



HAL
open science

Projection in the 2-Wasserstein sense on structured measure space

Léo Lebrat

► **To cite this version:**

Léo Lebrat. Projection in the 2-Wasserstein sense on structured measure space. Probability [math.PR].
INSA de Toulouse, 2019. English. NNT : 2019ISAT0035 . tel-02899953

HAL Id: tel-02899953

<https://theses.hal.science/tel-02899953>

Submitted on 15 Jul 2020

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

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



THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue le 19/12/2019 par :

LÉO LEBRAT

**Projection au sens de Wasserstein 2 sur des espaces structurés
de mesures**

JURY

FRANCK BARTHE	Professeur des Universités	Examineur
JÉRÔME BOLTE	Professeur des Universités	Examineur
FRÉDÉRIC DE GOURNAY	Maître de conférences	Directeur
JONAS KAHN	Chargé de recherche CNRS	Co-directeur
BRUNO LÉVY	Directeur de recherche	Rapporteur
QUENTIN MÉRIGOT	Professeur des Universités	Examineur
FILIPPO SANTAMBROGIO	Professeur des Universités	Examineur
GABRIELE STEIDL	Professeur des Universités	Rapporteur

École doctorale et spécialité :

MITT : Domaine Mathématiques : Mathématiques appliquées

Unité de Recherche :

Institut de Mathématiques de Toulouse (UMR 5219)

Directeur(s) de Thèse :

Frédéric de Gournay et Jonas Kahn

Rapporteurs :

Bruno Lévy et Gabriele Steidl

*À mon parrain,
Laurent.*

Table des matières

1	Introduction	1
1.1	Distance de transport optimal	5
1.2	Calcul numérique de la distance de Transport optimal	7
1.2.1	Transport optimal discret	7
1.2.2	Transport optimal continu	9
1.2.3	Transport optimal semi-discret	9
1.3	Génération de schéma d'échantillonnage en IRM	13
1.4	Plan de la thèse	14
2	Introduction	17
2.1	Optimal transport distance	21
2.2	Numerical computation of the optimal transport distance	23
2.2.1	Discrete optimal transport	23
2.2.2	Continuous optimal transport	25
2.2.3	Semi-discrete optimal transport	25
2.3	MRI sampling scheme generation	28
2.4	Thesis plan	30
3	Approches variationnelles pour le stippling : distances L^2 ou transport optimal ?	33
3.1	Introduction	35
3.2	Méthodes numériques	36
3.2.1	Distance de convolution	36
3.2.2	W_2 discret	37
3.2.3	W_2 discret régularisé	38
3.2.4	W_2 semi-discret	39
3.3	Comparaison des méthodes	40

4	Differentiation and regularity of semi-discrete optimal transport	47
4.1	Introduction	49
4.1.1	Semi-discrete optimal transport	49
4.1.2	Link with Voronoi diagrams	51
4.1.3	Organization of the paper	52
4.2	Second order differentiability	52
4.2.1	Hypotheses and notation	53
4.2.2	Main result	55
4.2.3	The Euclidean case	56
4.2.4	Technical lemmas	58
4.2.5	Proof of the results of Section 4.2	64
4.3	Numerical experiments	66
4.3.1	Lloyd's algorithm	68
4.3.2	Newton's Algorithm	69
4.3.3	Other considerations	70
4.3.4	Numerical results	70
4.3.5	Direct comparaisn of the algorithms	70
4.3.6	Adding a point	73
4.4	Conclusion	73
5	Optimal Transport Approximation of 2-Dimensional Measures	77
5.1	Introduction	79
5.1.1	Contributions	80
5.1.2	Related works	80
5.1.3	Paper outline	83
5.2	The minimization framework	83
5.2.1	Discretization	83
5.2.2	Overall algorithm	84
5.3	Computing the Wassertein distance F : ψ -step	85
5.3.1	Semi-discrete optimal transport	85
5.3.2	Solving the dual problem	87
5.3.3	Numerical integration	91
5.4	Optimizing the weights and the positions : \mathbf{w} and \mathbf{x} steps	92
5.4.1	Computing the optimal weights	92
5.4.2	Gradient $\nabla_{\mathbf{x}} F$ and the metric Σ_k	93
5.5	Links with other models	94

5.5.1	Special cases of the framework	94
5.5.2	Comparison with electrostatic halftoning	95
5.6	Projections on curves spaces	99
5.6.1	Discrete curves	99
5.6.2	Numerical projectors	102
5.6.3	Numerical examples	104
5.6.4	Multiresolution implementation	104
5.7	Applications	105
5.7.1	Non Photorealistic Rendering with curves	105
5.7.2	Path planning	108
5.8	Theoretical convergence of Algorithm 3	110
5.9	Outlook	111
6	3/4 discrete optimal transport	113
6.1	Introduction	115
6.2	Setting	118
6.2.1	Reminders on 2-Wasserstein distance	118
6.2.2	Setting	119
6.3	Derivatives of the cost function	121
6.3.1	First order derivative with respect to ϕ	121
6.3.2	Computation of the second order derivative with respect to ϕ	121
6.3.3	Computation of the first order derivative with respect to P and ρ	123
6.4	Numerical implementation	127
6.4.1	Integration computation	127
6.4.2	Parallelism	129
6.5	Computation of the optimal transport	130
6.5.1	Choice of optimization method	132
6.6	Numerical examples	137
6.6.1	Representation of picture	137
6.7	Conclusion	139
7	Approximation of continuous curves	141
7.1	Introduction	143
7.2	Notation	144
7.3	Main result	147
7.4	Proof of the theorems	151

7.4.1	Notations and technical lemmas	151
7.4.2	Approximation of function by splines	153
7.4.3	Approximation of splines by functions	156
7.4.4	Wasserstein distance	159
7.4.5	Proof of theorems	163
7.5	Numerical implementation	165
7.6	Outlook	165
8	Numerical issues	169
8.1	Performing the integration over Laguerre cells	170
8.1.1	Green's method of integration	171
8.1.2	Detail of the computations	172
8.1.3	Complexity analysis	175
8.2	Standard tools of optimization and computational linear algebra	177
8.2.1	Basin of attraction of the Newton method	177
8.2.2	Damped Newton method	179
8.2.3	Condition number and its effect in solving a linear system	180
8.3	Optimal transport optimization for curvling	182
8.3.1	Geometric properties of Laguerre tessalations in the curvling setting	182
8.3.2	Condition number of the Hessian in OT	183
8.3.3	Size of the basin of attraction in the semi-discrete OT .	187
8.3.4	Non definite Hessian in OT	189
8.3.5	Regularized Newton method for Curvling	192
8.3.6	Multiscale algorithm for Curvling	195

Chapitre 1

Introduction

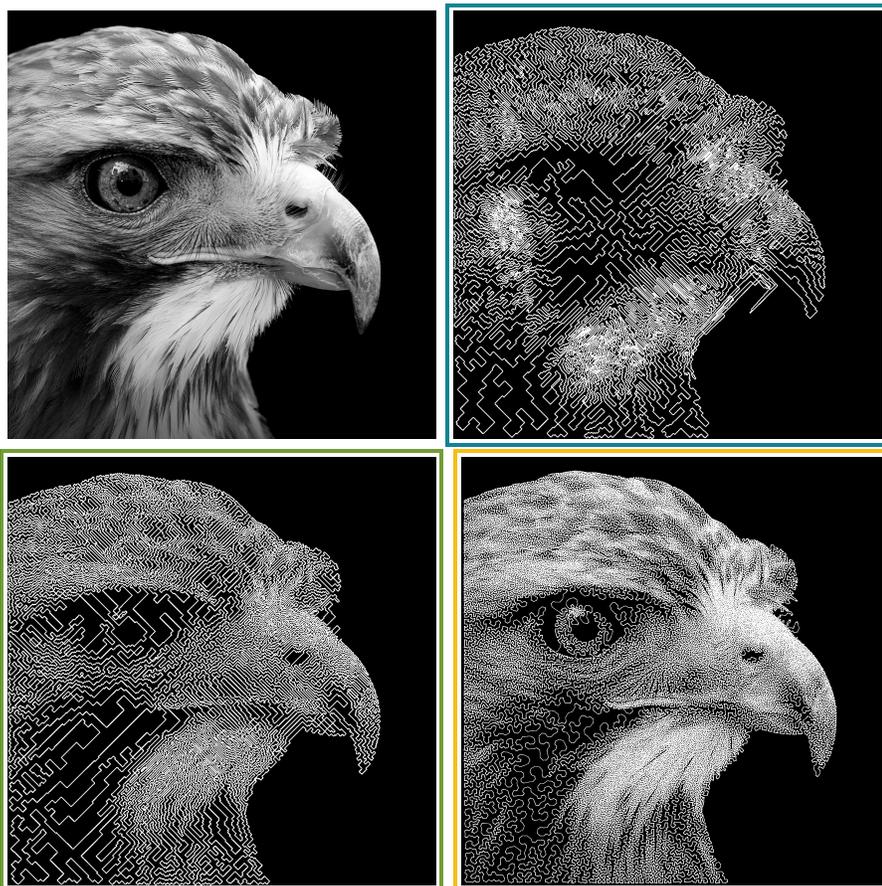


FIGURE 1.1 – En haut à gauche, l'image de référence. Les trois autres sont des approximations par courbe de l'image de référence.

L’objectif de cette thèse est double, tout d’abord cette thèse décrit une méthode permettant de calculer une distance entre une courbe et une image. Notre oeil est un instrument d’optique merveilleux qui nous permet de dire que dans la Figure 1.1 l’image encadrée de jaune “semble” la plus proche de l’image d’origine, suivie par la verte puis la bleue. Même si physiologiquement ce problème nous apparaît facile, sa formulation mathématique est non triviale.

Pour mesurer le degré de similarité entre ces images, nous pouvons penser à une méthode mathématique consistant à mesurer le carré de la différence entre l’image de référence et les images construites avec des courbes (norme L^2). Cependant cette méthode donne des résultats contre intuitifs (l’image encadrée en vert est plus proche que l’image encadrée en jaune qui a été malicieusement translaturée vers le coin supérieur droit).

Une méthode plus intéressante est de flouter l’image de référence ainsi que l’image représentée par une courbe puis de calculer la même différence L^2 . Une façon primitive de se convaincre de l’intérêt de cette méthode est de regarder les quatre images en fronçant les yeux et d’observer une différence moins prononcée entre l’image d’origine et les images représentées par des courbes¹. Moralement, il apparaît cavalier de comparer une courbe où l’information est très localisée avec une image continue. Ajouter un flou à la courbe va rendre sa représentation continue et ainsi nous permettre de mieux pouvoir comparer deux objets qui vont être de même nature. Cette distance est étudiée mathématiquement et porte le nom de distance de convolution [118, 26, 61, 127, 66].

Cette thèse traite d’un autre type de distance : la distance de transport, imaginons cette fois que les images soient représentées avec du sable à la manière de l’artiste ukrainienne *Kseniya Simonova*². La distance de transport entre deux images a une interprétation très physique, il s’agit de l’énergie minimale que l’on doit fournir pour déplacer le sable de la première image vers la seconde image. Cette énergie définit une distance entre deux images, c’est la distance de transport optimal.

Le problème du calcul de transport optimal pour des objets similaires a déjà été étudié par le passé. Soit d’un point de vue théorique [92] ou d’un point de vue plus applicatif mais sans garanties théoriques pour l’infogra-

1. Vous pouvez retrouver les images que vous devriez obtenir en Figure 1.2 plus bas.

2.  :<https://youtu.be/NaqZ9Eo5psA>

phie [72, 70]. Cette thèse est à l'entre-deux dans la mesure où les algorithmes proposés permettent la résolution numérique du transport optimal. Aussi une attention particulière est donnée à la justification théorique des méthodes proposées.

Malheureusement, le calcul exact du transport optimal entre une courbe continue et image reste numériquement hors de portée. De façon à pouvoir le rendre possible nous proposons d'approcher la courbe : soit par un ensemble de points soit par un ensemble de segments. De plus, nous apportons des résultats de consistance pour les approximations par points et par segments proposées.

Enfin le second objectif est de développer des algorithmes permettant d'obtenir les images présentées ci-dessus. Nous proposons des méthodes itératives pour optimiser la position de la mesure structurée (courbe discrétisée) de façon à réduire sa distance par rapport à une mesure cible donnée. Ces méthodes nous ont permis de réaliser une collection d'applications numériques montrant la polyvalence de notre méthode.



FIGURE 1.2 – Image originale et images représentées par des courbes floutées par une gaussienne de variance 13. Toutes les images présentées dans cette figure sont de résolution de 1024×1024 .

