi_i$$

and we will call their sum $\Phi(\mathcal{X}) = \Phi_x(\mathcal{X}) + \Phi_y(\mathcal{X})$.

Centroidal constraints that we wish to cancel out are denoted:

$$\beta_{x,i}(\mathcal{X}) = \int_{x \in Pow^{\phi}(x_i)} (x - y_i) d\mu(x)$$
$$\beta_{y,i}(\mathcal{X}) = \int_{y \in Pow^{\psi}(y_i)} (y - x_i) d\nu(y)$$

$\beta_{x,i}$ is the gap between the (non-normalized) barycenter of the cell $Pow^{\phi}(x_i)$ and the site y_i (scaled by the area of the cell). It cancels out when $\int_{x \in Pow^{\phi}(x_i)} x d\mu(x) = y_i \cdot \int_{x \in Pow^{\phi}(x_i)} d\mu(x)$, i.e. when $y_i = \frac{\int_{x \in Pow^{\phi}(x_i)} x d\mu(x)}{\int_{x \in Pow^{\phi}(x_i)} d\mu(x)}$, which is the definition of the barycenter of $Pow^{\phi}(x_i)$ (given that the cell has non-null mass).

We will also denote $n_i(x)$ the outwards-pointing normal to $Pow^{\phi}(x_i)$ at point $x \in \partial Pow^{\phi}(x_i)$, K_i the number of edges of the boundary of $Pow^{\phi}(x_i)$, i_k the index of the k-th neighboring site to x_i and E_{i,i_k} the common edge to $Pow^{\phi}(x_i)$ and $Pow^{\phi}(x_{i_k})$. $\epsilon, \eta \in \{1, 2\}$ denote one of the two-dimensions coordinates (i.e., x_η is x_1 or x_2 where $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$). These notations are illustrated in Figure 4.1.

Barycenter formulations We have experimented with other barycenter expressions, such as $\beta_{x,i}(\mathcal{X}) = \frac{\int_{x \in Pow^{\phi}(x_i)} (x - y_i) d\mu(x)}{\int_{x \in Pow^{\phi}(x_i)} d\mu(x)}$ (computing the difference between the normalized barycenter and the site), or the formulation (inspired

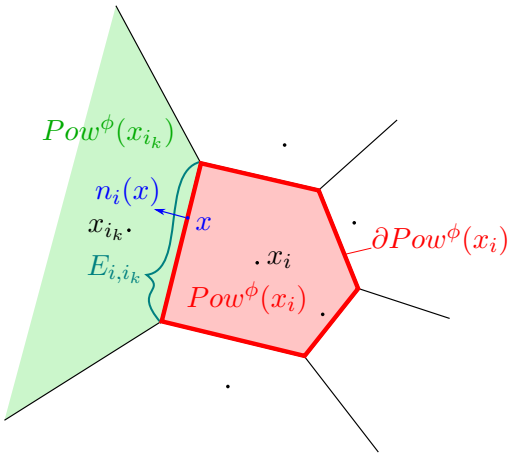


Figure 4.1: The power cell $Pow^{\phi}(x_i)$ is associated to the site x_i. Its boundary $\partial Pow^{\phi}(x_i)$, in the 2-d case, is a polygon with K_i edges (here $K_i = 5$). Its k-th neighbor, for $1 \leq k \leq K_i$, is $Pow^{\phi}(x_{i_k})$. Their common edge is E_{i,i_k}, and the outwards-pointing normal to $x \in \partial Pow^{\phi}(x_i)$ is $n_i(x)$.

by the Lloyd functional 1.7) $\beta_{x,i}(\mathcal{X}) = \int_{x \in Pow^{\phi}(x_i)} \|x - y_i\|^2 \, d\mu(x)$, to be minimized instead of canceled. However, they did not yield any significantly different results.

4.2 Optimization algorithms

We have first considered exploring gradient descent algorithms on an objective function combining both the functionals to optimize and the constraints to fulfill. However, this class of algorithms is bound to fail on this particular problem, because the term $\nabla_{x_i}(\Phi_x) = 2 \int_{x \in Pow^{\phi}(x_i)} (x - x_{\mathrm{i}}) d\mu(x)$ does not point in the direction of a solution. Detail of the methods we experimented with, and the reason for their failure, are detailed in Appendix A.

We thus experimented with second-order techniques with a theoretical basis for correctness.

4.2.1 Constrained Newton optimization

In this section, we present a more principled approach based on a second-order Newton optimization with Lagrange multipliers to enforce constraints. We use the method presented in Boyd and Vandenberghe [BV04] to establish a constrained optimization algorithm drawing upon Newton's method.

This method seemed theoretically promising, but it did give any results. We nonetheless present it for the sake of completeness.

Computation of the Newton step

The Lagrangian associated to the problem is:

$$\mathcal{L}(\mathcal{X}, \Lambda) = \Phi(\mathcal{X}) + \Lambda \cdot B(\mathcal{X})$$

where $\Phi(\mathcal{X}) = \Phi_x(\mathcal{X}) + \Phi_y(\mathcal{X})$ is the objective function of the unconstrained problem, $B(\mathcal{X}) = \begin{pmatrix} \beta_x(\mathcal{X}) \\ \beta_y(\mathcal{X}) \end{pmatrix}$ is the barycentric constraint vector, and Λ is the vector of Lagrange multipliers. We will denote by J the Jacobian of B, and by H the hessian of Φ.

The optimality condition for the Lagrangian is:

$$\begin{cases} \nabla\Phi(\mathcal{X}) + J(\mathcal{X})^\top \cdot \Lambda = 0 \\ B(\mathcal{X}) = 0 \end{cases} \tag{4.1}$$

The goal is, starting from a point $\mathcal{X}$, to find a direction $\Delta\mathcal{X}$ such that $\mathcal{X} + \Delta\mathcal{X}$ verifies condition 4.1. After substituting $\mathcal{X}$ with $\mathcal{X} + \Delta\mathcal{X}$, the gradient in the first equation and the constraint B in the second can be linearized as:

$$\nabla\Phi(\mathcal{X} + \Delta\mathcal{X}) = \nabla\Phi(\mathcal{X}) + H(\mathcal{X})\Delta\mathcal{X} + O(\|\Delta\mathcal{X}\|) \tag{4.2}$$

$$B(\mathcal{X} + \Delta\mathcal{X}) = B(\mathcal{X}) + J(\mathcal{X})\Delta\mathcal{X} + O(\|\Delta\mathcal{X}\|) \tag{4.3}$$

This allows us to rewrite the optimality condition as a linear system:

$$\begin{pmatrix} H(\mathcal{X}) & J(\mathcal{X})^\top \\ J(\mathcal{X}) & 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta\mathcal{X} \\ \Lambda \end{pmatrix} = \begin{pmatrix} -\nabla\Phi(\mathcal{X}) \\ -B(\mathcal{X}) \end{pmatrix} \tag{4.4}$$

Solving system 4.4 for $\begin{pmatrix} \Delta\mathcal{X} \\ \Lambda \end{pmatrix}$ gives us our Newton step $\Delta\mathcal{X}$ and the new value of the Lagrange multiplier Λ.

If we dismiss the Lagrange multiplier Λ and the second column block of the matrix, this method amounts to finding a zero of the expression $\begin{pmatrix} \nabla\Phi(\mathcal{X}) \\ B(\mathcal{X}) \end{pmatrix}$ using Newton's method, through linearization of $\nabla\Phi$ and B as previously done. The cancellation of this vector implies the fulfillment of the constraints stated in 2.2: the transport functional's gradient is null when the transport plan is optimal, and β_x and β_y are zero when the sites are located in the barycenters.

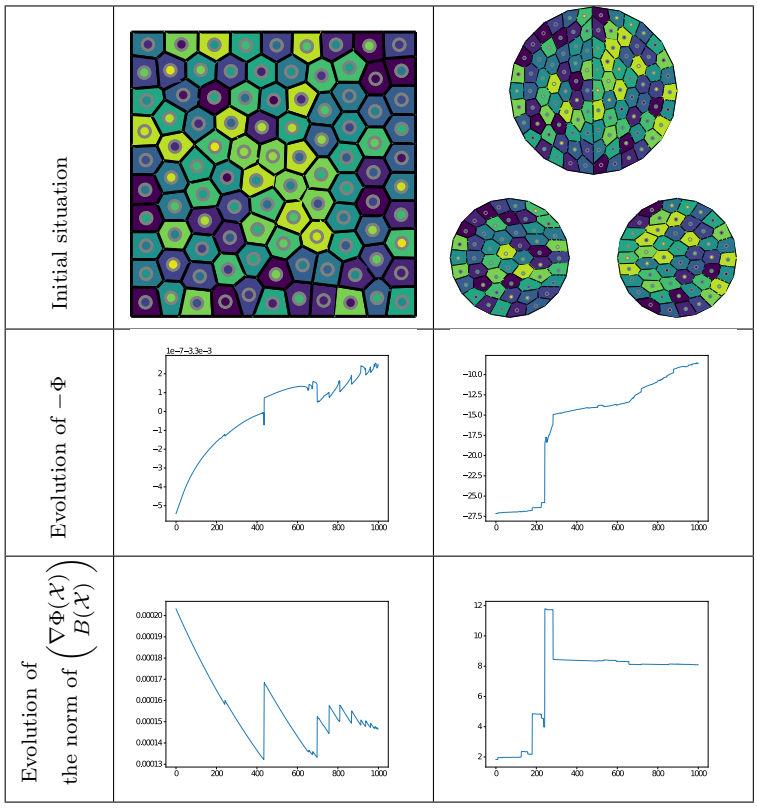


Figure 4.2: Evolution of Φ and of the norm of $\begin{pmatrix} \nabla\Phi(\mathcal{X}) \\ B(\mathcal{X}) \end{pmatrix}$ during the Newton optimization, for two different settings, both using the result of the converged algorithm as a starting point. Both ran for 1000 iterations.

Algorithm

The Newton method stemming from Equation 4.4 to compute coupled semi-discrete transport plans is given in Algorithm 6.

This algorithm initializes the weights so that the transport plans start out as Voronoi diagrams. However, one can also carry out a number of steps of the alternating algorithm (Algorithm 5) before starting the main loop. This is conjectured to allow the optimization to start at a point closer to the constrained optimum and thus to avoid divergence.

The version of the Newton algorithm presented in Algorithm 6 uses a fixed stepsize α. Another possibility would be to recompute the stepsize at each

Algorithm 6 Newton algorithm for solving the constrained problem 2.1

Input: Two measures μ and ν, a number of samples N, a stepsize α

Sample ν with sites $(x_i)_{1\leq i\leq N}$

Sample μ with sites $(y_i)_{1\leq i\leq N}$

Initialize transport weights $(\phi_i)_{1\leq i\leq N}$ and $(\psi_i)_{1\leq i\leq N}$ with a identical value for all weights (0 in practice)

while not converged **do**

Compute the step $\begin{pmatrix}\Delta\mathcal{X}\\ \Lambda\end{pmatrix} = \begin{pmatrix}H(\mathcal{X}) & J(\mathcal{X})^\top\\ J(\mathcal{X}) & 0\end{pmatrix}^{-1} \cdot \begin{pmatrix}-\nabla\Phi(\mathcal{X})\\ -B(\mathcal{X})\end{pmatrix}$

Update the site positions and the transport weights

$$\begin{pmatrix}x_i\\ \phi_i\\ y_i\\ \psi_i\end{pmatrix} = \Delta\mathcal{X} + \alpha \cdot \begin{pmatrix}x_i\\ \phi_i\\ y_i\\ \psi_i\end{pmatrix}$$

end while

step, for example by using the same method as KMT 1, by searching for the maximum stepsize that does not create empty cells and diminishes the norm of the right hand of the system $\left\|\begin{pmatrix}-\nabla\Phi(\mathcal{X})\\ -B(\mathcal{X})\end{pmatrix}\right\|$. In order to use this method, the initial weights (ϕ_i) and (ψ_i) have to describe transport plans that present no empty cells. Starting the algorithm with a few alternating steps can achieve this goal.

Experiments

Figure 4.2 presents the results of two example runs of this algorithm. As we stated in the beginning of Section 4.2.1, this algorithm did not give conclusive results. We thus chose to showcase its execution on the simplest possible case. The starting point of the algorithm is computed by doing 50 iterations of the alternating algorithm. This should be close to a stationary point of $\mathcal{L}$.

We plot the value of $-\Phi$, which we aim to minimize[1], and the norm of $\begin{pmatrix}\nabla\Phi(\mathcal{X})\\ B(\mathcal{X})\end{pmatrix}$, which we aim to cancel and thus also minimize. The results

[1]This is due to the implementation of optimal transport in Graphite [Lé19], where, instead of maximizing the objective functions Φ_x and Φ_y, the optimizer minimizes their opposite.

in the left column were computed on two identical square domains, while those in the right column correspond to transporting one disk towards two disks. We chose to start the optimization at convergence of the alternating algorithm, in order to test it on a minimally difficult setting. These starting points are given on the first line.

We would expect to see both the objective function and the norm of the right hand side of Equation 4.4 eventually decrease, especially since these optimizations are starting from a — theoretically — already converged setting. In the case where the domains are identical (i.e. where our problem boils down to computing two CCCPDs), the norm of the constraint vector and the objective function stay mostly constant while locally increasing. This is consistent with what we would expect to see. However, in the other case, where the domains are disjunct, both value significantly increase, with a marked "jump" at one point: the algorithm does not converge.