1.1 Distance de transport optimal

Le transport optimal est une discipline avec de nombreuses applications, notamment en économie [54] pour modéliser le marché de recherche d'emploi ou déterminer les prix de l'immobilier. Le transport optimal peut être aussi utilisé dans le domaine de l'affectation de ressources, par exemple pour le placement optimal de zones de collecte par rapport à une population [71]. Ces applications couvrent aussi la planification optimale [129]. Plus récemment, des nouvelles applications sont proposées en infographie pour comparer et mesurer des distances entre des images [93, 94, 18] ou simuler des écoulements de fluides [35]. Le transport optimal peut être aussi utilisé pour assister la conception d'instrument d'optique [97], mais aussi dans le domaine mathématique des équations aux dérivées partielles où il permet la discrétisation d'équations aux dérivées partielles pour permettre leur résolution numérique [11]. Le transport optimal peut être aussi utilisé en mécanique des fluides où il permet la résolution numérique des équations d'Euler incompressible [55]. Plus récemment le transport optimal a connu un nouvel essor dans le machine learning avec l'introduction de nouveaux algorithmes très rapides [32, 10, 105, 119] permettant le calcul de distances de transport en grande dimension. Ces nouvelles applications sont nombreuses et énumérées dans le papier [81].

Dans la suite, (X, \mathcal{A}_X) et (Y, \mathcal{A}_Y) sont deux espaces mesurables polonais, on note $\mathbb{P}(X)$ l'ensemble des mesures de probabilité sur X . Étant donné \mathcal{M} , un sous-ensemble paramétrisé de $\mathbb{P}(X)$ et une mesure cible $\mu \in \mathbb{P}(Y)$, l'objectif de cette thèse est de trouver la mesure de probabilité $\pi \in \mathcal{M}$ qui soit la plus proche possible de μ au sens de la distance de transport.

La distance de transport ou distance Wasserstein a été introduite par le mathématicien Français Gaspard Monge au XVIII^{ème} siècle. On rappelle qu'une application mesurable $T : X \rightarrow Y$ induit une notion de transport de mesure $T_{\#} : \mathbb{P}(X) \rightarrow \mathbb{P}(Y)$ (mesure image) définie par :

$$T_{\#}\pi = \nu \iff \forall B \in \mathcal{A}_Y, \quad \nu(B) = \pi(T^{-1}(B)).$$

La version de Monge du transport optimal associé à une fonction coût de transport $c : X \times Y \rightarrow \mathbb{R}^+ \cup \{+\infty\}$ revient à chercher un plan de transport T minimisant la fonctionnelle suivante :

$$\inf_{T_{\#}\pi = \mu} \int_X c(x, T(x)) d\mu(x), \quad (1.1)$$

Des hypothèses supplémentaires sur les mesures π et μ sont nécessaires pour garantir l'existence d'un plan de transport T . En effet, le plan de transport T déplace la masse d'un atome $\delta_{\mathbf{x}}$ vers $\delta_{T(\mathbf{x})}$ et ne peut scinder cette masse. Par exemple, si les deux mesures sont atomiques avec les mêmes masses sur tous les atomes et que le nombre d'atomes de la mesure π n'est pas un multiple du nombre d'atomes de la mesure μ alors il n'y a pas d'existence d'un plan de transport T satisfaisant $T_{\#}\pi = \mu$. Enfin, la formulation de Monge n'est pas symétrique, cependant elle permet de se familiariser avec le caractère "physique" du transport optimal où l'on cherche à déplacer la masse d'une mesure vers une autre de façon optimale.

Face à ces problèmes, une version relaxée du transport optimal a été développée par Kantorovitch [76, 77] dans les années 1940. Elle est basée sur la notion de couplage optimal. On rappelle qu'un couplage entre deux mesures $\pi \in \mathcal{P}(X)$ et $\mu \in \mathcal{P}(Y)$ est une mesure de probabilité sur l'espace produit $X \times Y$ dont les marginales respectives sont π et μ . En d'autres termes, en notant $\Gamma(\pi, \mu)$ l'ensemble des couplages entre π et μ on a :

$$\gamma \in \Gamma(\pi, \mu) \Leftrightarrow \begin{cases} \int_X \phi(x) d\pi(x) = \int_{X \times Y} \phi(x) d\gamma(y, x) & \forall \phi \in L^1(\pi) \\ \text{et} \\ \int_Y \psi(y) d\mu(y) = \int_{X \times Y} \psi(y) d\gamma(y, x) & \forall \psi \in L^1(\mu) \end{cases} . \quad (1.2)$$

La généralisation du problème de Monge (1.1), nommée formulation de Monge Kantorovitch, ne consiste non plus à minimiser les coûts de transport c sur l'ensemble des plans de transport T mais sur l'ensemble des couplages $\Gamma(\pi, \mu)$:

$$\inf_{\gamma \in \Gamma(\pi, \mu)} \int_{X \times Y} c(x, y) d\gamma, \quad (1.3)$$

L'ensemble des couplages $\Gamma(\pi, \mu)$ est non vide. En effet, il contient *a minima* $\pi \otimes \mu$. De plus si le coût de transport c est à valeurs dans $[0, +\infty]$ et est semi-continu inférieurement alors le problème (1.3) admet une solution (Théorème 1.7 de [116]).

La formulation (1.3) peut être réécrite sous sa version duale. En interprétant $\phi \in L^1(\pi)$ et $\psi \in L^1(\mu)$ comme des multiplicateurs de Lagrange de la contrainte $\gamma \in \Gamma(\pi, \mu)$, le problème primal (1.3) est minoré par le problème dual suivant en optimisant en ϕ :

$$\sup_{\phi(x) + \psi(y) \leq c(x, y)} \int_X \phi d\pi + \int_Y \psi d\mu \quad (1.4)$$

1.2. CALCUL NUMÉRIQUE DE LA DISTANCE DE TRANSPORT OPTIMAL 7

Sous les mêmes hypothèses pour le coût de transport c , le saut de dualité est nul et il y a égalité entre le problème primal (1.3) et le problème dual (1.4) (Théorème 1.42 de [116]). De plus, si c est uniformément continu et borné sur $X \times Y$, le problème (1.4) admet une solution finie donnée par le couple (ϕ, ϕ^c) solution du problème de maximisation suivant :

$$\sup_{\phi \in \Phi} \int_X \phi(x) d\pi(x) + \int_Y \phi^c(y) d\mu(y), \quad (1.5)$$

où ϕ^c est donnée par la formule suivante :

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

et Φ est l'ensemble des fonctions c -concave :

$$\Phi = \left\{ \phi \in L^1(X), \phi(x) = \inf_{y \in Y} c(x, y) - \phi^c(y) \right\}$$

Dans cette thèse, la distance de transport optimal étudiée est la distance de 2-Wasserstein (\mathcal{W}_2) où le coût de transport considéré est la norme euclidienne au carré $c(x, y) = \|x - y\|^2$. Cette fonction coût vérifie toutes les hypothèses de continuité. Aussi, les espaces X et Y sont supposés compacts, ce qui garantit l'égalité entre toutes les formulations (1.3), (1.4) et (1.5).

1.2 Calcul numérique de la distance de Transport optimal

Il existe trois grandes familles de méthodes permettant le calcul numérique de la distance de transport optimal. Ces méthodes sont catégorisées selon la nature de la paire de mesures (π, μ) considérée.

1.2.1 Transport optimal discret

Pour la première méthode, on considère des mesures π et μ atomiques :

$$\pi = \sum_{i=1}^n a_i \delta_{\mathbf{x}_i} \quad \text{et} \quad \mu = \sum_{j=1}^m b_j \delta_{\mathbf{y}_j}$$

Le problème de Kantorovitch se réécrit alors :

$$\begin{aligned} \min_{\gamma \in \mathbb{R}^{n \times m}} \sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} C_{ij} & \quad (1.6) \\ \text{s.c. } \gamma_{ij} \geq 0 \quad \text{et } \forall i, \quad \sum_{j=1}^m \gamma_{ij} = a_i \quad \text{et } \forall j, \quad \sum_{i=1}^n \gamma_{ij} = b_j, \end{aligned}$$

avec C la matrice des coûts de transport donnée par $C_{ij} = c(\mathbf{x}_i, \mathbf{y}_j)$ et où les deux contraintes linéaires sont les contraintes de marginales (1.2). Comme les deux mesures sont discrètes, alors le couplage γ est de la forme $\gamma = \sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} \delta_{\mathbf{x}_i} \delta_{\mathbf{y}_j}$ de sorte que l'intégrale de la formulation (1.3) se réécrit comme une somme finie.

Le problème (1.6) est un problème de programmation linéaire sous contraintes linéaires pour lequel il existe différents algorithmes tels que les solveurs linéaires **CPLEX** ou **Lemon**. Toutefois, le nombre de variables d'optimisation est nm ce qui rend cette méthode prohibitive pour un nombre d'atomes important.

Dans le cas particulier où les deux mesures sont constituées du même nombre d'atomes $m = n$ de masse uniforme $a_i = b_i = 1/n$, l'algorithme Hongrois permet de résoudre (1.6) en temps polynomial $\mathcal{O}(n^3)$, voir [82].

Pour des mesures discrètes quelconques l'algorithme d'enchère (*auction algorithm*) [15, 14], visant à améliorer de façon itérative une des variables duales du problème (1.6) permet lui aussi une résolution en temps polynomial $\mathcal{O}((n+m)^3 \log(\min(m, n) \|C\|_\infty))$ où $\|C\|_\infty = \max_{ij} |C_{ij}|$.

Enfin, l'algorithme *Network simplex* [106] permet de résoudre ce problème en modélisant le problème de transport optimal par un graphe acyclique où les sommets représentent des atomes et où les arcs entre ces sommets représentent les échanges de masses entre les deux mesures. Cet algorithme permet lui aussi d'obtenir une complexité en temps polynomiale en $\mathcal{O}((n+m)nm \log(n+m) \log((n+m) \|C\|_\infty))$.

Ces trois méthodes permettent de résoudre exactement le problème de Kantorovitch (1.3) et ne nécessitent aucune hypothèse de régularité sur le coût de transport c . Cependant leur complexité cubique ne leur permet pas d'être utilisées de façon intensive dans les domaines émergents du machine learning ou du data mining où la taille des données (n et m) est importante.

Une manière élégante de contourner la difficulté de résolution du problème linéaire (1.6), est de régulariser avec un terme entropique. L'ajout d'une ε -régularisation "lisse" les plans de transport γ et rend le problème strictement

1.2. CALCUL NUMÉRIQUE DE LA DISTANCE DE TRANSPORT OPTIMAL 9

convexe. Le problème régularisé se réécrit alors comme la minimisation d'une distance de Kullback-Leibler entre la matrice de couplage γ et le noyaux de Gibbs $e^{-C/\varepsilon}$ [32, 10]. En passant au problème dual, le nombre de variables devient $n + m$ et la solution régularisée peut-être calculée rapidement à l'aide de l'algorithme de Sinkhorn, voir Section 3.2.3 pour plus de détails. De plus l'algorithme de Sinkhorn se prête aisément à une implémentation GPU [32, 4, 83, 58, 49]. C'est pour cette raison que cette méthode de calcul du transport est devenue très populaire, notamment dans le domaine du machine learning.

1.2.2 Transport optimal continu

Si les deux mesures sont à densité et si le coût de transport est la norme euclidienne au carré $c(x, y) = \|x - y\|^2$, le problème de Monge (1.1) peut être résolu numériquement. La distance de transport peut être calculée en cherchant un chemin de taille minimale entre les deux mesures. Ce chemin est décrit par un champ de vecteur transportant la mesure source vers la mesure cible. Dans ce cadre, le chemin et le champ de vecteur obéissent à des lois de conservation ; c'est formulation de Benanou-Brenier [9]. Ce système dynamique d'équations aux dérivées partielles peut être résolu à l'aide d'une méthode de Lagrangien augmenté. Cette méthode peut être interprétée comme un cas particulier de l'algorithme de Douglas Rachford, et d'autres types d'éclatements proximaux peuvent être considérés pour résoudre ce problème [108]. Cependant pour être résolues numériquement les équations doivent être discrétisées en temps et en espace. Toutefois cette méthode est inadaptée aux mesures qui ne sont pas à densité.