We explored whether the stepsize choice of Kitagawa et al. [KMT16] could be adapted to our case to help our algorithm converge. However, when we tested it on non-trivial situations, the algorithm systematically reached a point where it could not find a suitable stepsize. This suggests that the step $\Delta\mathcal{X}$ given by our algorithm might not always be a descent direction.

4.2.2 The issue with descent algorithms

It seems that the issue encountered by our optimization methods stems from a very noisy function: the algorithms tend to take directions that do not globally decrease the objective function.

As illustrated in Figure 4.3, a reasonably small step, applied to a site's position, dramatically modifies the geometry of its associated cell. Thus, a step that intended on bringing us closer to a solution might modify the setting enough for it to actually be further away from a solution than it originally was.

To give a more precise quantification of this phenomenon, we need to evaluate how a cell evolves when its parameters are slightly perturbated. We recall the normal component of the frontier's derivatives with respect to its site, which we compute in 4.3.1.

$$\left\langle \frac{\partial x}{\partial x_{i,\eta}}, n_i(x) \right\rangle = \frac{x_\eta - x_{i,\eta}}{\|x_{i_k} - x_i\|}$$

where $\eta \in \{0, 1\}$ is one of the 2-d coordinates.

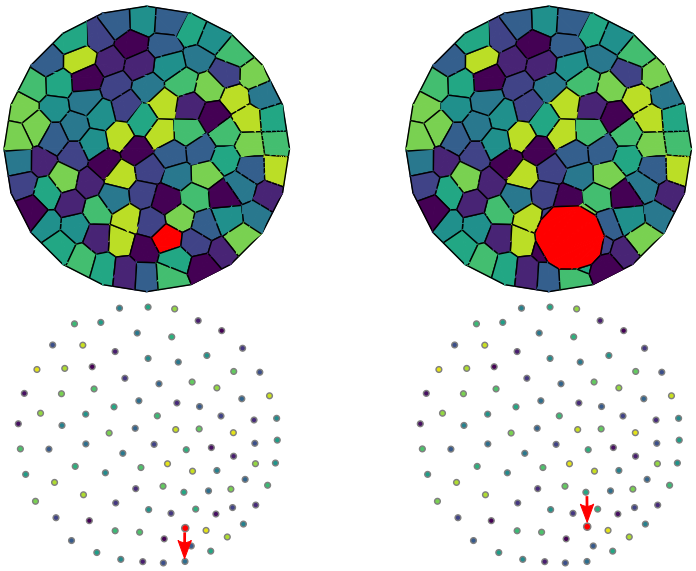


Figure 4.3: Evolution of a cell when its site is perturbated. As the red site is slightly moved up while keeping the power weights identical, its associated cell swells.

We can see that this term is proportional to the distance between the site and its cell: cells whose sites are far away are more sensitive to small site displacements. This makes iterative methods quite impractical in this setting.

4.2.3 Alternating algorithm as an energy optimization

It seems that the best way to find a constrained optimal point in our case is to alternate between an optimization step and a constraint enforcement step. This describes the alternating algorithm we presented in 3.1.1.

We can see in Figure 4.4 that the norm of the constraints vector $\left\| \begin{pmatrix} \nabla\Phi \\ B \end{pmatrix} \right\|$ decreases to a minimum value, and that the transport objective function Φ increases globally to a maximum value during the iterations of the algorithms. The non-zero minimal value of the first expression corresponds to the term $\nabla_{x_i,y_i}\Phi$, which is not null. These values locally increase during the recentering step of the main loop, but this increase is usually compensated during the next transport step.

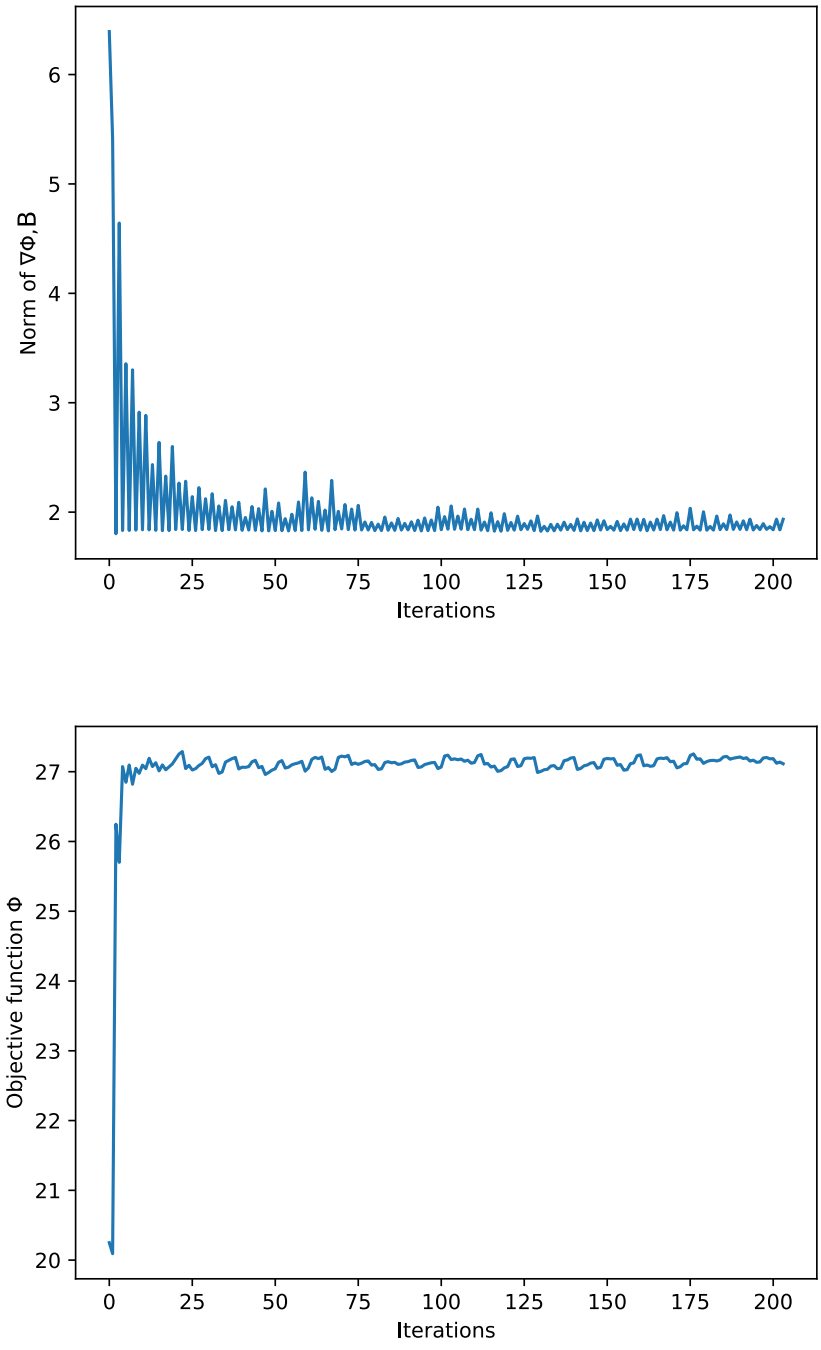


Figure 4.4: Evolution of $\left\| \begin{pmatrix} \nabla\Phi \\ B \end{pmatrix} \right\|$ and Φ during the alternating algorithm.

4.3 First order derivatives

We compute the analytical expressions of the derivatives of Φ_x, Φ_y, $\beta_{x,i}$ and $\beta_{y,i}$. We made the choice of computing these expressions explicitly, instead of using autodifferentiation (as is done in applications such as Light in Power [MMT17b]). This allowed for a direct implementation of these expressions in Graphite [Lé19] without having to integrate an autodifferentiation library into the software. In addition, this leaves open the possibility of proving further theoretical properties of the algorithms using these expressions in future work.

As stated in Section 4.1, indices η and $\epsilon \in \{1, 2\}$ denote one of the 2-d coordinates, with ϵ typically being used for the coordinate of the differentiated quantity and η when differentiating with respect to the η-th coordinate of a site.

4.3.1 Reynolds theorem

Both our objective function and our constraints involve integrals over power cells, that depend on both the positions of the sites and the values of the weights. Thus, we will need to know how to differentiate an integral over a domain that depends on the differentiation parameter.

Fortunately, this is exactly the problem solved by the Reynolds transport theorem [Rey03]. It reads as follows:

Theorem 2 (Reynolds transport theorem)**.** *Given an integration domain $\Omega(t)$ depending on the real-valued parameter t, and a function $f(x,t)$, the derivative of $\int_{x\in\Omega(t)} f(x,t)dx$ with respect to t is:*

$$\frac{\partial}{\partial t}\int_{x\in\Omega(t)} f(x,t)dx = \int_{x\in\Omega(t)} \frac{\partial f}{\partial t}(x,t)dx \tag{4.5}$$

$$+ \int_{\theta\in\mathbb{S}^1} \left\langle \frac{\partial \gamma}{\partial t}(\theta,t), n(\gamma(\theta,t)) \right\rangle f(\gamma(\theta,t),t) \left\| \frac{\partial \gamma}{\partial \theta}(\theta,t) \right\| d\theta \tag{4.6}$$

$$\tag{4.7}$$

where $\partial\Omega(t)$ is the boundary of $\Omega(t)$, $\gamma(.,t) : \mathbb{S}^1 \to \partial\Omega(t)$ is a parameterization of $\partial\Omega(t)$ defined on the unit circle $\mathbb{S}^1$ *and $n(x)$ is the outwards-pointing normal at $x \in \partial\Omega(t)$.*

As stated by Cortes et al. [CMB04], the parameterization $\gamma(.,t)$ of the domain's boundary needs to be continuous and piecewise differentiable for

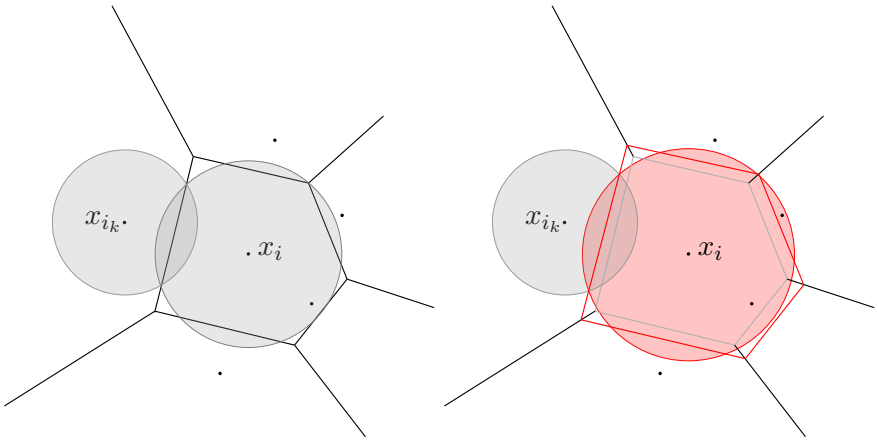


Figure 4.5: The weights defining a power diagram can be visually represented as circles with radii $r = \sqrt{\phi_i}$. When modifying the weight associated to a cell, its boundaries move parallel to themselves.

all t, and $\gamma(\theta, .)$ needs to be $\mathcal{C}^1$ for all θ as well. The integrated function f also needs to be continuous on $\Omega(t)$ and integrable over $\partial\Omega(t)$, and its derivative $\frac{\partial f}{\partial t}(x,t)dx$ needs to be integrable over $\Omega(t)$ so that the integrals in the expression 4.5.

Intuitively, the variation of the integral of a function over a variable domain comes from both the variation of the function inside the domain and the variation of the frontier's position.

Application to power diagram cells

In practice, we will use the Reynolds transport theorem with $\Omega = Pow^\phi(x_i)$ and differentiate the integral with respect to any site $x_{j,\eta}$ or weight ϕ_j ($1 \leq j \leq N^2$ and $\eta \in \{0, 1\}$), these variables playing the role of t.

Figures 4.5 and 4.6 display the evolution of a power cell when its parameters (respectively its weight and its defining site) undergo a small modification. These modifications impact both the total mass of a cell and the outline of its boundary, which in turn transmit to the expressions of the derivatives.

We need to define a continuous and differentiable parameterization of $\partial Pow^\phi(x_i)$. Since $\partial Pow^\phi(x_i)$ is a polygon, we define a different parameterization over $[0, 1]$ for each of its edges, that can then be assembled through a

[2] As we shall see, these expressions are null unless j is equal to i or one of its neighbors.

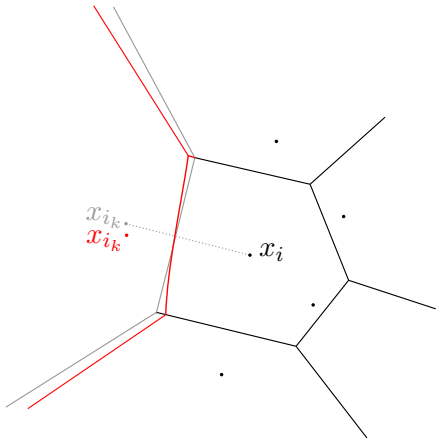


Figure 4.6: When the site defining a power cell changes position, the boundaries of the cell rotate around their intersection point with segment $[x_i; x_{i_k}]$

change of variables into a parameterization over $\mathbb{S}^1$.

A parameterization of edge E_{i,i_k} of $\partial Pow^{\phi}(x_i)$, bounded by vertices s_k and s_{k+1}, can be $\gamma_k(\theta) = (1-\theta)\cdot s_k + \theta \cdot s_{k+1}$. Partial derivatives of γ_k are, in the general case:

$$\frac{\partial \gamma_k}{\partial \theta}(\theta) = s_{k+1} - s_k \quad (4.8)$$

$$\frac{\partial \gamma_k}{\partial t}(\theta) = (1-\theta)\cdot \frac{\partial s_k}{\partial t} + \theta \cdot \frac{\partial s_{k+1}}{\partial t} \quad (4.9)$$

where t is either ϕ_j or $x_{j,\eta}$.