1.2.3 Transport optimal semi-discret

Si la mesure π est discrète et la mesure μ est absolument continue par rapport à la mesure de Lebesgue et de densité ρ :

$$\pi = \sum_{i=1}^n \omega_i \delta_{\mathbf{x}_i} \quad \text{et} \quad \mu = \rho(y) dy, \quad (1.7)$$

alors le caractère discret de la mesure π nous permet de réécrire la formulation (1.5) de la façon suivante :

$$\max_{\phi \in \mathbb{R}^n} \sum_{i=1}^n \phi_i \omega_i + \int_Y \min_{i \in [1, n]} (c(\mathbf{x}_i, y) - \phi_i) d\mu(y). \quad (1.8)$$

En effet, pour le calcul de la c -transformée $\phi^c(y)$ nous pouvons nous limiter au support de π . Le problème du calcul du transport optimal se réécrit comme un problème de maximisation en dimension finie (1.8) et peut être résolu numériquement.

Pour chaque i notons \mathcal{T}_i la cellule :

$$\mathcal{T}_i = \left\{ y \in Y : \forall j, \quad c(\mathbf{x}_i, y) - \phi_i \leq c(\mathbf{x}_j, y) - \phi_j \right\},$$

alors,

$$\phi^c(y) = c(\mathbf{x}_i, y) - \phi_i, \quad \forall y \in \mathcal{T}_i.$$

En supposant que $\forall j \neq i, \mu(\mathcal{T}_i \cap \mathcal{T}_j) = 0$, alors le problème dual de Kantorovitch se réécrit :

$$\max_{\phi \in \mathbb{R}^n} \sum_{i=1}^n \phi_i \omega_i + \sum_{i=1}^n \int_{\mathcal{T}_i} (c(\mathbf{x}_i, y) - \phi_i) d\mu(y). \quad (1.9)$$

La condition $\mu(\mathcal{T}_i \cap \mathcal{T}_j) = 0$ est vérifiée pour tous les coûts de transport de la distance de p -Wasserstein, $c(x, y) = \|x - y\|^p$.

Pour le cas $p = 2$, les cellules \mathcal{T}_i portent le nom de cellules de Laguerre. Elles sont notées $\mathcal{L}_i(\phi, \mathbf{x})$ et sont données par la formule suivante :

$$\mathcal{L}_i(\phi, \mathbf{x}) = \left\{ y \in Y : \forall j \neq i, \|\mathbf{x}_i - y\|^2 - \phi_i \leq \|\mathbf{x}_j - y\|^2 - \phi_j \right\}. \quad (1.10)$$

Notons que si $\phi_i = \phi_j$ pour tout i, j , le diagramme de Laguerre est un diagramme de Voronoï qui est un objet important en géométrie algorithmique, en infographie, et en recherche opérationnelle [6]. Le diagramme de Laguerre est une version à poids du diagramme de Voronoï. Les cellules de Laguerre (1.10) sont des polytopes convexes.

Le diagramme de Laguerre en dimension d peut être construit comme l'intersection d'un diagramme de Voronoï de dimension $d + 1$ et \mathbb{R}^d (voir Observation 7. [85]). La complexité en temps relative à la construction d'un diagramme de Laguerre est donnée par $\mathcal{O}(n \log n)$ pour $d = 2$ et $\mathcal{O}(n \log n + n^{\lfloor \frac{d}{2} \rfloor + 1})$ pour $d > 2$ [5].

Pour le cas $p = 1$, l'ensemble des partitions \mathcal{T}_i portent le nom de diagramme d'Apollonius [17, 47]. Chacune des cellules d'Apollonius a une forme étoilée et sa frontière est constituée d'arcs ou de surfaces hyperboliques ou de droites et de plan. La complexité en temps associé à la construction d'un tel diagramme est $\mathcal{O}(n^{\lfloor \frac{d}{2} \rfloor + 1} \log n)$.

Pour évaluer la fonction profit à maximiser (1.9), il faut intégrer la densité μ sur chacune des cellules \mathcal{T}_i . Algorithmiquement, il est cependant beaucoup plus aisé de calculer ces intégrales sur des cellules aux frontières linéaires, telles que les cellules de Laguerre (1.10). De plus, la distance de transport associée au coût quadratique $p = 2$ est liée à la minimisation d'une énergie cinétique. Ainsi les méthodes développées pour résoudre le problème de transport optimal semi-discret ont connu un essor plus important pour la norme euclidienne au carré [94, 34, 85, 19].

Dans le cadre de cette thèse nous nous sommes intéressés au calcul du transport optimal L^2 semi-discret

Les résultats préliminaires au développement numérique de ces méthodes sont dus à Aurenhammer [7]. Le premier théorème établit l'optimalité du plan de transport induit par la mosaïque de Laguerre.

Théorème 1.2.1. *Le plan de transport T donné par la mosaïque de Laguerre de centroïdes \mathbf{x} et de poids ϕ , défini presque partout par :*

$$T : \begin{cases} \mathbb{R}^d \rightarrow \mathbb{R}^d \\ x \mapsto \mathbf{x}_i, \quad \text{t.q. } x \in \mathcal{L}_i(\phi, \mathbf{x}) \end{cases} ,$$

est un plan de transport optimal au sens de Monge pour W^2 entre la mesure μ absolument continue de densité ρ et la mesure atomique $\pi = \sum_i \omega_i \delta_{\mathbf{x}_i}$ si et seulement si $\omega_i = \mu(\mathcal{L}_i(\phi, \mathbf{x}))$.

Le second théorème établit la faisabilité d'un plan de transport optimal L^2 lorsque les masses des Diracs ω_i sont prescrites.

Théorème 1.2.2. *Soit μ une mesure de probabilité absolument continue par rapport à la mesure de Lebesgue à densité non nulle ρ , et π une mesure atomique définie en (1.7) avec $\mathbf{x} \in \mathbb{R}^{n \times d}$. Alors, il existe un vecteur de poids $\phi^* \in \mathbb{R}^n$ tel que :*

$$\forall i, \mu(\mathcal{L}_i(\phi^*, \mathbf{x})) = \omega_i.$$

Numériquement la résolution du transport optimal L^2 semi-discret passe par la résolution du problème de maximisation suivant :

$$\max_{\phi \in \mathbb{R}^n} g(\phi, \mathbf{x}), \quad \text{avec } g(\phi, \mathbf{x}) = \sum_{i=1}^n \int_{\mathcal{L}_i(\phi, \mathbf{x})} (\|\mathbf{x}_i - x\|^2 - \phi_i) d\mu(x) + \sum_{i=1}^n \phi_i \omega_i \quad (1.11)$$

La fonction g est deux fois dérivable par rapport à la variable duale ϕ , ses dérivées sont données par :

$$\frac{\partial g}{\partial \phi_i} = \omega_i - \int_{\mathcal{L}_i(\phi, \mathbf{x})} d\mu(x)$$

$$\frac{\partial^2 g}{\partial \phi_i \partial \phi_j} = \begin{cases} \int_{\mathcal{L}_i(\phi, \mathbf{x}) \cap \mathcal{L}_j(\phi, \mathbf{x})} \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|} d\mu & \text{si } i \neq j \\ \sum_{j \neq i} -\frac{\partial^2 g}{\partial \phi_i \partial \phi_j} & \text{sinon} \end{cases} \quad (1.12)$$

Sous certaines hypothèses explicitées dans la Partie 4.2.1 la Hessienne est de classe $\mathcal{C}^{2,\alpha}$ [80] par rapport à la variable ϕ .

De plus, les points critiques vérifient la condition suivante :

$$\int_{\mathcal{L}_i(\phi, \mathbf{x})} d\mu(x) = \omega_i,$$

la fonction g est concave par rapport à ϕ comme un infimum de fonctions linéaires [38] ce qui peut se retrouver dans le calcul de sa Hessienne qui est à diagonale dominante avec tous ses termes diagonaux sont négatifs ou nuls. Les points critiques sont donc maxima globaux, ce qui permet de retrouver le Théorème 1.2.1.

Pour l'optimisation de ce problème de maximisation une méthode de *damped Newton* peut être mise en oeuvre, sous certaines hypothèses de régularité la convergence est globale [80, 96].

La fonctionnelle de Kantorovitch (1.11) peut-être aussi optimisée par rapport à la position des masses de Dirac \mathbf{x} . Si la mesure μ est à densité suffisamment régulière (de classe \mathcal{C}^2) alors la fonctionnelle de Kantorovitch est C^1 par rapport à \mathbf{x} et sa dérivée est donnée par :

$$\frac{\partial g}{\partial \mathbf{x}_i} = 2 \int_{\mathcal{L}_i} (\mathbf{x}_i - x) d\mu(x).$$

De plus, pour les diagrammes de Voronoï, Lévy *et al.* [89] a montré la régularité \mathcal{C}^2 presque partout de $g(0_{\mathbb{R}^n}, \mathbf{x})$ par rapport à la variable \mathbf{x} et conjecturé que la régularité de la mesure de fond pouvait être affaiblie à \mathcal{C}^0 . Enfin il a montré l'intérêt pratique des méthodes de second ordre pour le problème du CVT (centroidal Voronoi tessellation) plus rapides et plus robustes que les méthodes du premier ordre.

1.3 Génération de schéma d'échantillonnage en IRM

Ce travail est motivé par le problème d'acquisition compressé en imagerie IRM [21, 84] où l'objectif est d'échantillonner le domaine de Fourier pour reconstruire à l'aide des coefficients de Fourier une image médicale. L'IRM acquiert des échantillons le long de trajectoires suffisamment régulières, notre objectif est d'optimiser ces trajectoires. L'ensemble des trajectoires générées \mathcal{S} doit respecter des conditions cinématiques, de façon à ce que la trajectoire soit réalisable avec un IRM et que le temps d'acquisition soit limité, à savoir :

$$\mathcal{S} = \left\{ s : [0, 1] \rightarrow \mathbb{R}^d, s \in \mathcal{C}^2, \forall t \quad \|s'(t)\| \leq c_1 \quad \text{et} \quad \|s''(t)\| \leq c_2 \right\}.$$

Pour avoir la meilleure reconstruction possible, la mesure portée par la trajectoire doit *approcher* une densité cible μ non uniforme [21, 1]. Pour mesurer la qualité d'approximation de cette densité cible nous utilisons la distance de transport optimal L^2 , de sorte que le problème se réécrite :

$$\inf_{s \in \mathcal{S}} \mathcal{W}_2(s_{\#}\lambda, \mu),$$

où $s_{\#}\lambda$ est la mesure image de λ : la mesure de Lebesgue de $[0, 1]$, par l'application s .

La mesure portée par la trajectoire n'étant pas à densité, le transport continu ne peut être utilisé.

Si on suppose maintenant que la courbe est discrétisée par un ensemble de points et que la mesure cible μ est discrète, l'utilisation du transport optimal discret et de l'algorithme de Sinkhorn est légitime. Cependant, nos tests numériques ont révélé que le terme de régularisation isotrope ε devait être choisi avec précaution et suffisamment petit. Or, un paramètre de régularisation trop petit génère de grandes instabilités numériques. Nous décrivons avec plus de précisions ces effets dans le Chapitre 3.

Face à ce problème nous avons décidé dans un premier temps d'adopter le formalisme semi-discret en considérant cette fois que la mesure μ est absolument continue et que la mesure portée par la trajectoire est discrétisée par un ensemble de points. Par la suite nous appelons ce problème *curvling*. Ce problème est similaire au *bluenoise* semi-discret [34] si ce n'est que nous imposons des contraintes sur la position des masses de Dirac \mathbf{x} . Nous étudions ce problème dans le Chapitre 5.

Finalement, si la mesure μ est discrète et $s_{\#}\lambda$ est portée par une courbe, nous pouvons utiliser le transport optimal semi-discret. L'optimisation de la fonctionnelle (1.11) nécessite d'intersecter la mesure portée par la trajectoire $s_{\#}\lambda$ avec chacune des cellules de Laguerre. Toutefois, le calcul de ces intersections pour une trajectoire continue est numériquement difficile. Dans le Chapitre 6, en introduisant une discrétisation sur la courbe (linéaire par morceaux), nous explicitons une méthode rapide pour le calcul de ces intersections et du transport optimal associé à ce problème. Le calcul du transport optimal pour cette nouvelle méthode diffère de la méthode employée dans le cas du *stippling* [96], où la fonctionnelle de Kantorovitch (1.11) est de classe \mathcal{C}^2 .

1.4 Plan de la thèse

Dans cette thèse nous voulons résoudre le problème d'optimisation sur des espaces de mesures suivant :

$$\inf_{s \in \mathcal{S}} \mathcal{W}_2(s_{\#}\lambda_{([0,1])}, \mu),$$

Dans le Chapitre 3 les différentes méthodes numériques permettant de calculer une distance entre deux mesures et notamment le transport optimal L^2 sont comparées. Nous décrivons les limitations propres à chacune de ces méthodes et l'algorithmie qui leur est associée. Cette discussion motive notre choix pour le transport optimal semi-discret.