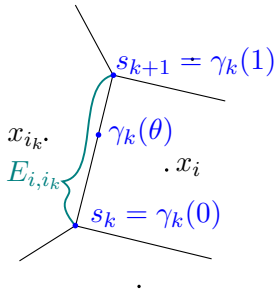


Figure 4.7: Parameterization γ_k of edge E_{i,i_k}.

Note that we do not need $\frac{\partial \gamma_k}{\partial t}(\theta)$ on its own, but only $\left\langle \frac{\partial \gamma_k}{\partial t}(\theta), n_i(\gamma_k(\theta)) \right\rangle = \left\langle \frac{\partial \gamma_k}{\partial t}(\theta), \frac{x_{i_k} - x_i}{\|x_{i_k} - x_i\|} \right\rangle$.

s_k can be characterized as the intersection between edges E_{i,i_k} and $E_{i,i_{k+1}}$, using the definition of a power cell 3:

$$\begin{cases} \|s_k - x_i\|^2 - \phi_i &= \|s_k - x_{i_k}\|^2 - \phi_{i_k} \\ \|s_k - x_i\|^2 - \phi_i &= \left\|s_k - x_{i_{k+1}}\right\|^2 - \phi_{i_{k+1}} \end{cases} \quad (4.10)$$

However, since we are only looking for the component of $\frac{\partial s_k}{\partial \phi_j}$ and $\frac{\partial s_k}{\partial x_{j,\eta}}$ along direction n_i, only the first equation is needed:

$$\|s_k - x_i\|^2 - \phi_i = \|s_k - x_{i_k}\|^2 - \phi_{i_k}$$

- If $t = \phi_j$, both sides of the first equation can be differentiated to obtain:

  $$2 \cdot \left\langle s_k - x_i, \frac{\partial s_k}{\partial \phi_j} - \frac{\partial x_i}{\partial \phi_j} \right\rangle - \frac{\partial \phi_i}{\partial \phi_j} = 2 \cdot \left\langle s_k - x_{i_k}, \frac{\partial s_k}{\partial \phi_j} - \frac{\partial x_{i_k}}{\partial \phi_j} \right\rangle - \frac{\partial \phi_{i_k}}{\partial \phi_j}$$

  Since x_i and x_{i_k} do not depend on any weight ϕ_j, this equation simplifies as:

  $$\left\langle x_{i_k} - x_i, \frac{\partial s_k}{\partial \phi_j} \right\rangle = \frac{1}{2} \cdot \left(\frac{\partial \phi_i}{\partial \phi_j} - \frac{\partial \phi_{i_k}}{\partial \phi_j} \right)$$

  The expression $\left\langle x_{i_k} - x_i, \frac{\partial s_k}{\partial \phi_j} \right\rangle$ is thus only non-null when $j = i$ or $j = i_k$:

  $$\left\langle \frac{\partial s_k}{\partial \phi_j}, \frac{x_{i_k} - x_i}{\|x_{i_k} - x_i\|} \right\rangle = \begin{cases} \frac{1}{2 \cdot \|x_{i_k} - x_i\|} & j = i \\ \frac{-1}{2 \cdot \|x_{i_k} - x_i\|} & j = i_k \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

- If $t = x_{j,\eta}$, differentiating the equation yields:

  $$2 \cdot \left\langle s_k - x_i, \frac{\partial s_k}{\partial x_{j,\eta}} - \frac{\partial x_i}{\partial x_{j,\eta}} \right\rangle - \frac{\partial \phi_i}{\partial x_{j,\eta}} = 2 \cdot \left\langle s_k - x_{i_k}, \frac{\partial s_k}{\partial x_{j,\eta}} - \frac{\partial x_{i_k}}{\partial x_{j,\eta}} \right\rangle - \frac{\partial \phi_{i_k}}{\partial x_{j,\eta}}$$

  Since ϕ_i and ϕ_{i_k} do not depend on any site position x_j, this equation simplifies as

  $$\left\langle s_k - x_i, \frac{\partial s_k}{\partial x_{j,\eta}} - \frac{\partial x_i}{\partial x_{j,\eta}} \right\rangle = \left\langle s_k - x_{i_k}, \frac{\partial s_k}{\partial x_{j,\eta}} - \frac{\partial x_{i_k}}{\partial x_{j,\eta}} \right\rangle$$

  $$\left\langle x_{i_k} - x_i, \frac{\partial s_k}{\partial x_{j,\eta}} \right\rangle = \left\langle s_k - x_i, \frac{\partial x_i}{\partial x_{j,\eta}} \right\rangle - \left\langle s_k - x_{i_k}, \frac{\partial x_{i_k}}{\partial x_{j,\eta}} \right\rangle$$

  which is only non-null when $j = i$ or $j = i_k$:

  $$\left\langle \frac{\partial s_k}{\partial x_{j,\eta}}, \frac{x_{i_k} - x_i}{\|x_{i_k} - x_i\|} \right\rangle = \begin{cases} \frac{(s_k - x_i)_\eta}{\|x_{i_k} - x_i\|} & j = i \\ \frac{(s_k - x_{i_k})_\eta}{\|x_{i_k} - x_i\|} & j = i_k \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

We can now reinject these expressions into Equation 4.9:

$$\left\langle \frac{\partial \gamma_k}{\partial \phi_j}(\theta), n_i(\gamma_k(\theta)) \right\rangle = \begin{cases} \frac{1}{2 \cdot \left\| x_{i_k} - x_i \right\|} & j = i \\ \frac{-1}{2 \cdot \left\| x_{i_k} - x_i \right\|} & j = i_k \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

$$\left\langle \frac{\partial \gamma_k}{\partial x_{j,\eta}}(\theta), n_i(\gamma_k(\theta)) \right\rangle = \begin{cases} \frac{(\gamma_k(\theta) - x_i)_\eta}{\left\| x_{i_k} - x_i \right\|} & j = i \\ \frac{(\gamma_k(\theta) - x_{i_k})_\eta}{\left\| x_{i_k} - x_i \right\|} & j = i_k \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

We will use $\frac{\partial x}{\partial t}$ as a shorthand to denote $\frac{\partial \gamma_k}{\partial t}(\theta)$ when $x = \gamma_k(\theta)$.

We can now give more explicit general expressions for the derivatives of integrals over $Pow^\phi(x_i)$ we will encounter. We give an expression independent of parameterization and write the last part as an integral over the boundary:

$$\begin{aligned}
\frac{\partial}{\partial \phi_i} \int_{x \in Pow^\phi(x_i)} f(x, \phi_i) dx &= \int_{x \in Pow^\phi(x_i)} \frac{\partial f}{\partial \phi_i}(x, \phi_i) dx \\
&\quad + \sum_{i=1}^{K_i} \int_{x \in E_{i,i_k}} \frac{1}{2 \cdot \left\| x_{i_k} - x_i \right\|} f(x, \phi_i) dx \\
\frac{\partial}{\partial \phi_{i_k}} \int_{x \in Pow^\phi(x_i)} f(x, \phi_{i_k}) dx &= \int_{x \in Pow^\phi(x_i)} \frac{\partial f}{\partial \phi_{i_k}}(x, \phi_{i_k}) dx \\
&\quad - \int_{x \in E_{i,i_k}} \frac{1}{2 \cdot \left\| x_{i_k} - x_i \right\|} f(x, \phi_{i_k}) dx \\
\frac{\partial}{\partial x_{i,\eta}} \int_{x \in Pow^\phi(x_i)} f(x, x_{i,\eta}) dx &= \int_{x \in Pow^\phi(x_i)} \frac{\partial f}{\partial x_{i,\eta}}(x, x_{i,\eta}) dx \\
&\quad + \sum_{i=1}^{K_i} \int_{x \in E_{i,i_k}} \frac{(x - x_i)_\eta}{\left\| x_{i_k} - x_i \right\|} f(x, x_{i,\eta}) dx \\
\frac{\partial}{\partial x_{i_k,\eta}} \int_{x \in Pow^\phi(x_i)} f(x, x_{i,\eta}) dx &= \int_{x \in Pow^\phi(x_i)} \frac{\partial f}{\partial x_{i_k,\eta}}(x, x_{i_k,\eta}) dx \\
&\quad + \int_{x \in E_{i,i_k}} \frac{(x - x_{i_k})_\eta}{\left\| x_{i_k} - x_i \right\|} f(x, x_{i_k,\eta}) dx
\end{aligned} \tag{4.15}$$

4.3.2 Objective function

We present here the first-order derivatives of the unconstrained objective function $(\Phi_x + \Phi_y)$ with regards to the sites x_i and y_i, and the weights ϕ_i and ψ_i.

$$\nabla(\Phi_x + \Phi_y)(\mathcal{X}) = \begin{pmatrix} 2\int_{x \in Pow^{\phi}(x_i)} (x_i - x) d\mu(x) \\ p_i - \mu(Pow^{\phi}(x_i)) \\ 2\int_{y \in Pow^{\psi}(y_i)} (y_i - y) d\mu(y) \\ q_i - \mu(Pow^{\psi}(y_i)) \end{pmatrix} \tag{4.16}$$

We refer the reader to the appendix of [DGBOD12] for the detail of the computations.

The gradient with respect to the weights is the difference between the actual area of a cell and its prescribed area. In a gradient descent, this term would diminish the weight of cells with too much mass and increase that of cells with too little. The gradient with respect to the sites is aligned with the difference between the position of a site and the position of the associated barycenter. In a gradient ascent, this term would tend to bring a site closer to the barycenter of its own cell, instead of the barycenter of the corresponding cell in the other domain, while a gradient descent would send the sites away from the meshes to infinity. As pointed out in Section A, neither of these behaviors are desirable in our application.

4.3.3 Constraints

We do not carry out the computation of $\frac{\partial \beta_{x,i}(\mathcal{X})_{\varepsilon}}{\partial \psi_j}$ as it is trivially 0, since no ψ_j intervenes in the expression of $\beta_{x,i}$.

Derivative of the centering constraint $\beta_{x,i}$ with respect to the position of the associated site x_i

$$\begin{aligned} \frac{\partial \beta_{x,i}(\mathcal{X})_{\varepsilon}}{\partial x_{i,\eta}} &= \int_{x \in \partial Pow^{\phi}(x_i)} \left\langle \frac{\partial x}{\partial x_{i,\eta}}, n_i(x) \right\rangle (x - y_i)_{\varepsilon} d\mu(x) \\ &= \sum_{k=1}^{K_i} \int_{x \in E_{i,i_k}} \frac{x_{\eta} - x_{i,\eta}}{\|x_i - x_{i_k}\|} (x - y_i)_{\varepsilon} \\ &= \sum_{k=1}^{K_i} \frac{1}{\|x_i - x_{i_k}\|} \int_{x \in E_{i,i_k}} (x - x_i)_{\eta} (x - y_i)_{\varepsilon} d\mu(x) \end{aligned} \tag{4.17}$$

Computing the derivative of an integral over a cell demands the use of Reynolds' theorem. Here, the integrand $x - y_i$ is constant with respect to

x_i, thus only the term integrating over $\partial Pow^\phi(x_i)$ remains. One then has to replace $\left\langle \frac{\partial x}{\partial x_{i,\eta}}, n_i(x) \right\rangle$ with its already computed value (Equation 4.14) and to split $\partial Pow^\phi(x_i)$ into its edges E_{i,i_k}.

Derivative of the centering constraint $\beta_{x,i}$ with respect to the position of a neighboring site x_{i_k}

$$\begin{aligned}
\frac{\partial \beta_{x,i}(\mathcal{X})_\varepsilon}{\partial x_{i_k,\eta}} &= \int_{x \in \partial Pow^\phi(x_i)} \left\langle \frac{\partial x}{\partial x_{i_k,\eta}}, n_i(x) \right\rangle (x - y_i)_\varepsilon d\mu(x) \\
&= \int_{x \in E_{i,i_k}} -\left\langle \frac{\partial x}{\partial x_{i_k,\eta}}, n_{i_k}(x) \right\rangle (x - y_i)_\varepsilon d\mu(x) \\
&= \int_{x \in E_{i,i_k}} -\frac{x_\eta - x_{i_k,\eta}}{\|x_{i_k} - x_i\|} (x - y_i)_\varepsilon d\mu(x) \\
&= -\frac{1}{\|x_{i_k} - x_i\|} \int_{x \in E_{i,i_k}} (x - x_{i_k})_\eta (x - y_i)_\varepsilon d\mu(x)
\end{aligned} \tag{4.18}$$

This computation is fairly similar to the previous one in 4.3.3, except for the fact that $\frac{\partial x}{\partial x_{i_k,\eta}}$ is null when x is not part of $\partial Pow^\phi(x_{i_k})$. Here, this means that only the common edge E_{i,i_k} to $Pow^\phi(x_i)$ and $Pow^\phi(x_{i_k})$ remains in the integral. When $x \in E_{i,i_k}$, $n_i(x) = -n_{i_k}(x)$, hence the minus sign appearing at line 2.

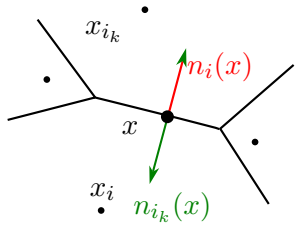


Figure 4.8: Outwards-pointing normals $n_i(x)$ and $n_{i_k}(x)$ at point x from neighboring cells $Pow^\phi(x_i)$ and $Pow^\phi(x_{i_k})$ have opposite directions.

Derivative of the centering constraint $\beta_{x,i}$ with respect to the weight of the associated site ϕ_i

$$\begin{aligned}
\frac{\partial \beta_{x,i}(\mathcal{X})_\varepsilon}{\partial \phi_i} &= \int_{x \in \partial Pow^\phi(x_i)} \left\langle \frac{\partial x}{\partial \phi_i}, n_i(x) \right\rangle (x - y_i)_\varepsilon d\mu(x) \\
&= \sum_{k=1}^{K_i} \frac{1}{2 \cdot \|x_i - x_{i_k}\|} \int_{x \in E_{i,i_k}} (x - y_i)_\varepsilon d\mu(x)
\end{aligned} \tag{4.19}$$

As in 4.3.3, we apply Reynolds' theorem, here differentiating with respect to the weight ϕ_i instead of to the site x_i. $x - y_i$ is also constant with respect to ϕ_i, which yields a similar expression except for the derivative of x.

Derivative of the centering constraint $\beta_{x,i}$ with respect to the weight of a neighboring site ϕ_{i_k}

$$\begin{aligned}\frac{\partial \beta_{x,i}(\mathcal{X})_\varepsilon}{\partial \phi_{i_k}} &= \int_{x\in\partial Pow^\phi(x_i)} \left\langle \frac{\partial x}{\partial \phi_{i_k}}, n_i(x) \right\rangle (x-y_i)_\varepsilon d\mu(x) \\ &= \int_{x\in E_{i,i_k}} -\left\langle \frac{\partial x}{\partial \phi_{i_k}}, n_{i_k}(x) \right\rangle (x-y_i)_\varepsilon d\mu(x) \\ &= -\frac{1}{2\cdot\|x_i - x_{i_k}\|} \int_{x\in E_{i,i_k}} (x-y_i)_\varepsilon d\mu(x)\end{aligned} \tag{4.20}$$

This expression is the same as in 4.3.3 but with only the term related to the k-th edge of $\partial Pow^\phi(x_i)$ remaining.

Derivative of the centering constraint $\beta_{x,i}$ with respect to the position **of the associated site in the opposite domain y_i**

$$\begin{aligned}\frac{\partial \beta_{x,i}(\mathcal{X})_\varepsilon}{\partial y_{i,\eta}} &= \int_{x\in Pow^\phi(x_i)} \frac{\partial (x-y_i)_\varepsilon}{\partial y_{i,\eta}} d\mu(x) \\ &= \begin{cases} -\mu(Pow^\phi(x_i)) \text{ if } \varepsilon = \eta \\ 0 \text{ otherwise} \end{cases}\end{aligned} \tag{4.21}$$

This computation also uses Reynolds' theorem, but, contrary to the previous ones, only keeps the integral over $Pow^\phi(x_i)$, since the geometry of $Pow^\phi(x_i)$ does not depend on y_i, and thus neither does $x \in \partial Pow^\phi(x_i)$. Passing to the next line is done by noticing that $\frac{\partial (x-y_i)_\varepsilon}{\partial y_{i,\eta}} = -\delta_{\varepsilon,\eta}$

Computing $\frac{\partial \beta_{x,i}(\mathcal{X})_\varepsilon}{\partial y_{i_k,\eta}}$ similarly yields 0, for y_i does not depend on y_{i_k}.

4.4 Hessian of the objective function

We compute here the second-order derivatives of the unconstrained objective function $(\Phi_x + \Phi_y)$ that make up the Hessian H. We compute derivatives with regards to the sites positions and the weights as well as crossed terms.

These results coincide with those appearing in de Gournay et al. [dGKL18], where they are computed using another method.

4.4.1 Derivatives with respect to the site positions

Second derivative of Φ **with respect to the same site** x_i **twice**

$$\begin{aligned}\frac{\partial^2(\Phi_x+\Phi_y)(\mathcal{X})}{\partial x_{i,\eta}^2} &= \frac{\partial 2 \int_{x\in Pow^\phi(x_i)} (x_i-x)_\eta d\mu(x)}{\partial x_{i,\eta}} \\ &= 2\cdot\int_{x\in Pow^\phi(x_i)} \frac{\partial (x_i-x)_\eta}{\partial x_{i,\eta}} d\mu(x) \\ &+ 2\cdot\int_{x\in \partial Pow^\phi(x_i)} \left\langle \frac{\partial x}{\partial x_{i,\eta}}, n_i(x)\right\rangle (x_i-x)_\eta d\mu(x) \\ &= 2\cdot\mu(Pow^\phi(x_i)) - 2\cdot\sum_{k=1}^{K_i}\int_{x\in E_{i,i_k}} \frac{(x_{i,\eta}-x_\eta)^2}{\|x_{i_k}-x_i\|} d\mu(x)\end{aligned} \tag{4.22}$$

The first line of the computation is obtained by substituting the value of $\frac{\partial(\Phi_x+\Phi_y)(\mathcal{X})}{\partial x_{i,\eta}}$ already stated in Section 4.3.2. The second and third lines stem from the application of Reynolds' theorem, with both terms remaining this time. The last line is obtained by substituting $\frac{\partial(x_i-x)_\eta}{\partial x_{i,\eta}} = 1$ and $\left\langle \frac{\partial x}{\partial x_{i,\eta}}, n_i(x)\right\rangle = \frac{x_\eta - x_{i,\eta}}{\|x_{i_k}-x_i\|}$ (Equation 4.14).

Second **derivative of** Φ **with respect to a cell's site** x_i **and a neighboring site** x_{i_k}

$$\begin{aligned}\frac{\partial^2(\Phi_x+\Phi_y)(\mathcal{X})}{\partial x_{i,\eta}\,\partial x_{i_k,\eta'}} &= \frac{\partial 2 \int_{x\in Pow^\phi(x_i)} (x_i-x)_\eta d\mu(x)}{\partial x_{i_k,\eta'}} \\ &= 2\cdot\int_{x\in Pow^\phi(x_i)} \frac{\partial (x_i-x)_\eta}{\partial x_{i_k,\eta'}} d\mu(x) \\ &+ 2\cdot\int_{x\in \partial Pow^\phi(x_i)} \left\langle \frac{\partial x}{\partial x_{i_k,\eta'}}, n_i(x)\right\rangle (x_i-x)_\eta d\mu(x) \\ &= 2\cdot\int_{x\in E_{i,i_k}} -\left\langle \frac{\partial x}{\partial x_{i_k,\eta'}}, n_{i_k}(x)\right\rangle (x_i-x)_\eta d\mu(x) \\ &= 2\cdot\int_{x\in E_{i,i_k}} \frac{(x_{i_k}-x)_{\eta'}\cdot(x_i-x)_\eta}{\|x_{i_k}-x_i\|} d\mu(x)\end{aligned} \tag{4.23}$$

This computation follows the same principles as the one in Section 4.4.1. The main differences stem from the fact that $\frac{\partial (x_i - x)_\eta}{\partial x_{i_k,\eta'}} = 0$ and that only the term related to the k-th edge of the boundary of $Pow^\phi(x_i)$ remains in the sum.

Second **derivative of Φ with respect to two different coordinates of the same site** x_i

$$\begin{aligned}
\frac{\partial^2(\Phi_x + \Phi_y)(\mathcal{X})}{\partial x_{i,1}\, \partial x_{i,2}} &= \frac{\partial 2 \int_{x \in Pow^\phi(x_i)} (x_i - x)_1 d\mu(x)}{\partial x_{i,2}} \\
&= 2 \cdot \int_{x \in Pow^\phi(x_i)} \frac{\partial (x_i - x)_1}{\partial x_{i,2}} d\mu(x) \\
&+ 2 \cdot \int_{x \in \partial Pow^\phi(x_i)} \left\langle \frac{\partial x}{\partial x_{i,2}}, n_i(x) \right\rangle (x_i - x)_1 d\mu(x) \\
&= 2 \cdot \sum_{k=1}^{K_i} \int_{x \in E_{i,i_k}} \frac{(x_i - x)_1 \cdot (x_i - x)_2}{\|x_{i_k} - x_i\|} d\mu(x)
\end{aligned} \tag{4.24}$$

This computation is very similar as well to 4.4.1, with $\frac{\partial (x_i - x)_1}{\partial x_{i,2}} = 0$ canceling the first term from Reynolds' theorem.

Second **derivative** of Φ **with respect to a site** x_i **and its corresponding site** y_i **in the opposite domain**

$$\begin{aligned}
\frac{\partial^2(\Phi_x + \Phi_y)(\mathcal{X})}{\partial x_{i,\eta}\, \partial y_{i,\eta'}} &= \frac{\partial 2 \int_{x \in Pow^\phi(x_i)} (x_i - x)_\eta d\mu(x)}{\partial y_{i,\eta'}} \\
&= 2 \cdot \int_{x \in Pow^\phi(x_i)} \frac{\partial (x_i - x)_\eta}{\partial y_{i,\eta'}} d\mu(x) \\
&+ 2 \cdot \int_{x \in \partial Pow^\phi(x_i)} \left\langle \frac{\partial x}{\partial y_{i,\eta'}}, n_i(x) \right\rangle (x_i - x)_\eta d\mu(x) \\
&= 0
\end{aligned} \tag{4.25}$$

The first line of the computation is obtained by substituting the value of $\frac{\partial (\Phi_x + \Phi_y)(\mathcal{X})}{\partial x_{i,\eta}}$ already stated in Section 4.3.2. Reynolds' theorem is then applied to the resulting integral. Both terms cancel out, for one because x_i does not depend on y_i and for the other because $\partial Pow^\phi(x_i)$ is also independent on y_i.

Second **derivative** of Φ **with respect to a site** x_i **and a neighboring site** y_{i_k} **in the opposite domain**

$$\begin{aligned}\frac{\partial^2(\Phi_x+\Phi_y)(\mathcal{X})}{\partial x_{i,\eta}\,\partial y_{i_k,\eta'}} &= \frac{\partial 2\int_{x\in Pow^\phi(x_i)}(x_i-x)_\eta d\mu(x)}{\partial y_{i_k,\eta'}} \\ &= 2\cdot\int_{x\in Pow^\phi(x_i)}\frac{\partial(x_i-x)_\eta}{\partial y_{i_k,\eta'}}d\mu(x) \\ &+ 2\cdot\int_{x\in\partial Pow^\phi(x_i)}\left\langle\frac{\partial x}{\partial y_{i_k,\eta'}},n_i(x)\right\rangle(x_i-x)_\eta d\mu(x) \\ &= 0\end{aligned} \tag{4.26}$$

This computation follows the same path as that of Section 4.4.1.

4.4.2 Derivatives with respect to the weights

Second derivative of Φ **with respect to the same weight** ϕ_i **twice**

$$\begin{aligned}\frac{\partial^2(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i^2} &= \frac{\partial(p_i-\mu(Pow^\phi(x_i)))}{\partial\phi_i} \\ &= -\int_{x\in\partial Pow^\phi(x_i)}\left\langle\frac{\partial x}{\partial\phi_i},n_i(x)\right\rangle d\mu(x) \\ &= -\sum_{k=1}^{K_i}\frac{1}{2\cdot\|x_i-x_{i_k}\|}\int_{x\in E_{i,i_k}}d\mu(x)\end{aligned} \tag{4.27}$$

The first line of the computation is obtained by substituting the value of $\frac{\partial(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i}$ already stated in Section 4.3.2. Only the value of $\mu(Pow^\phi(x_i)) = \int_{x\in Pow^\phi(x_i)}d\mu(x)$ depends on ϕ_i, and Reynolds' theorem is applied to compute the derivative of this integral. The term integrating over $Pow^\phi(x_i)$ disappears as $\frac{\partial 1}{\partial\phi_i}=0$, and only $\int_{x\in\partial Pow^\phi(x_i)}\left\langle\frac{\partial x}{\partial\phi_i},n_i(x)\right\rangle d\mu(x)$ remains. The last line is computed by splitting the integral over $\partial Pow^\phi(x_i)$ into its edges and substituting $\left\langle\frac{\partial x}{\partial\phi_i},n_i(x)\right\rangle$ for its already-computed value (Equation 4.13).

Second **derivative** of Φ **with respect to a cell's weight** ϕ_i **and the weight** ϕ_{i_k} **of a neighboring cell**

$$\begin{aligned}\frac{\partial^2(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i\,\partial\phi_{i_k}} &= \frac{\partial(p_i-\mu(Pow^\phi(x_i)))}{\partial\phi_{i_k}}\\ &= -\int_{x\in\partial Pow^\phi(x_i)}\left\langle\frac{\partial x}{\partial\phi_{i_k}},n_i(x)\right\rangle d\mu(x)\\ &= \int_{x\in\partial Pow^\phi(x_i)}\left\langle\frac{\partial x}{\partial\phi_{i_k}},n_{i_k}(x)\right\rangle d\mu(x)\\ &= \frac{1}{2\cdot\|x_i-x_{i_k}\|}\int_{x\in E_{i,i_k}} d\mu(x)\end{aligned} \tag{4.28}$$

This computation is very similar to that in Section 4.4.2. The only difference comes from the fact that, for $x\in\partial Pow^\phi(x_i)$, $\frac{\partial x}{\partial\phi_{i_k}}\neq 0$ only if $x\in E_{i,i_k}$. Thus, only the integral over E_{i,i_k} remains.