Dans le Chapitre 4, nous déterminons les conditions suffisantes rendant la fonctionnelle du transport optimal semi-discret (1.11) deux fois continûment dérivable par rapport à la variable duale mais aussi par rapport à la position des masses de Dirac. Les résultats sur la régularité \mathcal{C}^2 par rapport la variable duale développés par Mérigot et ont permis d'établir la convergence de l'algorithme de *damped Newton* [80, 96]. Aussi, certaines de nos hypothèses sont similaires aux travaux de Mérigot *et al.* [80, 96], cependant la méthode de preuve proposée dans le Chapitre 4 est plus générale et permet aussi de prouver la régularité du transport optimal par rapport à la position des masses de Dirac. L'existence d'une dérivée seconde par rapport à la position des masses de Dirac laisse entrevoir l'utilisation d'une méthode de second ordre pour l'optimisation du placement des masses de Dirac à la manière de Lévy *et al.* [89]. Dans la Partie 4.3, l'intérêt pratique des méthodes du second ordre est éprouvé pour deux problèmes particuliers que

sont le *Bluenoise* et le *Stippling* et pour différentes expériences numériques. Le résultat de ces expériences est nuancé : le nombre d'itérations globale est, comme attendu, plus faible avec un algorithme du second ordre, notamment grâce aux dernières itérations où la convergence est quadratique. Cependant, le coût d'un pas de descente peut nécessiter la résolution de plusieurs problèmes de transport optimal et est plus important que celui que l'algorithme de *Lloyd* (qui consiste simplement à déplacer la masse de Dirac sur la position du barycentre de la cellule de Laguerre qui lui est associée).

Dans le Chapitre 5, nous présentons les différentes méthodes numériques utilisées pour la résolution du transport optimal L^2 . Ces méthodes permettent d'approcher une mesure absolument continue par une mesure atomique quelconque, supportée par une trajectoire à vitesse et à accélération bornée ou constante ou par un ensemble de segments. Le calcul du transport optimal L^2 pour ces mesures est similaire à celles utilisées pour le transport optimal semi-discret [34, 85, 96] à la différence qu'elles sont rendues plus robustes pour fonctionner dans des configurations de points compliquées et des mesures de fond plus alambiquées. Nous décrivons aussi les différents projecteurs utilisés permettant de discrétiser une trajectoire avec des contraintes sur sa vitesse et son accélération par une mesure atomique. Enfin, notre algorithme robuste de calcul distance de transport L^2 est comparé avec une méthode préexistante [34], et avec une implémentation de la distance de convolution [21]. La comparaison s'établit sur la rapidité des algorithmes mais aussi sur leur complexité théorique.

Dans le Chapitre 6, la discrétisation de trajectoire s n'est plus constante par morceaux mais linéaire par morceaux et la mesure de fond est supposée atomique. De cette façon la mesure d'approximation n'est plus une mesure atomique mais par une mesure portée par une polyligne (trajectoire continue et linéaire par morceau). Toujours à l'aide de la bibliothèque de géométrie algorithmique *CGAL* [128], un nouvel algorithme de calcul de transport optimal est présenté, il permet de calculer le transport optimal entre une mesure atomique et une mesure portée par une polyligne. Malgré les récents progrès théoriques [96] prouvant la convergence de l'algorithme de *damped Newton* pour des mesures portées par des simplexes de dimension de Hausdorff inférieure à la dimension ambiante, la question des mesures portées par des courbes est vierge, et est le sujet du Chapitre 6. Nous avons baptisé ce nouveau problème transport optimal $\frac{3}{4}$ -discret. Dans ce cas précis, la fonctionnelle de Kantorovitch est moins régulière par rapport à sa variable duale ϕ : elle n'est plus que \mathcal{C}^1 . La méthode d'optimisation doit être modifiée. Nous

essayons donc de nombreuses méthodes afin d'établir celle qu'il est la plus rapide.

Dans le Chapitre 7 nous vérifions la consistance des approximations faites au cours des chapitre 5 et 6. En effet, la trajectoire s continue a été discrétisée soit par une trajectoire constante par morceaux ou par une trajectoire linéaire par morceaux. Une question naturelle est d'estimer vitesse convergence de l'approximation considérée vers la courbe continue s recherchée. Dans ce chapitre, la distance de Hausdorff entre des polygones ou des courbes atomiques dont les dérivées discrètes sont bornées avec l'ensemble des courbes à dérivées bornées est explicitée. Le résultat de cette section est intuitif, à savoir, l'approximation par une polygone converge quadratiquement vers l'ensemble des courbes continues là où l'approximation constante par morceaux ne converge que linéairement.

Enfin, dans le Chapitre 8 nous décrivons en détail la méthode d'intégration rapide, utilisée lors du Chapitre 5 pour le calcul de la distance de transport semi-discrète, permettant de convertir les intégrales sur les volumes des cellules de Laguerre en intégrales sur leur frontières via la formule de Green. Après de brefs rappels théoriques, les problèmes inhérents à la maximisation de la fonctionnelle de Kantorovitch dans le cadre du *curvling* sont décrits. Nous exposons avec précision les difficultés numériques auxquelles nous avons été confrontés et les différentes méthodes, plus ou moins concluantes, qui ont été essayées. Enfin, nous examinons les avantages de la méthode de Newton régularisée. Dernièrement, nous présentons l'algorithme multi-échelle qui nous a permis de réaliser toutes les simulations numériques pour le *curvling* et pour un grand nombre de points.

Chapter 2

Introduction

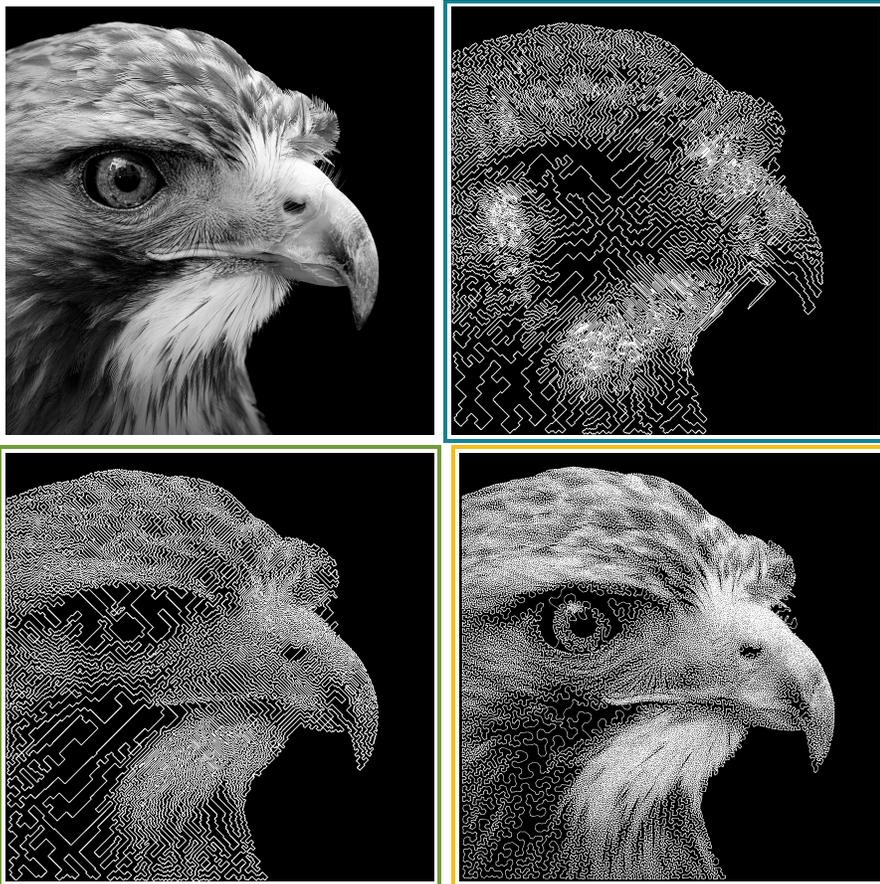


FIGURE 2.1 – The original image is top left. The other three images are curve approximations of the original one.

The aim of this thesis is two-fold. First, it describes a method that enables the computation of a distance between a curve and an image. The human eye is a wonderful optical instrument that allows us to say whether in Figure 2.1 the image framed in yellow, green or blue is closest to the original image. Although this problem seems simple to us, its mathematical formulation is non-trivial.

To measure the similarity between these images, one possible approach is to measure the square of the difference between the original image and the images constructed using curves (L^2 norm). However this method leads to counter intuitive results : the image framed in green is closer than the image framed in yellow, as the latter has been slightly shifted towards the upper right-hand corner.

A more interesting method involves blurring the reference image as well as the images constructed using curves, and then computing the same L^2 distance. A simple way of convincing ourselves of the benefit of this method is to look at the four images with slightly closed eyes and observe that the difference between the original image and the curve approaching images is then reduced¹. Morally, it seems cavalier to wish to compare a curve, for which information is very localized, to a continuous image. Blurring the curve makes its representation continuous and allows us to better compare two objects that are now of the same nature. The distance involved when using this method is known as the convolution distance and has been studied mathematically [118, 61, 26, 127, 66].

This thesis deals with another type of distance : the transport distance. Imagine this time that the images are represented by piles of sand, in the style of the Ukrainian artist *Kseniya Simonova*². The optimal transport distance between the two images has a very physical interpretation : it is the minimal energy needed in order to move the sand from the original image to the second image. This energy defines a distance between two images, it is the optimal transport distance.

The optimal transport computation problem has already been studied for similar objects, either with a very theoretical point of view [92] or with a very applied point of view in computer graphics [72, 70]. This thesis is in between the two previous topics as it provides numerical algorithms to

1. Figure 2.2 show the images you should see when using this method.

2.  :<https://youtu.be/NaqZ9Eo5psA>

solve the optimal transport problem along with a strong emphasis on their theoretical foundations.

Unfortunately, the exact calculation of optimal transport between a continuous curve and an image remains numerically inaccessible. As a workaround, we attempt to approach the curve either by a set of points or by a set of line segments. We justify these methods by providing consistency results for both point and line segment approximations.

Finally, the second aim is to develop algorithms that generate the curve approximating images presented above. We propose iterative methods to optimize the position of the structured measure (discretized curve) in order to reduce its distance from the given target measure. Our methods have allowed us to carry out many numerical applications, thus highlighting the versatility of our method.



FIGURE 2.2 – Reference image and curve-approaching images, images are blurred performing a convolution with a gaussian kernel of standard deviation 13. The displayed images are of size 1024×1024

2.1 Optimal transport distance

Optimal transport is a discipline whose possible applications are numerous. Indeed, it is used in economy [54] to model the job market or compute real estate prices. Another field of application is resource allocation where it can be used to place collection zones optimally with respect to people's density in a city [133]. These applications also cover optimal planning [129]. More recently, new applications are proposed in infography to compare and measure distances between images [93, 94, 18] or simulate fluid flow [35]. Optimal transport can also be used to help with the design of optical instruments [97], but can additionally be used to discretize partial differential equations and thus enable their numerical resolution [11]. Another field of application is fluid mechanics where it enables the numerical resolution of incompressible Euler equations [55]. More recently, optimal transport has known a new boom in machine learning with the introduction of extremely quick algorithms [32, 10, 105, 119] allowing the computation of the transport distances in high dimension. These new applications are numerous and are listed in the following paper [81].

In the following (X, \mathcal{A}_X) et (Y, \mathcal{A}_Y) are two Polish measurable spaces, we denote $\mathbb{P}(X)$ the set of probability measures on X . Given \mathcal{M} , a parametrized subset of $\mathbb{P}(X)$ and a target measure $\mu \in \mathbb{P}(Y)$, the aim of this thesis is to find the probability measure $\pi \in \mathcal{M}$ that is the closest to μ for the transport distance. The transport distance, or Wasserstein distance, was introduced by the French mathematician Gaspard Monge in the XVIIIth century. We recall that a measurable application $T : X \rightarrow Y$ prompts the notion of push-forward measure $T_{\#} : \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$ defined by :

$$T_{\#}\pi = \mu \iff \text{for any } \Omega \in \mathcal{B}(Y), \quad \mu(\Omega) = \pi(T^{-1}(\Omega)).$$

Computing the optimal transport formulated by Monge for a transport cost function $c : X \times Y \rightarrow \mathbb{R}^+ \cup \{+\infty\}$ equates to finding the transport plan T under the constraint $T_{\#}\mu = \pi$ that minimizes the following functional :

$$\inf_{T_{\#}\pi=\mu} \int_X c(x, T(x))d\mu(x). \quad (1.1)$$

We need extra hypotheses on measures π and μ to guarantee the existence of a transport plan T . Indeed, the transport plan T shifts the mass of an atom δ_x to $\delta_{T(x)}$ and cannot split this mass. For example, if the two measures are

atomic with the same mass on each atom, and if the number of atoms of the measure π is not a multiple of the number of atoms of the measure μ , then there does not exist a transport plan T satisfying $T_{\#}\pi = \mu$. Also, Monge's formulation is not symmetrical however it allows us to familiarize ourselves with the "physical" aspect of optimal transport where we wish to optimally transport the mass of one measure to another.