Second **derivative** of Φ **with respect to a cell's weight** ϕ_i **and the weight** ψ_i **of the corresponding cell in the opposite domain**

$$\begin{aligned}\frac{\partial^2(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i\,\partial\psi_i} &= \frac{\partial(p_i-\mu(Pow^\phi(x_i)))}{\partial\psi_i}\\ &= 0\end{aligned} \tag{4.29}$$

The value of $\frac{\partial(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i}=p_i-\mu(Pow^\phi(x_i))$ is differentiated with respect to ψ_i. None of these terms depend on ψ_i, thus the final derivative is null.

By an identical computation, $\frac{\partial^2(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i\,\partial\psi_{i_k}}=0$.

4.4.3 Crossed derivatives

Second **derivative** of Φ **with respect to the weight** ϕ_i **and the site's coordinates** x_i **of a cell**

$$\begin{aligned}\frac{\partial^2(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i\,\partial x_{i,\eta}} &= \frac{\partial(p_i-\mu(Pow^\phi(x_i)))}{\partial x_{i,\eta}}\\ &= -\int_{x\in\partial Pow^\phi(x_i)}\left\langle\frac{\partial x}{\partial x_{i,\eta}},n_i(x)\right\rangle d\mu(x)\\ &= \sum_{k=1}^{K_i}\frac{1}{\|x_i-x_{i_k}\|}\int_{x\in E_{i,i_k}}(x_i-x)_\eta d\mu(x)\end{aligned} \tag{4.30}$$

The value of $\frac{\partial(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i}=p_i-\mu(Pow^\phi(x_i))$ is differentiated with respect to x_i. Reynolds' theorem is applied to $\mu(Pow^\phi(x_i))=\int_{x\in Pow^\phi(x_i)}d\mu(x)$. $\frac{\partial 1}{\partial x_{i,\eta}}=0$, so only the integral over $\partial Pow^\phi(x_i)$ remains. The integral is then split over the edges of $\partial Pow^\phi(x_i)$ and the normal component of the derivative replaced by its value (Equation 4.14).

Second **derivative with respect to the weight** ϕ_i **of a cell and the coordinates** x_{i_k} **of a neighboring cell's site**

$$\begin{aligned}\frac{\partial^2(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i\,\partial x_{i_k,\eta}} &= \frac{\partial(p_i-\mu(Pow^\phi(x_i)))}{\partial x_{i_k,\eta}}\\ &= -\int_{x\in\partial Pow^\phi(x_i)}\left\langle\frac{\partial x}{\partial x_{i_k,\eta}},n_i(x)\right\rangle d\mu(x)\\ &= \int_{x\in E_{i,i_k}}\left\langle\frac{\partial x}{\partial x_{i_k,\eta}},n_{i_k}(x)\right\rangle d\mu(x)\\ &= -\frac{1}{\|x_i-x_{i_k}\|}\int_{x\in E_{i,i_k}}(x_{i_k}-x)_\eta d\mu(x)\end{aligned} \tag{4.31}$$

This computation follows the same path as that in Section 4.4.3, but only the integral over E_{i,i_k} remains.

Second **derivative with respect to the weight ϕ_i of a cell and the coordinates y_i of the corresponding site in the opposite domain**

$$\begin{aligned}\frac{\partial^2(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i\,\partial y_{i,\eta}} &= \frac{\partial(p_i-\mu(Pow^{\phi}(x_i)))}{\partial y_{i,\eta}} \\ &= 0\end{aligned} \tag{4.32}$$

The value of $\frac{\partial(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i} = p_i - \mu(Pow^{\phi}(x_i))$ is differentiated with respect to y_i. None of these terms depend on y_i, thus the final derivative is null.

By an identical computation, $\frac{\partial^2(\Phi_x+\Phi_y)(\mathcal{X})}{\partial\phi_i\,\partial y_{i_k,\eta}} = 0$.

Chapter 5

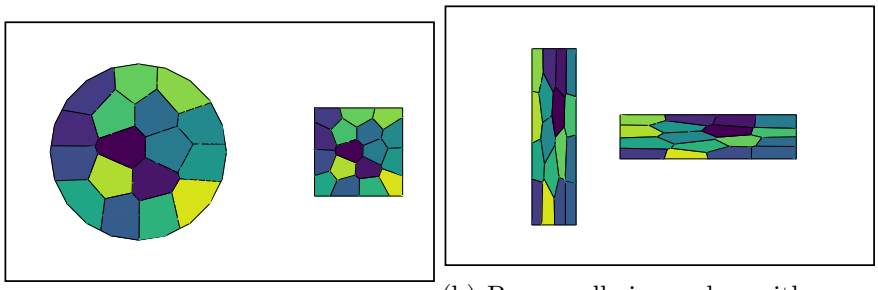
Interpolation

In this chapter, we present how the power cells of coupled transport plans are in one-to-one correspondence. As a consequence, both power diagrams have the same geometry, which allows to linearly interpolate between their vertices in order to approximate displacement interpolation. We present visual results for this interpolation method and qualitatively evaluate them against a reference interpolation.

5.1 Empirical properties of the coupled transport plans

The coupled transport plans specified in 2.2 and obtained by the alternating algorithm of Chapter 4 enjoy some remarkable properties that make them well suited for interpolation. Our algorithm builds two coupled power diagrams. In practice, we observe that neighboring relationships are preserved. This means that cells that are neighbors in a power diagram are also neighbors in the other. In practice, this implies that not only power cells are in correspondence, but also that their geometry is the same: the dual graphs of the power diagrams are identical. Concretely, our algorithm deals with power diagrams intersected with meshes. In the case where boundaries between cells lie outside of the mesh, the neighboring relationship between these cells is broken by the intersection. This appears to correspond to discontinuities in the transport plan — see figures in Section 5.3.1 for an illustration.

We were not able to prove this property, but we observe that it holds in practice. Figure 5.1 showcases it on two examples with same topology. We observe that in both cases the power diagrams keep the same geometry, even when the meshes are distorted in different directions.



(a) Power cells in similar meshes.

(b) Power cells in meshes with same topology but strong anisotropy.

Figure 5.1: In meshes of the same topology, the coupled power diagrams present the same geometry. Even when interpolating between the meshes involves distortion, the empirical property is preserved.

Choice of the number of samples It can sometimes happen that, even when the algorithm has seemingly converged, one cell overlaps between connected components, as shown in Figure 5.2.

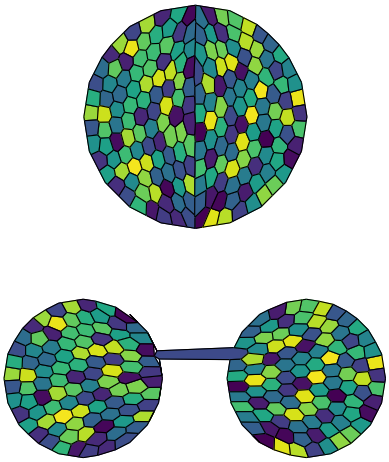


Figure 5.2: A cell intersecting two different connected components in converged transport maps

This is usually due to the ratio between the different connected components' areas being incompatible with the number of cells: for example, a domain composed of two connected components of the same area, transported to an odd number of samples with uniform mass, will have to split a cell between both components in order to fulfill the mass constraints of

optimal transport. This phenomenon is showcased on Figure 5.2, where 201 cells are used to approximate transport between a large disk and two smaller disks of identical areas. This calls for a wise choice of the exact number of samples used.

More precisely, let us consider a mesh made up of k connected components with areas $A_1, \dots, A_k$, of total mass $A = \sum_{i=1}^{k} A_k$. Let's assume that for each connected component, the ratio of its mass to the total mass $\frac{A_i}{A}$ is a rational $\frac{p_i}{q_i}$. Then, in the case where the samples are assigned identical masses, and their number is a multiple of all the denominators q_i, the power cells can be divided among the connected components without a remainder. This results in all power cells only intersecting with one connected component and thus not spanning potential discontinuities. This contributes to the power diagram properly approximating the transport plan and its discontinuities.

5.2 Interpolation algorithm

The fact that cells are in one-to-one correspondence allows us to come up with a simple interpolation technique: since corresponding power cells have the same geometry, we can put their vertices in correspondence and linearly interpolate between them. In addition, power cells boundaries tend to be aligned with the transport maps discontinuities, which helps capture tearing during displacement interpolations — see Figure 5.5 for an example. This allows to define the following 2-d mesh interpolation algorithm.

Our algorithm makes use of restricted power diagrams. We will use Nivoliers' [NYL11] classification of the vertices of such meshes.

Definition 4 (Vertex types)**.** *A vertex of a restricted power diagram is necessarily of one of the three following types:*

- type i *a vertex that originates from the underlying mesh and that does not depend on the (unrestricted) power diagram,*
- type ii *a vertex that is located at the intersection of an edge separating two power cells, and an edge of the underlying mesh,*
- type iii *a vertex that is at the intersection between three cells of the power diagram. Such a vertex can be uniquely identified by the triplet (i, j, k) of surrounding cells indices.*

As a consequence of the preservation of neighboring relationships stated in Section 5.1, and of the fact that each power diagram vertex is uniquely

defined by the power cells it is incident to, it is easy to check whether a vertex of one power diagram can be matched with a vertex of the other, and match them.

When intersecting the power diagram with a mesh, this translates to the fact that each type iii vertex on the one side, encoded as (i, j, k), is in correspondence with at most one type iii vertex, also encoded as (i, j, k), on the other side. We thus identify cell vertices based on their neighboring cells, and linearly interpolate between the two vertices that share the same neighboring cells in the two domains.

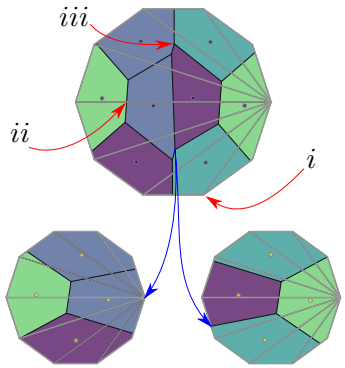


Figure 5.3: **Red arrows** point to vertices with type i, ii and iii, and **the blue arrow** show a vertex that splits during the interpolation. **Grey edges** represent edges that belong to the underlying mesh, while **black edges** come from the restriction of the power diagram to the mesh.

Correspondence between type ii vertices that lie on the boundary follows the same principle. In order to identify which vertices should be associated, we extend the triplet identification presented in Definition 4. If a vertex is adjacent to cells $Pow^{\phi}(x_i)$ and $Pow^{\phi}(x_j)$, it is identified by the triplet $(i, j, 0)$, where 0 represents the fact that the "third cell" it is adjacent to is in fact the outside of the mesh.

However, whenever the meshes present topological discontinuities, some type iii vertices are bound to split during the transport, typically giving birth to two type ii vertices, or one type ii vertex and one type i vertex. In these cases, we need to duplicate the type iii vertices identified by (i, j, k) into two new type ii vertices among $(i, j, 0)$, $(i, 0, k)$ and $(0, j, k)$, and seek corresponding vertices on the other side. A similar treatment is applied to type ii vertices that split into type i vertices.

The fact that there is no one-to-one correspondence between vertices involved in discontinuities offers us a practical criterion for identifying them. This could be used to extract with precision the locus of such discontinuities.

At last, we need to account for type i vertices lying on the boundary. In the best of cases, there is only one vertex identified by $(i, 0, 0)$ in each of the restricted Voronoi diagrams, and we can associate them right away. In the worst case, there are several vertices represented by the same identifier

on each side. In this situation, we chose to associate all the equivalent vertices on one restricted Voronoi diagram to a single vertex of the opposite. This potentially causes some artifacts in the morph and leaves room for improvement of the algorithm. See for example the detail shown in Figure 5.4 where a power cell collapses as two of its vertices are mapped to the same final vertex.

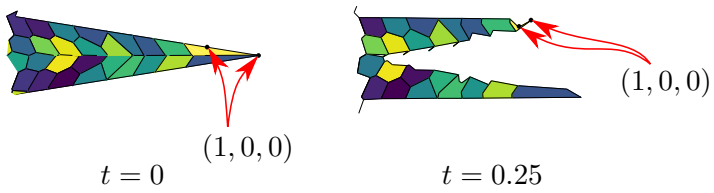


Figure 5.4: Detail from Figure 5.8. The interpolated mesh is shown at two steps of the interpolation. Distinct vertices of the power diagram intersected with the mesh can share the same identification. For example, if the cell at the upper right extremity of the branch has identifier 1, both pointed vertices will be identified by $(1, 0, 0)$ and thus interpolated towards the same final vertex. This results in an jagged outer boundary during the interpolation.

The key difference between this interpolation method and the one proposed by Lévy [Lé14] is that our method interpolates between power diagrams instead of Delaunay triangulations. In our method, the interpolated vertices are the vertices of the power cells, while in the other method the vertices are the samples. The power diagram covers all of the mesh while the Delaunay triangulations only covers a portion of it, as can be seen in Figure 2.4.

5.3 Interpolation results

In this section, we present the interpolated meshes generated through our alternating algorithm and the vertex association method presented in Section 5.2. We present 2-d and 3-d interpolation for visual evaluation. We also compute the Hausdorff distances of interpolated meshes to a reference interpolation, for quantitative evaluation.

5.3.1 Interpolations in 2-d

We compare our symmetrized algorithm with classical semi-discrete optimal transport [Lé14] for 2-d interpolation. The non-symmetrized algorithm interpolates between a sampling of one of the meshes and the barycenters of

a semi-discrete optimal transport plan, that transports from the other mesh to the sampling. Figures 5.5, 5.6, 5.7, 5.8 and 5.9 compare the result of our algorithm with interpolations obtained with the non-symmetrized algorithm. Two versions of the non-symmetrized interpolation are presented, each sampling a different mesh. This showcases the fact that these non-symmetrized interpolations properly capture the discontinuities from at most one of the meshes, while the symmetrized interpolation captures them from both.

Measures with uniform densities Figures 5.5 and 5.6 illustrate our algorithm on sets of disks — two disks interpolated against two other disks, one disk interpolated against two disks, and one disk interpolated against three disks — and compare it with the (non-symmetric) semi-discrete approach of Lévy [Lé14]. When a single disk is interpolated with a shape consisting of two or three disks, the classical semi-discrete approach works well only when sampling the non-connected shape and computing a semi-discrete transport plan from the single disk (thus considered continuous) to these samples. However, appropriately choosing the continuous measure is not possible when interpolating between two non-connected shapes: in that case, our method still nicely captures tearing (Fig. 5.5) while a non-symmetric approach poorly approximates it.

Figure 5.7 shows an interpolation from a single disk to connected but non-convex negatively-curved shapes, thus resulting in discontinuous transport maps by construction [CJL+15]. The non-symmetrized algorithm fails at capturing transport map discontinuities, both when transporting from or to the single disk, while our symmetric approach captures them well.

Figure 5.8 shows a more complex interpolation between stars, where each branch separates into two equal parts during interpolation. Our approach better preserves the thin branch structures.

Measures with non-uniform densities As illustrated in Figure 5.9, our algorithm also handles measures that are not uniform over their support. We observe that the measures are interpolated in a consistent way, with the two peaks from the source distribution splitting and joining into the two peaks from the target distribution. In contrast, the non-symmetrized algorithm results in interpolation of near uniform densities in this example.

5.3.2 Interpolations in 3-d

We showcase our algorithm on 3-d examples.

Figure 5.5: Interpolation between two horizontally aligned disks and two vertically aligned disks. Notice how the cells align with the discontinuities on both measures with our method, while they only align on the target measure with Levy's method, resulting in artifacts in the interpolation.

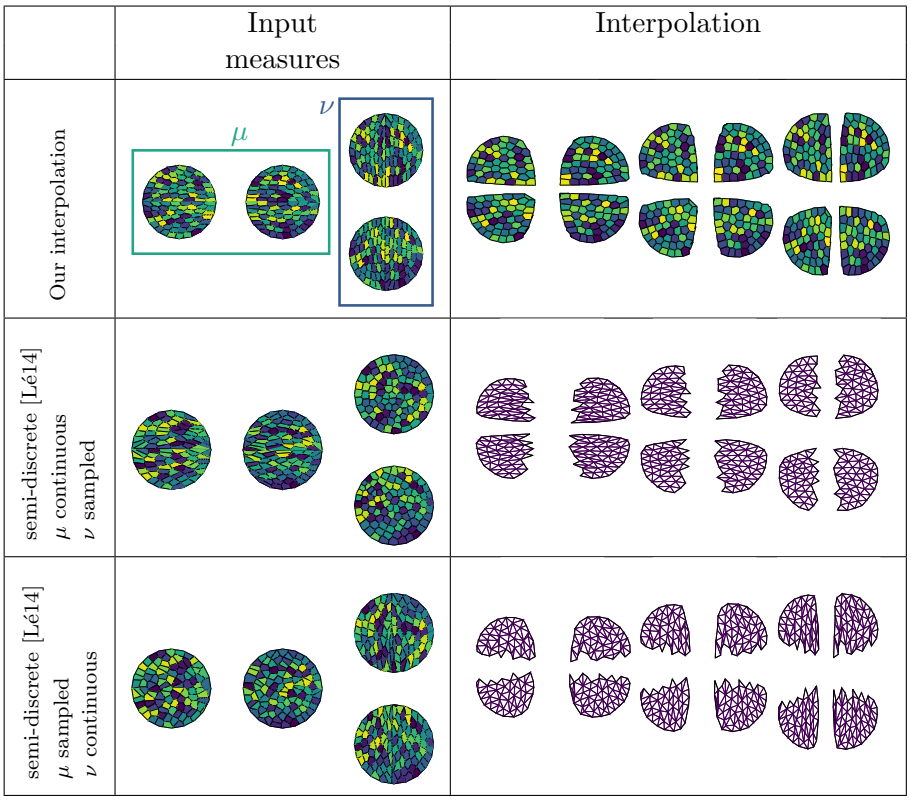


Figure 5.10 illustrates the importance of symmetry in 3-d as well: one can observe the cells aligning with the discontinuities on our algorithm's output, resulting in two rather accurate tearings, while the interpolations from Lévy's algorithm always display a ragged tear in at least one of the discontinuities, in a similar manner as what could be observed on Figure 5.5.

Figure 5.11 illustrates a similar phenomenon to the one observed on Figure 5.10: the power cells of symmetrized the transport map align with the discontinuity on our algorithm's output, while it is only the case with Levy's algorithm when the discontinuity is on the target measure.

5.4 Quantitative evaluation

In order to evaluate the accuracy of our interpolations, we compute a high quality interpolation using Lévy's algorithm with 100k samples and compare the Hausdorff distances between, respectively, our algorithm with 10k samples and the ground truth, and Lévy's algorithm with 10k samples and the ground truth. We observe that our results are systematically more precise than which corroborates the visual observations.

Shape	Algorithm	0	0.25	0.5	0.75	1
Six-pointed star to	Levy's	0.078	0.076	0.074	0.076	0.077
six-pointed star 5.8	Ours	0.028	0.033	0.038	0.044	0.045
Two disks to	Levy's	0.025	0.033	0.025	0.021	0.016
two disks 5.5	Ours	0.0086	0.017	0.0098	0.0094	0.0072
One disk to	Levy's	0.022	0.022	0.025	0.026	0.016
three disks 5.6	Ours	0.0098	0.0089	0.0017	0.0092	0.0064

Table 5.1: Hausdorff distances between the interpolated meshes at different interpolation steps. The mesh generated by the symmetrized algorithm (ours) is closer to the reference mesh than the algorithm generated by Lévy's algorithm.

Figure 5.6: Interpolations between a disk and two disks, and between a disk and three disks, using 200 samples. The non-symmetric approach poorly approximates discontinuities when the single disk is sampled. Our method best captures discontinuities in all cases.

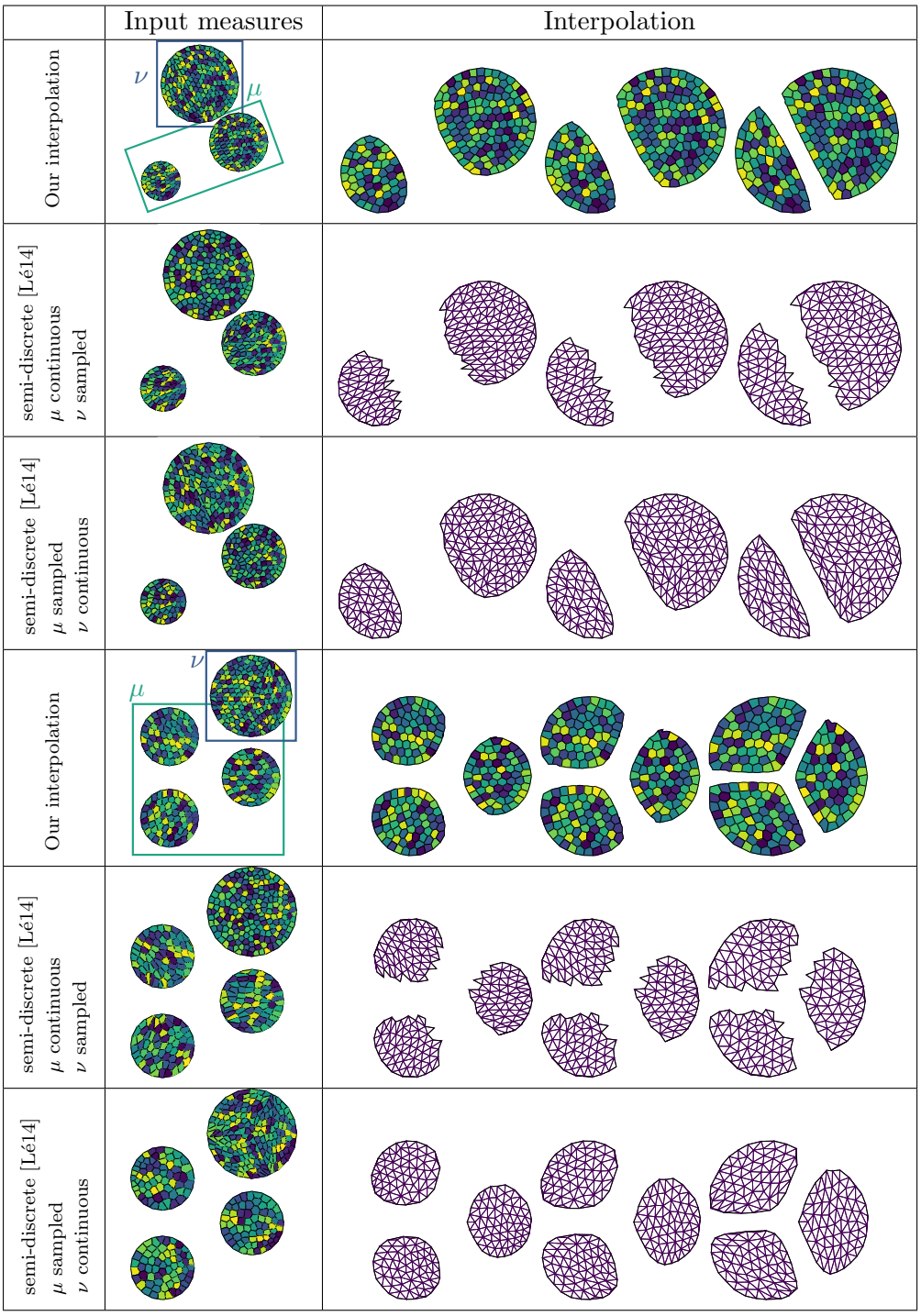


Figure 5.7: Interpolation between a disk and non-convex shapes. A discontinuity appears on the disk to create the non-convexity.

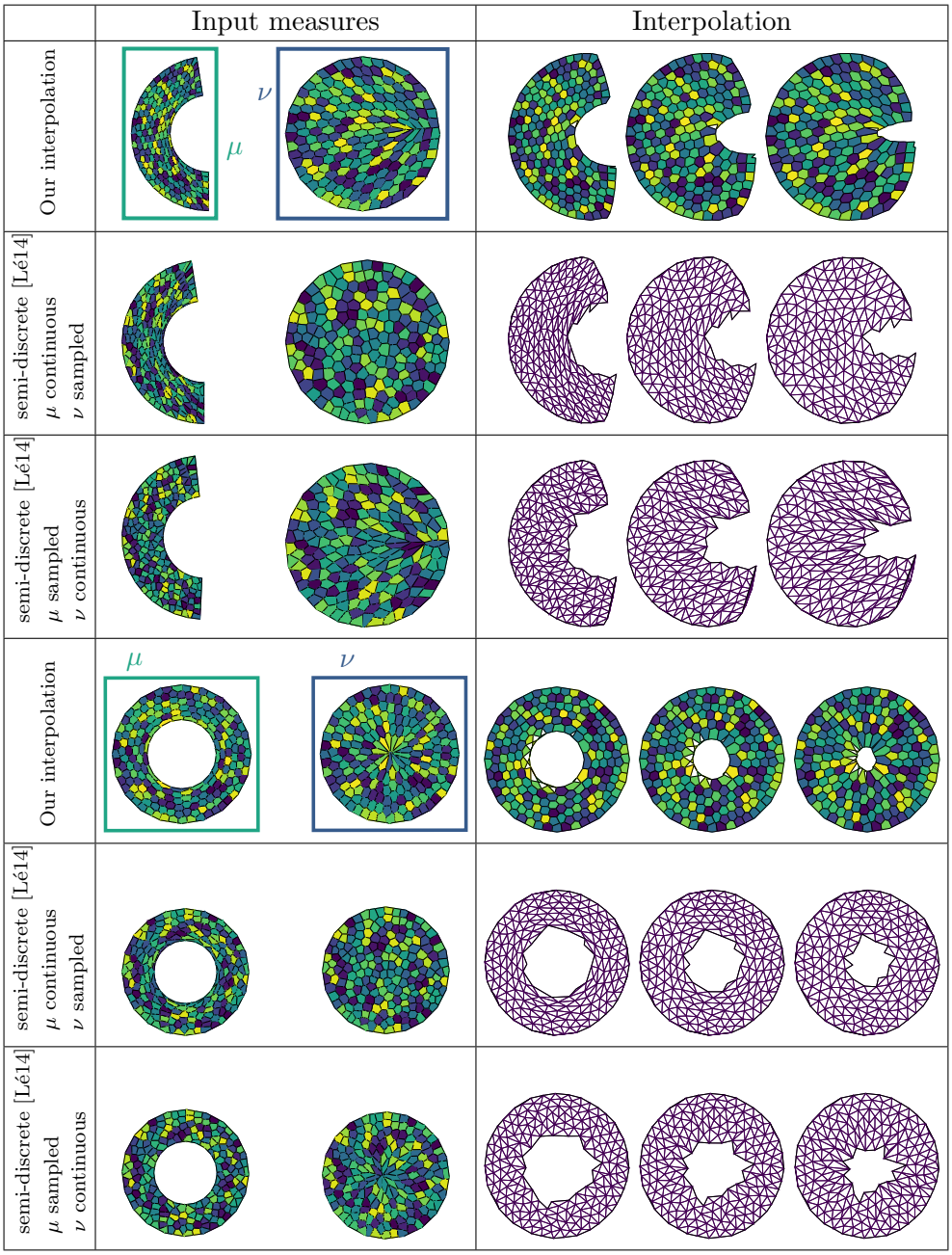


Figure 5.8: Interpolation between a star and another star with branches equidistant to the first star's branches. Each branch splits in two equal parts that join with their neighbors to create the target branches.