Consequently a relaxed formulation of optimal transport was developed by Kantorovitch [76, 77] in the 1940s. This formulation is based on the concept of optimal coupling. In probability theory, a coupling of two measures $\pi \in \mathcal{P}(X)$ and $\mu \in \mathcal{P}(Y)$ is a probability measure on the product space $X \times Y$ whose respective marginals are π and μ . In other terms, denoting $\Gamma(\pi, \mu)$ the set of couplings between π and μ we have :

$$\gamma \in \Gamma(\pi, \mu) \Leftrightarrow \begin{cases} \int_X \phi(x) d\pi(x) = \int_{X \times Y} \phi(x) d\gamma(y, x) & \forall \phi \in L^1(\pi) \\ \text{and} \\ \int_Y \psi(y) d\mu(y) = \int_{X \times Y} \psi(y) d\gamma(y, x) & \forall \psi \in L^1(\mu) \end{cases} . \quad (1.2)$$

The generalization of Monge's problem (1.1), named Monge Kantorovitch formulation, no longer involves minimizing transport costs c on the set of transport plans T but on the set of couplings $\Gamma(\pi, \mu)$:

$$\inf_{\gamma \in \Gamma(\pi, \mu)} \int_{X \times Y} c(x, y) d\gamma, \quad (1.3)$$

The set of couplings $\Gamma(\pi, \mu)$ is non empty. Indeed, it contains at least $\pi \otimes \mu$. Furthermore if c is valued in $[0, +\infty]$ and is lower semi-continuous then Problem (1.3) has a solution (Theorem 1.7 of [116]).

Formulation (1.3) can be rewritten under its dual form. Interpreting $\phi \in L^1(\pi)$ and $\psi \in L^1(\mu)$ as Lagrange multipliers of the constraint $\Gamma(\pi, \mu)$, the primal problem (1.3) is lower bounded by the following dual problem :

$$\sup_{\phi(x) + \psi(y) \leq c(x, y)} \int_X \phi d\pi + \int_Y \psi d\mu \quad (1.4)$$

The duality gap is zero and there is equality between the primal problem (1.3) and the dual problem (1.4) (Theorem 1.42 of [116]). Furthermore, if c is uniformly continuous and bounded on $X \times Y$, Problem (1.4) has a finite solution given by the couple (ϕ, ϕ^c) solution to the following maximization problem :

$$\sup_{\phi \in \Phi} \int_X \phi d\pi + \int_Y \phi^c, \quad (1.5)$$

where ϕ^c is given by the following formula :

$$\phi^c(y) = \inf_{a \in X} c(a, y) - \phi(a),$$

and where Φ is the set of c -concave functions :

$$\Phi = \left\{ \phi \in L^1(X), \phi(x) = \inf_{y \in Y} c(x, y) - \phi^c(y) \right\}.$$

In this thesis, the optimal transport distance studied is the 2-Wasserstein distance (\mathcal{W}_2), where the transport cost is $c(x, y) = \|x - y\|^2$ which satisfies all of the continuity hypotheses. Furthermore we suppose that the spaces X and Y are compacts, this guarantees equality of the formulations (1.3), (1.4) and (1.5).

2.2 Numerical computation of the optimal transport distance

To compute the optimal transport distance, three big families of numerical methods exist. These families are named according to the nature of the pairs of measures (π, μ) that they can be applied to.

2.2.1 Discrete optimal transport

For the first method we consider π and μ two atomic measures :

$$\pi = \sum_{i=1}^n a_i \delta_{\mathbf{x}_i} \quad \text{and} \quad \mu = \sum_{i=1}^m b_i \delta_{\mathbf{y}_i}$$

In this case, the Kantorovitch problem rewrites as :

$$\begin{aligned} \min_{\gamma \in \mathbb{R}^{n \times m}} \sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} C_{ij} & \quad (1.6) \\ \text{s.t. } \gamma_{ij} \geq 0 \quad \text{and } \forall i, \sum_{j=1}^m \gamma_{ij} = a_i \quad \text{and } \forall j, \sum_{i=1}^n \gamma_{ij} = b_j, & \end{aligned}$$

where C is the matrix of transport costs given by $C_{ij} = c(\mathbf{x}_i, \mathbf{y}_j)$ and where the two linear constraints are the marginal constraints (1.2). The

discrete nature of the two measures yields a discrete coupling of the form $\gamma = \sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} \delta_{\mathbf{x}_i} \delta_{\mathbf{y}_j}$ such that the integral of formulation (1.3) rewrites as a finite sum.

Problem (1.6) is a linear program with linear constraints for which several algorithms exist such as the linear solvers **CPLEX** and **Lemon**. However the number of optimization variables is nm , which can make the complexity blows off when using the method with a high number of atoms.

In the particular case where both measures are made up of the same number of atoms $m = n$ of uniform mass $a_i = b_i = 1/n$, the Hungarian algorithm solves Problem (1.6) with a polynomial time complexity $\mathcal{O}(n^3)$, see [82].

For general discrete measures, the *auction algorithm* [15, 14], that aims to iteratively improve one of the dual variables of the problem (1.6) also has a polynomial time complexity $\mathcal{O}((n+m)^3 \log(\min(m,n) \|C\|_\infty))$ where $\|C\|_\infty = \max_{ij} |C_{ij}|$.

Finally, the *Network simplex* algorithm [106] solves the problem by modeling the optimal transport problem as an acyclical graph where the vertices represent atoms and the edges represent the mass exchanges between the two measures. This algorithm has a polynomial time complexity of $\mathcal{O}((n+m)nm \log(n+m) \log((n+m) \|C\|_\infty))$.

These three methods allow us to solve the Kantorovitch problem (1.3) and do not require any hypotheses on the transport cost c . However their cubic complexity means that they cannot be used intensively in the emerging fields of machine learning or data mining where the data sizes (n and m) are big.

An elegant way of bypassing the difficulties linked to solving linear problem (1.6) is to regularize with an entropy term. This ε -regularization term "smooths" the coupling γ and makes the problem strictly convex. The regularized problem then rewrites as the minimization of a Kullback-Leibler distance between the coupling matrix γ and the Gibbs kernel $e^{-C/\varepsilon}$ [32, 10]. Writing the dual problem, the number of variables becomes $n+m$ and the regularized solution can be computed quickly using the Sinkhorn algorithm, see Section 3.2.3 for more details. Furthermore the Sinkhorn algorithm can lend itself to a GPU implementation [32, 4, 83, 58, 49]. This is why this regularized method has become very popular, especially in the field of machine learning.

2.2.2 Continuous optimal transport

If both measures are given by a probability density function and if the transport cost is the square of the euclidean norm $c(x, y) = \|x - y\|^2$, the Monge problem (1.1) can be solved numerically. The transport distance can be computed by searching for the shortest path between the two measures. This path is described as a vector field that advects the source measure to the target measure. In this context, the path and the vector field obey to some conservation laws; it is the Benanou-Brenier formulation [9]. This dynamic system of partial differential equations can be solved using an augmented Lagrangian method. This method can be interpreted as a particular case of the Douglas Rachford algorithm, and other types of proximal splitting can be considered to solve this problem [108]. However to be solved numerically, the equations must be discretized in time and space. Nevertheless, this method is available only if both of the measures have a density.

2.2.3 Semi-discrete optimal transport

If the measure π is discrete and the measure μ is absolutely continuous with respect to the Lebesgue measure, with density ρ :

$$\pi = \sum_{i=1}^n \omega_i \delta_{\mathbf{x}_i} \quad \text{and} \quad \mu = \rho(y) dy, \quad (1.7)$$

then the discrete nature of measure π allows us to rewrite formulation (1.5) in the following way :

$$\max_{\phi \in \mathbb{R}^n} \sum_{i=1}^n \phi_i \omega_i + \int_Y \min_{i \in [1, n]} (c(\mathbf{x}_i, y) - \phi_i) d\mu(y). \quad (1.8)$$

Indeed, to compute the c-transform $\phi^c(y)$ we can limit ourselves to the support of π . The new formulation (1.8), which is a finite dimensional maximization problem, can now be solved numerically.

For each i let us denote as \mathcal{T}_i the cell :

$$\mathcal{T}_i = \left\{ y \in Y : \quad \forall j, \quad c(\mathbf{x}_i, y) - \phi_i \leq c(\mathbf{x}_j, y) - \phi_j \right\},$$

then,

$$\phi^c(y) = c(\mathbf{x}_i, y) - \phi_i, \quad \forall y \in \mathcal{T}_i.$$

Supposing that $\forall j \neq i, \mu(\mathcal{T}_i \cap \mathcal{T}_j) = 0$, the Kantorovitch dual problem rewrites as :

$$\max_{\phi \in \mathbb{R}^n} \sum_{i=1}^n \phi_i \omega_i + \sum_{i=1}^n \int_{\mathcal{T}_i} (c(\mathbf{x}_i, y) - \phi_i) d\mu(y). \quad (1.9)$$

The condition $\mu(\mathcal{T}_i \cap \mathcal{T}_j)$ is verified for all of the transport costs of the p -Wasserstein distance, $c(x, y) = \|x - y\|^p$.

When $p = 2$, the cells \mathcal{T}_i are called Laguerre cells. They are denoted $\mathcal{L}_i(\phi, \mathbf{x})$ and are given by :

$$\mathcal{L}_i(\phi, \mathbf{x}) = \left\{ y \in Y : \forall j \neq i, \|\mathbf{x}_i - y\|^2 - \phi_i \leq \|\mathbf{x}_j - y\|^2 - \phi_j \right\}. \quad (1.10)$$

Note that if $\phi_i = \phi_j$ for all i and j , the Laguerre diagram is a Voronoi diagram which is an important object in algorithmic geometry, in computer graphics and in operational research [6]. The Laguerre diagram is a weighted version of the Voronoi diagram. The Laguerre cells (1.10) are convex polytopes.

The Laguerre diagram in dimension d can be constructed as the intersection of a Voronoi diagram of dimension $d+1$ and \mathbb{R}^d (see Observation 7. [85]). The time complexity relative to the construction of a Laguerre diagram is $\mathcal{O}(n \log n)$ for $d = 2$ and $\mathcal{O}(n \log n + n^{\lfloor \frac{d}{2} \rfloor + 1})$ for $d > 2$ [5].

In the case where $p = 1$, the set of partitions \mathcal{T}_i is called an Apollonius diagram [17, 47]. Each of the Apollonius cells is star shaped and its boundary is made up of edges or hyperbolic surfaces or lines and planes. The time complexity for the construction of such a diagram is $\mathcal{O}(n^{\lfloor \frac{d}{2} \rfloor + 1} \log n)$.

In order to evaluate the profit function that we wish to maximize (1.9), we have to integrate the density μ on each of the cells \mathcal{T}_i . Algorithmically, it is however easier to integrate the density μ on cells whose boundaries are linear. Furthermore, the transport distance for the quadratic cost $p = 2$ is linked to the minimization of a kinetic energy. Thus the methods developed to solve the semi-discrete optimal transport problem have evolved much more for the square of the euclidean norm $p = 2$ [94, 34, 85, 19].

In the context of this thesis we focused our interest on the calculation of L^2 semi-discrete optimal transport.

The preliminary results for the numerical development of this method are owed to Aurenhammer [7]. The first theorem establishes the optimality of the transport plan induced by the Laguerre tessellation.

Theorem 2.2.1. *The transport plan T given by the Laguerre tessellation with centroids \mathbf{x} and weights ϕ , defined almost everywhere as :*

$$T : \left\{ \begin{array}{l} \mathbb{R}^d \rightarrow \mathbb{R}^d \\ x \mapsto \mathbf{x}_i, \quad \text{s.t. } x \in \mathcal{L}_i(\phi, \mathbf{x}) \end{array} \right. ,$$

is an optimal transport plan (as defined by Monge for W^2) between the absolutely continuous measure μ of density ρ and the atomic measure $\pi = \sum_i \omega_i \delta_{\mathbf{x}_i}$ if and only if $\omega_i = \mu(\mathcal{L}_i(\phi, \mathbf{x}))$.

The second theorem establishes the feasibility of an L^2 optimal transport plan when the Dirac masses ω_i are prescribed.

Theorem 2.2.2. *Let μ be a probability measure, absolutely continuous with respect to the Lebesgue measure with non zero density ρ , and π an atomic measure defined in (1.7) with $\mathbf{x} \in \mathbb{R}^{n \times d}$. Then, there exists a weight vector $\phi^* \in \mathbb{R}^n$ such that :*

$$\forall i, \mu(\mathcal{L}_i(\phi^*, \mathbf{x})) = \omega_i.$$

The numerical resolution of the L^2 semi-discrete optimal transport involves solving the following maximization problem :

$$\max_{\phi \in \mathbb{R}^n} g(\phi, \mathbf{x}), \quad \text{with } g(\phi, \mathbf{x}) = \sum_{i=1}^n \int_{\mathcal{L}_i(\phi, \mathbf{x})} (\|\mathbf{x}_i - x\|^2 - \phi_i) d\mu(x) + \sum_{i=1}^n \phi_i \omega_i \quad (1.11)$$

The function g is twice differentiable with respect to the dual variable ϕ and its derivatives are given by :

$$\frac{\partial g}{\partial \phi_i} = \omega_i - \int_{\mathcal{L}_i(\phi, \mathbf{x})} d\mu(x)$$

$$\frac{\partial^2 g}{\partial \phi_i \partial \phi_j} = \begin{cases} \int_{\mathcal{L}_i(\phi, \mathbf{x}) \cap \mathcal{L}_j(\phi, \mathbf{x})} \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|} d\mu & \text{if } i \neq j \\ \sum_{j \neq i} -\frac{\partial^2 g}{\partial \phi_i \partial \phi_j} & \text{otherwise} \end{cases} \quad (1.12)$$

Under certain hypotheses stated in Part 4.2.1, the Hessian is of class $\mathcal{C}^{2,\alpha}$ [80] with respect to the variable ϕ .

Furthermore, the critical points verify the following condition :

$$\int_{\mathcal{L}_i(\phi, \mathbf{x})} d\mu(x) = \omega_i,$$

the function g is concave with respect to ϕ as an infimum of linear functions [38] (which can be found in the computation of its Hessian which has a dominating diagonal with only negative or zero diagonal terms). The critical points are thus global maxima, which is in line with Theorem 2.2.1.

To optimize this maximization problem, a *damped Newton* method can be implemented, under certain regularity hypotheses the convergence is global [80, 96].

The Kantorovitch functional (1.11) can also be optimized with respect to the position of the Dirac masses \mathbf{x} . If the density of measure μ is regular enough (of class \mathcal{C}^2) then the Kantorovitch functional is C^1 with respect to \mathbf{x} and its differential is given by :

$$\frac{\partial g}{\partial \mathbf{x}_i} = 2 \int_{\mathcal{L}_i} (\mathbf{x}_i - x) d\mu(x).$$

Furthermore, regarding Voronoï diagrams, Lévy *et al.* [89] showed the almost-everywhere \mathcal{C}^2 regularity of $g(0_{\mathbb{R}^n}, \mathbf{x})$ with respect to the variable \mathbf{x} and conjectured that the regularity of the base measure could be weakened to \mathcal{C}^0 . Finally he pointed out the practical interest of second order methods for the CVT (centroidal Voronoi tessellation) problem : these are faster and more robust than first order methods.

2.3 MRI sampling scheme generation

This work is motivated by the compressive sampling problem in MRI imaging [21, 84] where the aim is to sample in the Fourier domain to then reconstruct a medical image. The MRI machine collects samples along sufficiently continuous trajectories, our objective is to optimize these trajectories. However, the set of generated trajectories \mathcal{S} must satisfy kinetic conditions, in such a way that the trajectory can be followed by a MRI and also such that the acquisition time is limited, meaning that :

$$\mathcal{S} = \left\{ s : [0, 1] \rightarrow \mathcal{R}^d, s \in \mathcal{C}^2, \forall t \quad \|s'(t)\| \leq c_1 \quad \text{et} \quad \|s''(t)\| \leq c_2 \right\}.$$

In order to obtain the best possible reconstruction, the trajectory must approach a non uniform target density μ [21, 1]. To measure the quality of approximation we use the optimal transport distance L^2 in a such a way that the problem rewrites as :

$$\inf_{s \in \mathcal{S}} \mathcal{W}_2(s_{\#}\lambda, \mu),$$

where $s_{\#}\lambda$ is the image measure of the Lebesgue measure $[0, 1]$ by the application s .

As the measure carried by the trajectory does not have a density, continuous transport cannot be used.

If we now suppose that the curve is discretized by a set of points and that the target measure μ is discrete, the use of discrete optimal transport and of the Sinkhorn algorithm is justified. However, our numerical tests have shown that the isotropic regularization term ϵ has to be chosen carefully and be small enough, and when the regularization term is too small lots of numerical instability occurs. We describe these effects in more detail in Chapter 3.

To deal with this problem, we first use the semi-discrete formalism and consider this time that the measure μ is absolutely continuous and that the measure carried by the trajectory is discretized by a set of points. We call this problem *curving*. This problem is similar to semi-discrete *bluenoise* [34] up to the fact that we enforce constraints on the positions of the Dirac masses \mathbf{x} . We study this problem in Chapter 5.

Finally if the measure μ is discrete and $s_{\#}\lambda$ is carried by a curve, we may use semi-discrete optimal transport. To be able to perform the optimization, the functional (1.11) requires us to be able to intersect the measure carried by the trajectory $s_{\#}\lambda$ with each of the Laguerre cells. However computing these intersections for a continuous trajectory is numerically difficult. In Chapter 6, we provide a piecewise linear discretization of the curve and we explicit a fast method for the computation of these intersections and of the related optimal transport. The computation of the optimal transport for this new method differs from the method employed in the *stippling* case [96], where the Kantorovitch functional (1.11) belongs to the \mathcal{C}^2 class.

2.4 Thesis plan

In this thesis, we wish to solve the optimization problem on the following set of measures :

$$\inf_{s \in \mathcal{S}} \mathcal{W}_2(s_{\#\lambda}([0,1]), \mu),$$

In Chapter 3 we perform a comparison of the different numerical methods allowing the computation of distances between measures, with a focus on the L^2 optimal transport distance. We describe both the algorithmics and the limitations of each method. This discussion motivates our choice for the semi-discrete optimal transport.

In Chapter 4, we determine sufficient conditions that make the semi-discrete optimal transport functional two times continuously differentiable with respect to the dual variable but also with respect to the position of the Dirac masses. The results on the \mathcal{C}^2 regularity with respect to the dual variable were developed by Mérigot and led to the proof of convergence of the *damped Newton* algorithm [80, 96]. Also, some of our hypotheses are similar to those of the works of Merigot *et al.* [80, 96], but the method of proof given in Chapter 4 is more general and also allows us to prove the regularity of the optimal transport with respect to the position of the Dirac masses.

The existence of a second order derivative with respect to the Dirac masses suggests the use of a second order method for the optimization of the Diracs' positions in the style of Levy *et al.* [89]. In Part 4.3, the practical interest of second order methods is proven for two specific problems that are Bluenoise and Stippling, and for different numerical experiments. The result of these experiments is mixed : the global number of iterations is, as expected, lower with a second order method, notably thanks to the last iterations for which the convergence rate is quadratic. However, the cost of a descent step can require the resolution of several optimal transport problems and is higher than that of the *Lloyd* algorithm (which simply moves the Dirac mass onto the barycenter of its related Laguerre cell).

In Chapter 5, we present the different numerical methods used to solve L^2 optimal transport. These methods allow us to approach an absolutely continuous measure by an atomic measure, borne by a trajectory whose speed and acceleration are bounded or constant, or a set of line segments. The methods to compute L^2 optimal transport are similar to those used for semi-discrete optimal transport [34, 85, 96] with the difference that they are made more robust allowing them to work for complicated point configurations and

more intricate base measures. We also describe the different projectors used to discretize a trajectory with bounded speed and acceleration by an atomic measure. Finally we compare our robust algorithm for the computation of the L^2 transport distance with a pre-existing method [34], and with the convolution distance [21]. The comparison focuses on the numerical speed of the algorithms but also their theoretical complexity.

In Chapter 6, the discretization of the trajectory s is no more piecewise constant but piecewise linear and the base measure is supposed atomic. Thus the approximation measure is not atomic but is a measure carried by a polyline (continuous and piecewise linear trajectory). Still using the the algorithmical geometry library CGAL [128], a new algorithm for the computation of optimal transport is presented. It allows the computation of the optimal transport between an atomic measure and a measure carried by a polyline. Despite recent theoretical progress [96] proving the convergence of the *damped Newton* algorithm for measures carried by simplexes of Hausdorff dimension smaller than the ambient dimension, the problem has not been treated for measures carried by curves and is the object of Chapter 6. We coin this new problem $\frac{3}{4}$ -discrete optimal transport. In this specific case, the functional to optimize is less regular : \mathcal{C}^1 . The optimization method to use cannot be the same one and we have thus tried many methods in order to establish the quickest one.

In Chapter 7 we check that our previously made approximations are consistant. Indeed we have discretized a continuous trajectory as either a piecewise constant curve or a piecewise linear curve. A natural question that arises is that of the convergence speed of the approximation to the continuous curve. In this chapter we therefore bound the Hausdorff distance between polylines and atomic curves whose discrete derivatives are bounded with the set of curves with bounded derivatives. The result of this section is the one that we could expect : the approximation by a polyline converges quadratically whereas the piecewise continuous approximation only converges linearly.

Finally in Chapter 8 we provide a detailed description of the rapid integration method, used in Chapter 5 to compute the semi-discrete transport distance, allowing the conversion of Laguerre cell volume integrals into integrals on the boundaries via Green's formula. After some brief theoretical reminders, the problems related to maximizing the Kantorovitch functional in the curvling case are described. We expose with a great deal of precision the numerical difficulties that we were exposed to and the different methods that were tested. Finally, we examine the advantages of the regularized New-

ton method before presenting the multi-scale algorithm that allowed us to perform all of the numerical simulations for the curvling problem and for a great number of points.

Chapter 3

Approches variationnelles pour le stippling.

Contents

3.1	Introduction	35
3.2	Méthodes numériques	36
3.2.1	Distance de convolution	36
3.2.2	W_2 discret	37
3.2.3	W_2 discret régularisé	38
3.2.4	W_2 semi-discret	39
3.3	Comparaison des méthodes	40

Résumé

Le stippling est un problème qui a beaucoup progressé dernièrement grâce à l'introduction de méthodes variationnelles. On s'intéresse ici à deux types de formulations. L'une repose sur une distance L^2 entre mesures et fait appel à des outils d'analyse harmonique appliquée. L'autre repose sur la distance de Wasserstein et fait appel à des outils de géométrie algorithmique. Différentes méthodes de résolution et de discrétisation sont comparées et nous présentons leurs atouts et leurs limitations.

Abstract

Stippling is a problem that recently found elegant and efficient solutions thanks to the introduction of variational methods. The aim of this paper is to compare two state-of-the-art approaches : one is based on the minimization of an L^2 norm (with links to applied harmonic analysis), while the other is based on the Wasserstein distance (with links to computational geometry).

This chapter was published in

F. de Gournay, J. Kahn, L. Lebrat, and Pierre Weiss
Approches variationnelles pour le stippling : distance L^2 ou transport optimal ? In *GRETSI 2017 XXVI* Sept. 2017.

3.1 Introduction

Le stippling consiste à approcher une mesure de probabilité cible μ sur $\Omega \subset \mathbb{R}^2$ par une mesure discrète de la forme $\nu = \nu(\mathbf{p}) = \frac{1}{n} \sum_{i=1}^n \delta_{p_i}$, avec des positions $\mathbf{p} = (p_i)_{1 \leq i \leq n} \in \Omega^n$. Le principe des méthodes variationnelles est de trouver un ensemble de positions \mathbf{p} qui minimise une certaine distance $d(\mu, \nu)$ entre μ et ν :

$$\min_{\mathbf{p}} d(\mu, \nu(\mathbf{p})). \quad (3.1)$$

Deux types de distance ont été proposées récemment.