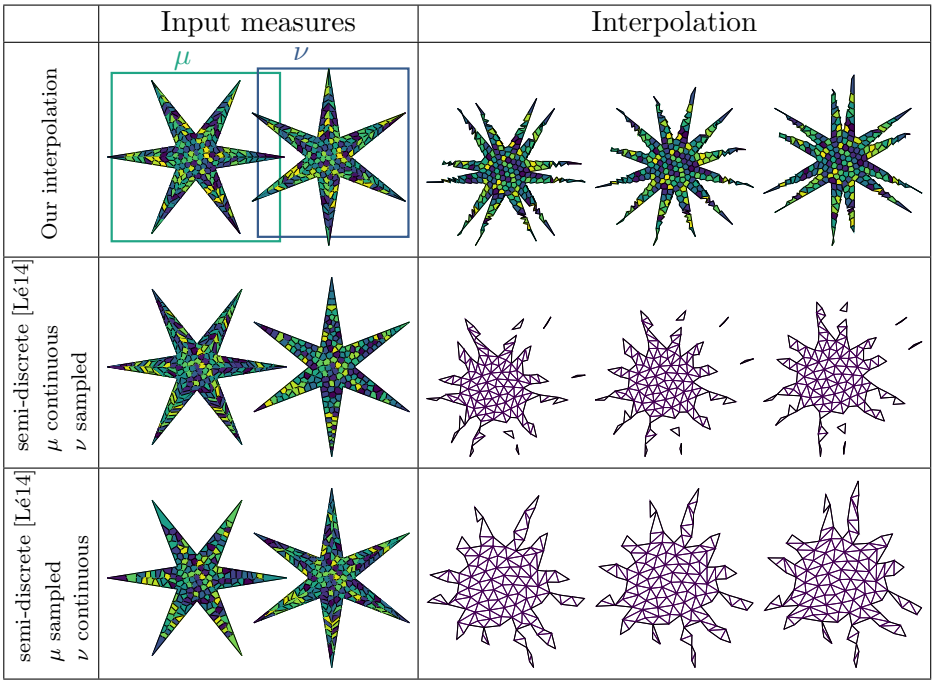


Figure 5.9: Interpolation between two squares equipped with non-uniform densities, each consisting of the sum of two gaussians

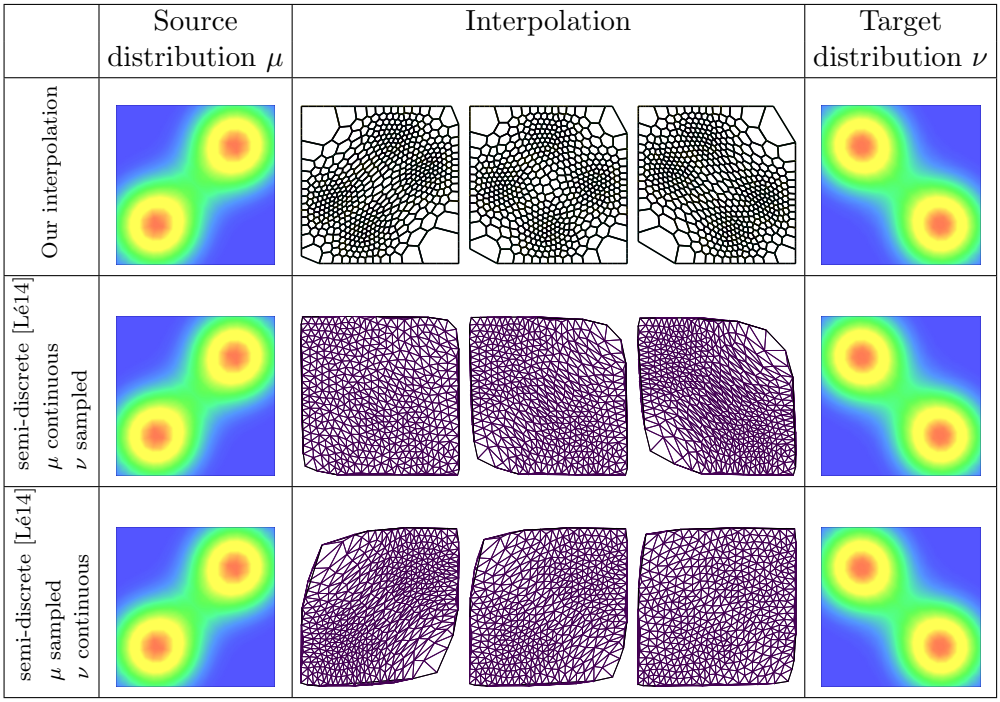


Figure 5.10: Interpolation between two horizontally aligned spheres and two vertically aligned spheres. Like in 2-d, power cells align with the discontinuities in the case of our algorithm, allowing for accurate tearings, while with Levy's algorithm, in each case at least one of the tearings is not accurately captured.

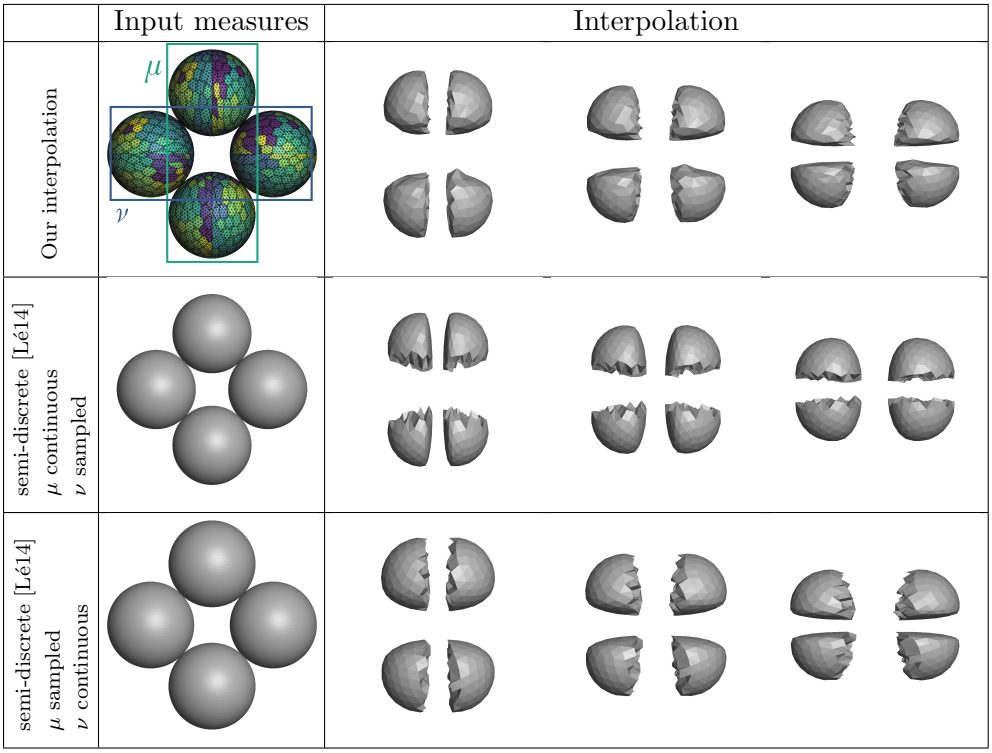
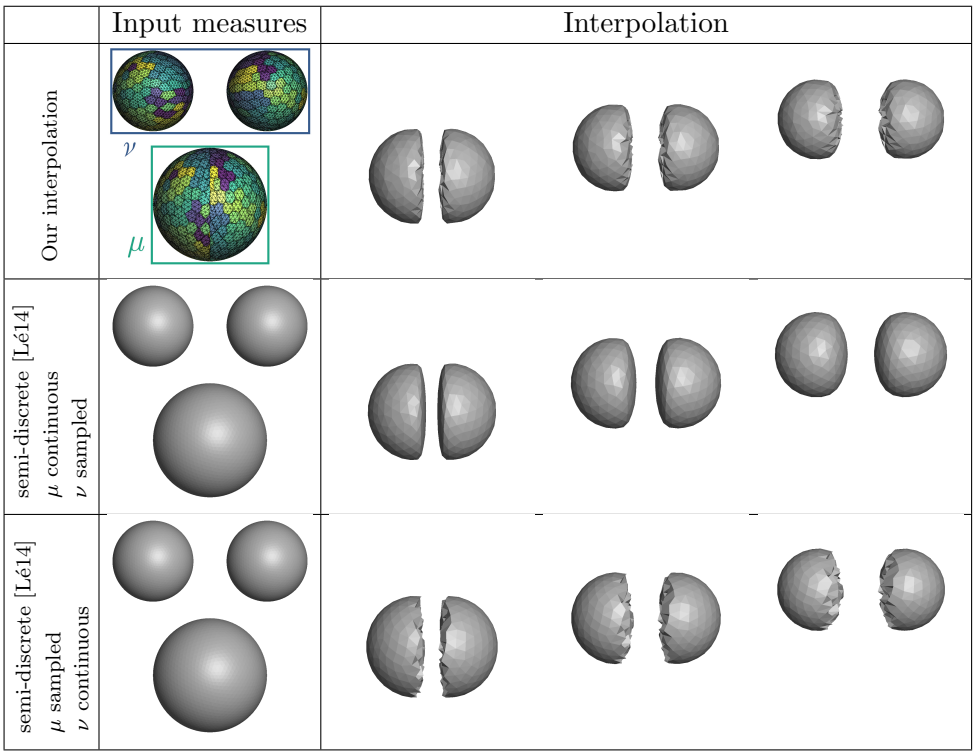


Figure 5.11: Interpolation between a single sphere and two smaller spheres. The power cells align with the discontinuity with our algorithm but only do so in one case with Levy's algorithm.



Conclusion

In this thesis, I studied methods to accurately approximate displacement interpolation using semi-discrete optimal transport. I proposed a way of coupling semi-discrete transport plans in order to reintroduce symmetry into this framework, while accurately capturing the discontinuities that could arise in the interpolation. I proposed a fixed point algorithm to compute such coupled transport plans. It consists in alternatively optimizing the weights of an optimal transport plan and relocating the discrete samples to the barycenters of the resulting power cells. This algorithm gives good practical results, which one can use to linearly interpolate between the resulting power diagram. This constitutes a good approximation of displacement interpolation, including when the discrete approximations of distributions have relatively few samples.

I also studied the feasibility of formulating this problem as a constrained optimization problem, which could then be solved using classical methods such as gradient descent or Newton's method. Despite experimenting with several formulations or optimizers, this has proved inconclusive. I hypothesize that this is due to the over sensitivity of function parameters, and the problem being possibility ill-posed. It seems that at best, this optimization approach could be shown to be solved through our alternating algorithm.

Future work

These results, positive or negative, raise more questions.

Theoretical guarantees I presented a way of coupling transport plans and interpolating between them that, empirically, appears to yield a good approximation of displacement interpolation. However, this result is for now mainly empirical, and could benefit from theoretical guarantees. For example, one could investigate whether the coupling conditions imply that the

image of a power cell through a transport map between the two continuous distributions would be the associated cell in the opposite domain. If this association turns out not to be exact, one could instead turn to quantifying the accuracy of this approximation. This last result could help studying to what extent the power cells boundaries actually align with discontinuities in the transport plan.

Another of our results that could benefit from theoretical guarantees is the convergence of our alternating algorithm. This could for example be done by proving that the norm of the constraint vector presented in 4.2.1 decreases over the algorithm's iterations. If this were to fail, one could at least try to prove that a fixed point of the algorithm does exist. A possibility would be to use Brouwer's fixed point theorem. It demands to verify two hypotheses. First, the functions applied during the algorithm need to be continuous (these functions being the optimal weights associated to a set of sites positions, and the barycenters associated to a power diagram). Second, the definition set needs to be convex, and compact, which, since we are dealing with a finite-dimensional problem, amounts to being closed and bounded. This definition set consists in the set of weights that can define optimal transport plans and the set of positions that can be power cells barycenters.

An alternative way of giving a theoretical foundation to the alternating algorithm would be to consider it as a sequence of alternating projections. In its classical version, the alternating projection method is used to compute the intersection of closed, convex sets [GPR67]. This method has been extended to sets that are not convex, such as manifolds [LM08]. Studying if the sets involved in our problem fulfill conditions would certainly provide valuable theoretical information about this problem.

Descent algorithms There might be optimization approaches that I did not consider but that would be adapted to solving the constrained problem. Studying the topological properties of the set of variables verifying the barycentric constraint could orient the choice of an adequate optimization algorithm. I worked with explicit expressions of derivatives, but one could also experiment with autodifferentiation, which might be more robust to edge-cases.

The optimization method we experimented implicitly assumed that the weights and sites positions were independent variables. This is not true: since our semi-discrete transport plans are optimal, the weights are uniquely defined for a given set of sites positions (up to an additive constant). This is

in accordance with the observation made in Section 4.2.2, which attributed the failure of our previous methods to the weights not being updated to take into account the new sites positions. This also hints at the reason behind the efficiency of the alternating algorithm, in which each relocation of the sites was followed by a computation of the new optimal transport plan.