Distances L^2 . La première [118, 26] repose sur une distance L^2 , après régularisation par un noyau de convolution. Elle est définie par :

$$d_{L^2}(\mu, \nu(\mathbf{p})) = \frac{1}{2} \|h \star (\mu - \nu(\mathbf{p}))\|_2^2, \quad (3.2)$$

où $h \in C^0(\Omega)$ est un noyau régularisant.

Distances de transport. La seconde [34] utilise la distance W_2 de Wasserstein (aussi appelée distance de transport) définie par :

$$d_{W_2}(\mu, \nu) = \left(\inf_{\gamma \in \Pi(\nu, \mu)} \int_{\Omega^2} \|x - y\|^2 d\gamma(x, y) \right)^{\frac{1}{2}}, \quad (3.3)$$

où Π est l'ensemble des couplages entre μ et ν , i.e. l'ensemble des mesures sur Ω^2 dont les marginales sur x et y sont respectivement μ et ν .

Contribution. Cet article est une étude de l'efficacité de ces deux distances d'un point de vue numérique et qualitatif. Nous décrivons plusieurs algorithmes, dont certains sont originaux. Ils font appel à des outils très différents tels que l'analyse harmonique appliquée (transformées de Fourier non uniformes), la géométrie algorithmique (diagrammes de Laguerre), l'optimisation convexe et non convexe (méthodes proximales). Ces outils sont utilisés par des communautés différentes qui s'ignorent partiellement. Notre ambition est d'aider à rapprocher ces communautés en popularisant certaines approches encore méconnues par un grand nombre de chercheurs en traitement du signal.

3.2 Méthodes numériques

Sous une apparente simplicité, la classe de problèmes (3.1) est en fait extrêmement complexe. Par exemple, elle contient le problème de Thomson (placer des points de façon “uniforme” sur une sphère) qui est dans la liste de Smale [121] des problèmes ouverts de mathématiques à résoudre pour le XXIème siècle. Nous détaillons ci-dessous les méthodes numériques émergentes pour le résoudre lorsque le nombre n de diracs est grand. Dans le cas de la distance de transport, nous présenterons trois approches : discrète, discrète régularisée et semi-discrète.

3.2.1 Distance de convolution

Théorie Lorsque $d = d_{L^2}$, le problème (3.1) peut se réécrire sous la forme d’un problème non convexe sur les positions \mathbf{p} :

$$\min_{\mathbf{p} \in \Omega^n} \underbrace{\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n H(p_i - p_j)}_{F(\mathbf{p}) \equiv \text{Répulsion}} - \underbrace{\frac{1}{n} \sum_{i=1}^n \int_{\Omega} H(x - p_i) d\mu(x)}_{G(\mathbf{p}) \equiv \text{Attraction}}, \quad (3.4)$$

où H est défini par sa transformée de Fourier : $\hat{H} = |\hat{h}|^2$.

Le problème (3.4) peut être interprété comme un problème d’attraction-répulsion [118] : des particules chargées positivement sont attirées par les zones claires de l’image et elles se repoussent deux à deux.

Algorithmie Bien que le problème (3.4) soit fortement non convexe, il peut être résolu par des techniques simples de type descente de gradient [26]. Il y a cependant trois difficultés techniques importantes : le choix du noyau H , le calcul des intégrales et le calcul du gradient de la fonctionnelle. Le noyau H doit être choisi de telle manière que des particules éloignées interagissent ensemble, ce qui semble assurer empiriquement une convergence assez rapide des algorithmes de descente de gradient. En pratique, nous avons choisi $H(x) = \|x\|_2$ dans ce papier. Pour ce choix, on obtient formellement :

$$[\nabla F(\mathbf{p})][i] = \frac{1}{n^2} \sum_{j \neq i} \frac{p_i - p_j}{\|p_i - p_j\|}, \quad (3.5)$$

et on voit que l’intensité de la force de répulsion de la particule j sur la particule i est *indépendante* de leur distance. Les intégrales peuvent être calculées avec des formules de quadrature. Enfin, lorsque le nombre de particules

est grand, calculer le gradient (3.5) directement coûte $O(n^2)$ opérations. En pratique, on fait appel à des outils de simulation de particules tels que les méthodes multipolaires ou les transformées de Fourier non uniformes [112] qui permettent de réduire le coût par itération à $O(n \log(n/\epsilon))$ opérations où ϵ représente la précision désirée sur le calcul de sommations rapide.

3.2.2 W_2 discret

Théorie On suppose ici que μ est une mesure atomique de la forme

$$\mu = \sum_{j=1}^k \mu_j \delta_{m_j}. \quad (3.6)$$

où les points m_j appartiennent à une grille cartésienne. Ainsi, l'image à quantifier est déjà discrète. Dans ces conditions, le plan de transport γ^* est discret, c'est-à-dire qu'il s'écrit sous la forme $\gamma^* = \sum_{ij} \gamma_{ij} \delta_{p_i \times m_j}$. Soit C la matrice de coût définie par $C_{ij} = \|p_i - m_j\|_2^2$, on obtient $d_{W_2}(\mu, \nu) = \langle C, \gamma^* \rangle$ où $\langle \cdot, \cdot \rangle$ est le produit scalaire de Frobenius. Le plan de transport optimal, pour \mathbf{p} fixé, est donné par la solution du problème linéaire :

$$\gamma^*(\mathbf{p}) = \underset{\substack{\gamma \geq 0 \\ \sum_i \gamma_{ij} = \mu_j \\ \sum_j \gamma_{ij} = 1/n}}{\operatorname{argmin}} \langle C, \gamma \rangle, \quad (3.7)$$

et (3.3) se réécrit comme :

$$d_{W_2}(\mu, \nu(\mathbf{p})) = \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^*(\mathbf{p}) \|p_i - m_j\|_2^2. \quad (3.8)$$

À γ^* fixé, la minimisation de (3.8) donne

$$p_i = \frac{1}{n} \sum_j \gamma_{ij}^* m_j, \quad (3.9)$$

ce qui correspond à choisir la position p_i comme le barycentre des positions m_j pondérées par le plan de transport γ^* .

Pour résoudre (3.1) on propose d'utiliser une minimisation alternée en γ et en \mathbf{p} , à l'itération k , $\gamma^*(\mathbf{p}_k)$ est obtenu en résolvant (3.7) avec un solveur linéaire, puis \mathbf{p}_{k+1} est obtenu par la relation (3.9).

Algorithmie Le calcul de la distance de Wasserstein est un problème de programmation linéaire (PL) à nk variables. Beaucoup de solveurs permettent de le faire, mais ils sont trop coûteux pour les dimensions typiques rencontrées en traitement du signal (e.g. $k = 10^6$ et $n = 10^5$). Ce type de méthode ne peut donc être appliqué qu'en dimension relativement modérée. De plus, les solutions de (3.7) ne sont pas continues par rapport à la variable \mathbf{p} . Ce phénomène peut entraîner des instabilités numériques lors de l'optimisation.

3.2.3 W_2 discret régularisé

Théorie Quand le nombre de points de discrétisation ($n+k$) est grand, ou quand la mesure cible μ est bruitée, le plan de transport donné par (3.7) peut être très irrégulier. Pour régulariser le problème, une méthode popularisée récemment consiste à ajouter un terme entropique [32, 10]. Soit

$$E(\gamma) = \sum_{ij} \gamma_{ij} (\log(\gamma_{ij}) - 1) \quad (3.10)$$

l'entropie de γ . Dans ces méthodes, on remplace $\langle C, \gamma \rangle$ dans (3.7) par $\langle C, \gamma \rangle - \varepsilon E(\gamma)$. Le problème ε -régularisé devient alors strictement convexe et admet une unique solution, il peut être réécrit sous la forme :

$$W_{2,\varepsilon}(\mu, \nu) = \min_{\gamma \in \Pi(\nu, \mu)} \mathbf{KL}(\gamma | \xi(\varepsilon)) \text{ avec } \xi(\varepsilon) = e^{-\frac{C}{\varepsilon}}, \quad (3.11)$$

où $\mathbf{KL}(\gamma | \xi) = \sum_{ij} \gamma_{ij} (\log(\gamma_{ij}) - \log(\xi_{ij}) - 1)$ est la divergence de Kullback-Leibler.

Algorithmie Une méthode de résolution numérique popularisée récemment [32, 10], consiste à utiliser des algorithmes simples de projections itérées de Bregman, voir algorithme 1. Dans cet algorithme l'exponentielle de matrice, la division et la multiplication entre vecteurs sont des opérations terme à terme. Le vecteur $\boldsymbol{\mu}$ représente les poids de la mesure cible. Le nombre de variables d'optimisation du problème direct est limité ($n+k$) mais la matrice ξ contient nk éléments et ne doit en général pas être calculée. Les produits $\xi \mathbf{u}$ et $\xi \mathbf{v}$ peuvent être calculés efficacement avec la transformée de Fourier rapide si les points \mathbf{m} sont sur une grille cartésienne.

L'algorithme 1 fournit en général une solution approchée acceptable en une centaine d'itérations. Le coût de chaque itération est dominé par les produit $\xi \mathbf{v}$ et $\xi^T \mathbf{u}$ qui peuvent être effectués en $O(n \log(n))$ opérations.

Algorithm 1 Algorithme de Sinkhorn

Input : $C, \varepsilon, \boldsymbol{\mu}, \text{IterMax}$
Output : γ^* le plan de transport régularisé
Initialisation : $\mathbf{v} = \mathbb{1}, \xi = e^{-\frac{C}{\varepsilon}}$

- 1: **while** $1 \leq \text{nit} \leq \text{IterMax}$ **do**
- 2: $\mathbf{u} = \frac{\mathbb{1}}{n\xi\mathbf{v}}$
- 3: $\mathbf{v} = \frac{\boldsymbol{\mu}}{\xi^T\mathbf{u}}$
- 4: **end while**

Return $\gamma^* = \text{Diag}(\mathbf{u})\xi\text{Diag}(\mathbf{v})$;

Une fois γ^* obtenu par l'algorithme présenté précédemment, on utilise la relation (3.8) pour optimiser la position des points.

Le choix du paramètre de régularisation ε est très délicat. Si ε est trop grand, les masses de Dirac s'agglutinent (voir Figure 3.2). A l'inverse si ε est trop petit les vecteurs $u = \frac{\nu}{\xi\mathbf{v}}$ et $v = \frac{\boldsymbol{\mu}}{\xi^T\mathbf{u}}$ deviennent trop grand (ce qui arrive rapidement comme $\xi = e^{-\frac{C}{\varepsilon}}$) et ne peuvent pas être représentés en double précision. Globalement, on voit qu'une implémentation "naïve" de cette méthode n'est pas satisfaisante.

3.2.4 W_2 semi-discret

Théorie Le cas semi-discret consiste à ne pas décrire μ comme une somme de masses de Dirac, mais comme une mesure à densité continue. Par exemple, on peut trianguler les pixels de l'image et supposer que l'image est linéaire sur chaque triangle. Un grand avantage de cette formulation est que le plan de transport est unique [132], ce qui assure une certaine stabilité. La forme duale du problème (3.3) est donnée par [132] :

$$W_2(\mu, \nu) = \sup_{\substack{\phi \in L^1(\Omega), \\ \psi \in L^1(\Omega), \\ \text{s.c. } \phi(x) + \psi(y) \leq \|x-y\|_2^2}} \int_{\Omega} \phi(x) d\mu(x) + \int_{\Omega} \psi(y) d\nu(y), \quad (3.12)$$

La résolution exacte de ce problème selon la variable ϕ donne l'équation suivante :

$$W_2(\mu, \nu) = \sup_{\psi} \int_{\Omega} \min_i \|x - p_i\|_2^2 - \psi(p_i) d\mu + \frac{1}{n} \sum_i \psi(p_i). \quad (3.13)$$

Ce problème peut être résolu numériquement en introduisant les cellules de Laguerre définies par :

$$\mathcal{L}_i = \left\{ x \in \Omega, \|x - p_i\|_2^2 - w_i \leq \|x - p_j\|_2^2 - w_j, \forall j \neq i \right\}.$$

Le diagramme de Laguerre $(\mathcal{L}_i)_{1 \leq i \leq n}$ est l'union de n polygones convexes formant une partition de Ω [7]. En posant $\psi(P_i) = w_i$, le problème (3.12) se réécrit comme la maximisation de la fonctionnelle $g(w)$ [85], avec

$$g(w) = \sum_i \left(\int_{\mathcal{L}_i} (\|x - p_i\|_2^2 - w_i) \right) d\mu + \frac{1}{n} \sum_i w_i. \quad (3.14)$$

où $g(w)$ est une fonction concave, sa dérivée et sa Hessienne sont données dans [80].