A promising direction for a new method can be derived from this observation. The implicit function theorem guarantees that the function ϕ^* associating the optimal weights (ϕ_i) to a set of sites positions (x_i) exists and is sufficiently regular. We can then compute the derivative of ϕ^* and reinject it in a Newton algorithm in order to cancel out the barycentric constraints. Conceptually, this is an inversion of the point of view adopted in this manuscript: instead of optimizing transport functional under barycentric constraints, the objective is to find barycenters through optimization on the constrained space defined by optimal transport weights.

Applications Detecting discontinuities in transport plans is particularly important in applications where the optimal transport formulation is the direct translation of physical principles, such as fluid simulations [Lé22, LMvH21]. In these applications, the cost minimization typically translates the least action principle, and the pushforward measures equality translates matter conservation. In these applications, detecting discontinuities in the transport plan can be used to partition matter early in the simulation according to which route it will take in the future.

Extensions Wasserstein barycenters are most interesting when applied to more than two distributions. The approach presented in this thesis heavily relies on the fact that we only consider two distributions. By consequence, it is not trivial to define a specification coupling more than two distributions in the same fashion while preserving the same properties. One could, for example, imagine chaining transport plans through their barycenters, the sites of the k-th transport plan being located at the barycenters of the cells of the $k-1$-th transport plan.

Bibliography

[AC11] Martial Agueh and Guillaume Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011. 1.1.4

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. 2017. (document), 1.5

[AGS05] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savare. Gradient flows in metric spaces and in the space of probability measures. 01 2005. 1.1.2

[AHA98] Franz Aurenhammer, Friedrich Hoffmann, and Boris Aronov. Minkowski-type theorems and least-squares clustering. *Algorithmica*, 20(1):61–76, 1998. (document), 1.2.1, 1.2.2

[Bal09] Michael Balzer. Capacity-constrained voronoi diagrams in continuous spaces. In *2009 Sixth International Symposium on Voronoi Diagrams*, pages 79–88, 2009. 1.2.2, 1.3.2

[Ban19] Mohit Bansil. Two stage algorithm for semi-discrete optimal transport on disconnected domains, 2019. 1.2.2

[BB96] Heinz H. Bauschke and Jonathan M. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Review*, 38(3):367–426, September 1996. 2.3.1

[BSD09] Michael Balzer, Thomas Schlömer, and Oliver Deussen. Capacity-constrained point distributions. *ACM Transactions on Graphics*, 28(3):1–8, July 2009. 1.3.2

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 4.2.1

[BvdPPH11] Nicolas Bonneel, Michiel van de Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using lagrangian mass transport. *ACM Transactions on Graphics (TOG)*, 30(6):1–12, December 2011. (document)

[Caf92] Luis A Caffarelli. The regularity of mappings with a convex potential. *Journal of the American Mathematical Society*, 5(1):99–104, 1992. (document), 2

[CCS18] Sebastian Claici, Edward Chien, and Justin Solomon. Stochastic wasserstein barycenters. *CoRR*, abs/1802.05757, 2018. (document)

[CJL+15] Otis Chodosh, Vishesh Jain, Michael Lindsey, Lyuboslav Panchev, and Yanir A Rubinstein. On discontinuity of planar optimal transport maps. *Journal of Topology and Analysis*, 7(02):239–260, 2015. (document), 2, 5.3.1

[CMB04] Jorge Cortes, Sonia Martinez, and Francesco Bullo. Spatially-distributed coverage optimization and control with limited-range interactions, 2004. 4.3.1

[Cut13] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transportation distances. 2013. (document), 1.5

[DCSA+14] Julie Digne, David Cohen-Steiner, Pierre Alliez, Fernando Goes, and Mathieu Desbrun. Feature-preserving surface reconstruction and simplification from defect-laden point sets. *J. Math. Imaging Vis.*, 48(2):369–382, feb 2014. (document), 1.5

[DEJ06] Qiang Du, Maria Emelianenko, and Lili Ju. Convergence of the lloyd algorithm for computing centroidal voronoi tessellations. *SIAM Journal on Numerical Analysis*, 44(1):102–119, January 2006. 1.3.1

[DFG99] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, January 1999. 1.3, 1.3.1, 1.3.1, 1.3.1, 1.3.1

[DGBOD12] Fernando De Goes, Katherine Breeden, Victor Ostromoukhov, and Mathieu Desbrun. Blue Noise through Optimal Transport. *ACM Transactions on Graphics*, 31:171:1–171:1, December 2012. 1.2.1, 1.3.2, 4.3.2, A

[dGKL18] Frédéric de Gournay, Jonas Kahn, and Léo Lebrat. Differentiation and regularity of semi-discrete optimal transport with respect to the parameters of the discrete measure, 2018. 4.4

[DSS10] Julie Delon, Julien Salomon, and Andrei Sobolevski. Fast transport optimization for monge costs on the circle. *SIAM Journal on Applied Mathematics*, 70(7):2239–2258, 2010. (document), 1.5

[DT10] Ayelet Dominitz and Allen Tannenbaum. Texture mapping via optimal mass transport. *IEEE transactions on visualization and computer graphics*, 16:419–33, 07 2010. 1.4.2

[GLR17] Bruno Galerne, Arthur Leclaire, and Julien Rabin. Semi-discrete optimal transport in patch space for enriching gaussian textures. In Frank Nielsen and Frédéric Barbaresco, editors, *Geometric Science of Information*, pages 100–108, Cham, 2017. Springer International Publishing. 1.4.5, 1.8

[GM18] Thomas O Gallouët and Quentin Mérigot. A lagrangian scheme à la brenier for the incompressible euler equations. *Foundations of Computational Mathematics*, 18(4):835–865, 2018. 3.1.2

[GPC18] Aude Genevay, Gabriel Peyre, and Marco Cuturi. Learning generative models with sinkhorn divergences. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1608–1617. PMLR, 09–11 Apr 2018. (document), 1.5

[GPR67] L.G. Gubin, B.T. Polyak, and E.V. Raik. The method of projections for finding the common point of convex sets. *USSR Computational Mathematics and Mathematical Physics*, 7(6):1–24, 1967. 5.4

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. A

[KMT16] Jun Kitagawa, Quentin Mérigot, and Boris Thibert. Convergence of a newton algorithm for semi-discrete optimal transport, 2016. 1.2, 1.2.1, 1.2.2, 1.2.2, 4.2.1

[LB12] Bruno Lévy and Nicolas Bonneel. Variational Anisotropic Surface Meshing with Voronoi Parallel Linear Enumeration. In *Proceedings of the 21st International Meshing Roundtable (IMR'12)*, October 2012. 3.1.1

[LBG80] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, 1980. 1.3.1

[Liu10] Yang Liu. HLBFGS: a hybrid L-BFGS optimization framework, 2010. 3.1.2

[Llo82] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982. 1.3.1

[LM08] Adrian S. Lewis and Jérôme Malick. Alternating projections on manifolds. *Mathematics of Operations Research*, 33(1):216–234, 2008. 5.4

[LMvH21] Bruno Lévy, Roya Mohayaee, and Sebastian von Hausegger. A fast semidiscrete optimal transport algorithm for a unique reconstruction of the early universe. *Monthly Notices of the Royal Astronomical Society*, 506(1):1165–1185, Jun 2021. (document), 1.4.3, 5.4

[LWL+09] Yang Liu, Wenping Wang, Bruno Lévy, Feng Sun, Dong-Ming Yan, Lin Lu, and Chenglei Yang. On Centroidal Voronoi Tessellation–Energy Smoothness and Fast Computation. *ACM Transactions on Graphics*, 28(4):Article 101, August 2009. 1.3.1

[Lé14] Bruno Lévy. A numerical algorithm for l_2 semi-discrete optimal transport in 3d, 2014. 2, (document), 1.1, 1.1.2, 1.2.1, 1.2.2, 1.4.1, 3, 1.4, 2, 2.1, 4, 3.1.1, 3.2, 5.2, 5.3.1, 5.3.1, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11

[Lé19] Bruno Lévy. Graphite, 2019. 3.1.2, 1, 4.3

[Lé22] Bruno Lévy. Partial optimal transport for a constant-volume lagrangian mesh with free boundaries. *Journal of Computational Physics*, 451:110838, feb 2022. 1.4.3, 1.6, 5.4

[McC97] Robert J. McCann. A convexity principle for interacting gases. *Advances in Mathematics*, 128(1):153–179, 1997. 1.1.4

[Mér11] Quentin Mérigot. A multiscale approach to optimal transport. *Computer Graphics Forum*, 30(5):1584–1592, August 2011. 18 pages. 1.2.2, 2

[Mey19] Jocelyn Meyron. Initialization procedures for discrete and semi-discrete optimal transport. *Computer-Aided Design*, May 2019. 1.2.2, 3

[MMT17a] Quentin Mérigot, Jocelyn Meyron, and Boris Thibert. An algorithm for optimal transport between a simplex soup and a point cloud. *CoRR*, abs/1707.01337, 2017. 1.2.2

[MMT17b] Quentin Mérigot, Jocelyn Meyron, and Boris Thibert. Light in power: A general and parameter-free algorithm for caustic design. *CoRR*, abs/1708.04820, 2017. 1.4.4, 1.7, 4.3

[Mon81] Gaspard Monge. *Mémoire sur la théorie des déblais et des remblais*. Académie Royale des Sciences de Paris, 1781. 1.1.1

[MS02] Paul Milgrom and Ilya Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002. 1.2.2

[Noc80] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980. 1.3.2

[NYL11] Vincent Nivoliers, Dong-Ming Yan, and Bruno Lévy. Fitting Polynomial Surfaces to Triangular Meshes with Voronoi Squared Distance Minimization. pages 601–617, October 2011. The original publication is available at www.springerlink.com. 5.2

[PC+17] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Center for Research in Economics and Statistics Working Papers*, (2017-86), 2017. (document)

[RDG10] Julien Rabin, Julie Delon, and Yann Gousseau. Regularization of transportation maps for color and contrast transfer. *2010 IEEE International Conference on Image Processing*, pages 1933–1936, 2010. (document), 1.5, 1.9

[Ree81] William T. Reeves. Inbetweening for computer animation utilizing moving point constraints. In *Proceedings of the 8th Annual*

Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '81, page 263–269, New York, NY, USA, 1981. Association for Computing Machinery. (document)

[Rey03] Osborne Reynolds. *The sub-mechanics of the universe*, volume 3. University Press, 1903. 4.3.1

[RPDB11] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer, 2011. (document)

[RTG04] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 2004. (document), 1.5

[San15] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015. (document)

[Sca15] Raymond P. Scaringe. Apollonius solutions in rd. *CoRR*, abs/1512.08846, 2015. 1.2

[Tó01] G.Fejes Tóth. A stability criterion to the moment theorem. *Studia Scientiarum Mathematicarum Hungarica*, 38(1-4):209 – 224, 2001. 1.3.1

[Vil09] Cédric Villani. *Optimal transport : old and new*. Grundlehren der mathematischen Wissenschaften. Springer, Berlin, 2009. (document), 1.1, 1.1.4

[XLC+16] Shi-Qing Xin, Bruno Lévy, Zhonggui Chen, Lei Chu, Yaohui Yu, Changhe Tu, and Wenping Wang. Centroidal power diagrams with capacity constraints. *ACM Transactions on Graphics*, 35(6):1–12, November 2016. 1.3.2

[ZSG+13] Xin Zhao, Zhengyu Su, Xianfeng Gu, Arie Kaufman, Jian Sun, Jie Gao, and Feng Luo. Area-preservation mapping using optimal mass transport. *IEEE transactions on visualization and computer graphics*, 19:2838–47, 12 2013. 1.4.2, 1.5

Appendix A

Gradient descent algorithms

We have experimented with a solution that incorporates the constraints and $(\Phi_x + \Phi_y)$ into a single objective function using a quadratic barrier. A quadratic barrier adds the constraints in the objective function within a highly weighted squared norm term. Doing so adds extreme penalty when constraints are violated.

The resulting objective function to minimize would be, in our case:

$$-(\Phi_x(\mathcal{X}) + \Phi_y(\mathcal{X})) + A \cdot \left(\sum_{i=1}^{N} \|c_{x,i}(\mathcal{X})\|^2 + \|c_{y,i}(\mathcal{X})\|^2 \right)$$

where A is orders of magnitude higher than $\frac{\Phi_x + \Phi_y}{c_{x,i} + c_{y,i}}$ in a neighborhood of a solution. We incorporate the unconstrained objective function in its negative form $-(\Phi_x + \Phi_y)$ in order to minimize it, to stay coherent with the minimization of the quadratic barrier.

However, all gradient-based methods for this problem are bound to fail, because of the fact that Φ_x and Φ_y are concave only in ϕ_i and ψ_i, and that they're actually *convex* in x_i and y_i. This can be seen when considering the fact that $\nabla_{x_i}(\Phi_x) = 2 \int_{x \in Pow^{\phi}(x_i)} (x - x_j) d\mu(x)$. A gradient ascent would thus drive the sites as far away as possible from their cells' barycenters, with a speed proportional to the site distance to the barycenter. Conversely, a gradient descent would direct the sites towards their cells' barycenters. This property was successfully exploited in applications such as Blue Noise through Optimal Transport [DGBOD12], where the objective was to force the sites to coincide with the cells' barycenters, thus the problem could be formulated in terms of searching for a saddle point.

In our problem, however, neither a gradient descent nor a gradient ascent suit our needs. Since our typical use case involves computing symmetrized

optimal transport between disjunct meshes, the barycenter of a cell and the barycenter of its associated cell do not coincide. Thus, a gradient descent would defeat our purpose.

This issue is intrinsic to all methods using the gradient as a descent direction, and cannot be solved by using a different or more refined optimization technique, such as stochastic gradient or Adam [KB14].