Algorithme La solution w^* de (3.14) pour \mathbf{p} fixé peut être calculée avec un algorithme de Newton avec recherche linéaire de Wolfe [80]. A chaque évaluation du poids w^* , la position des points \mathbf{p} peut être optimisée grâce à un algorithme de type Lloyd : chaque masse de Dirac est déplacée sur le barycentre de la cellule de Laguerre associée :

$$p_i = \frac{\int_{\mathcal{L}_i} x d\mu(x)}{\mu(\mathcal{L}_i)}. \quad (3.15)$$

Dans notre implémentation, le calcul du diagramme de Laguerre est réalisé par la librairie CGAL [128]. Celui-ci peut être calculé en $O(n \log(n))$ opérations. Une fois les cellules de Laguerre (polygones) calculées, les intégrations nécessaires au calcul des dérivées de $g(w)$ sont réalisées rapidement grâce à la formule de Green (intégration de fonctions sur les arêtes uniquement). On peut montrer que le coût de calcul est en $O(\sqrt{kn})$ opérations. Cet algorithme couple une méthode de second ordre pour le calcul du transport optimal et une méthode de premier ordre pour l'optimisation de la position des points. Il fournit des résultats satisfaisants visuellement en très peu d'itérations. Par exemple, les auteurs de [34] suggèrent que quatre itérations suffisent en général.

3.3 Comparaison des méthodes

La Figure 3.1 montre un premier test de quantification de l'indicatrice d'un disque. La Figure 2 montre l'influence du paramètre de régularisation

ε introduit en Section 3.2.3. La Figure 3 montre un résultat de stippling sur un dégradé linéaire. Ces quelques résultats montrent que les méthodes les plus simples à implémenter (transport optimal discret et transport optimal régularisé) souffrent malheureusement de sérieux problèmes numériques qui les rendent inutilisables pour le problème de stippling. Le transport optimal semi-discret et la distance d_{L^2} fournissent tous les deux des résultats très satisfaisants, cependant l'implémentation est bien plus difficile. En terme de rapidité de calcul, le transport optimal semi-discret semble le meilleur. Cependant il n'est pas clair que ce constat reste valable en dimension supérieure, car la complexité de construction des diagrammes de Laguerre augmente, alors que la complexité de la NFFT est inchangée. Pour finir le tableau 3.4 donne quelques propriétés supplémentaires des différentes méthodes.

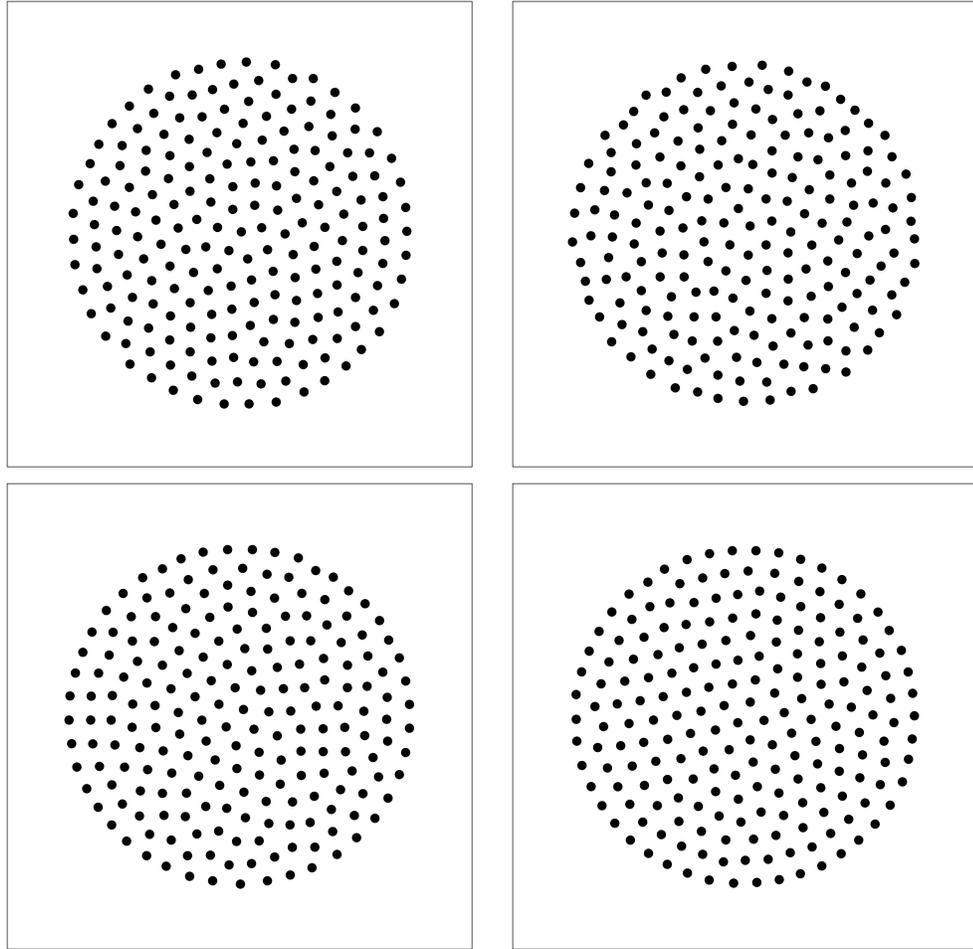


FIGURE 3.1 – Approximation de la fonction caractéristique d’un disque par les méthodes : L^2 (en haut à gauche), W_2 discret (en haut à droite), W_2 discret régularisé (en bas à gauche) et W_2 semi-discret (en bas à droite). Ici, toutes les méthodes produisent des résultats satisfaisants. Noter cependant les nombreuses irrégularités obtenues avec le transport optimal discret.

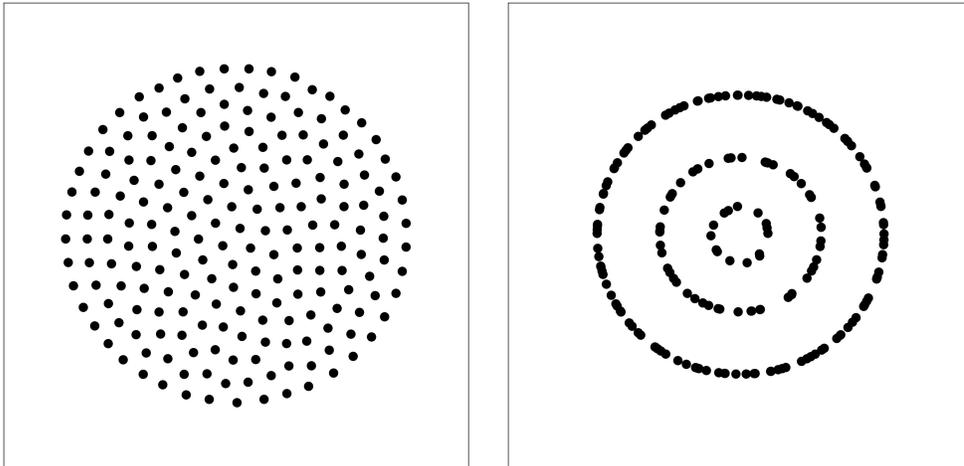


FIGURE 3.2 – Approximation de la fonction caractéristique d'un disque par la méthode de transport optimal régularisé pour un paramètre de régularisation ε valant respectivement $2.5 \cdot 10^{-4}$ et $6.0 \cdot 10^{-3}$. Si ε est trop important, la régularisation entropique a tendance à agréger les points plutôt que les disposer de façon uniforme.

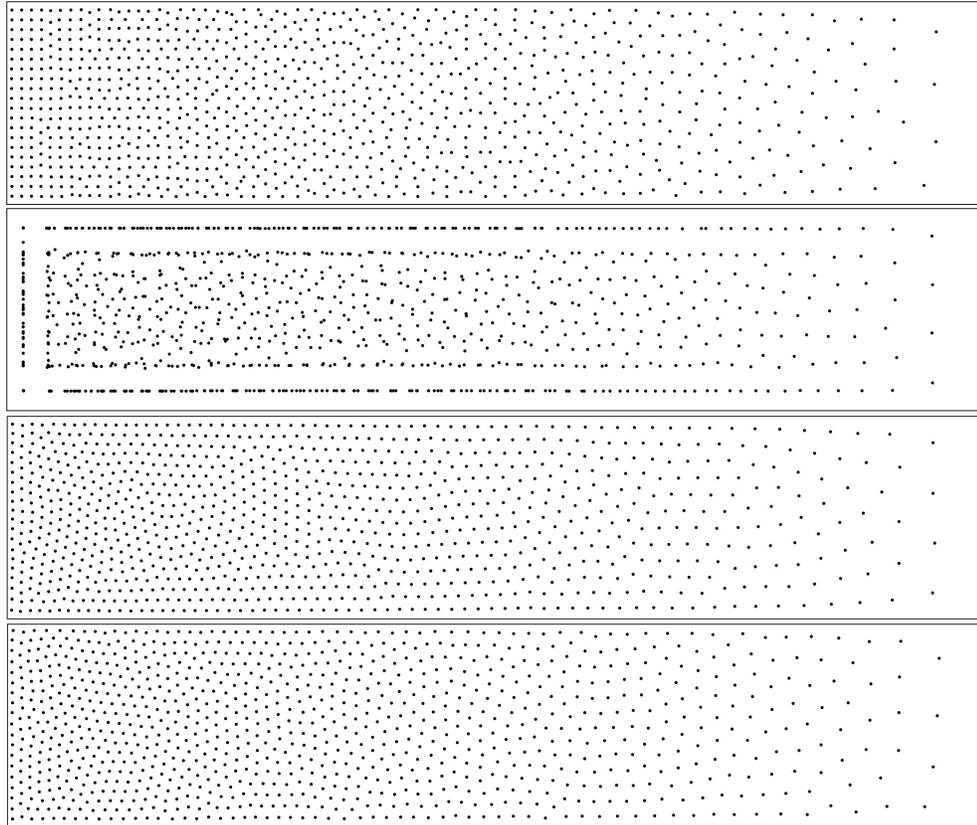


FIGURE 3.3 – Stippling d'un dégradé linéaire. De haut en bas : W_2 discret, W_2 discret régularisé, W_2 semi-discret et d_{L^2} . Noter que le transport optimal discret tend à aligner les points avec la grille, car les masses de Dirac de la mesure cible sont localisées sur une grille. Pour le transport régularisé, la valeur de ε est la plus petite possible avant que des NaN apparaissent, car des divisions par des nombres trop petits ont lieu.

Méthode	Propriétés					
	Invariance isométrie	Dim. supérieure	Unicité γ^*	Robustesse bruit	Bons résultats	Technologie
d_{L^2} (semi-discret)	✗	✓	✓	✓	✓	NFFT
d_{W_2} discret	✓	✓	✗	✗	✗	PL
d_{W_2} régularisé	✓	✓	✓	✓	✗	FFT
d_{W_2} semi-discret	✓	✗	✓	✓	✓	CGAL

FIGURE 3.4 – Quelques propriétés des différents algorithmes.

Chapter 4

Differentiation and regularity of semi-discrete optimal transport with respect to the parameters of the discrete measure.

Contents

4.1	Introduction	49
4.1.1	Semi-discrete optimal transport	49
4.1.2	Link with Voronoi diagrams	51
4.1.3	Organization of the paper	52
4.2	Second order differentiability	52
4.2.1	Hypotheses and notation	53
4.2.2	Main result	55
4.2.3	The Euclidean case	56
4.2.4	Technical lemmas	58
4.2.5	Proof of the results of Section 4.2	64
4.3	Numerical experiments	66
4.3.1	Lloyd's algorithm	68
4.3.2	Newton's Algorithm	69
4.3.3	Other considerations	70
4.3.4	Numerical results	70
4.3.5	Direct comparaisn of the algorithms	70

4.3.6	Adding a point	73
4.4	Conclusion	73

————— **Abstract** —————

This paper aims at determining under which conditions the semi-discrete optimal transport is twice differentiable with respect to the parameters of the discrete measure and exhibits numerical applications. The discussion focuses on minimal conditions on the background measure to ensure differentiability. We provide numerical illustrations in stippling and blue noise problems.

This chapter was published in

F. de Gournay, J. Kahn, and L. Lebrat

Differentiation and regularity of semi-discrete optimal transport with respect to the parameters of the discrete measure.

In *Numerische Mathematik*, 141(2) : 429 :453, 2019.

4.1 Introduction

Optimal transport [99, 77] is a blossoming subject that has known major breakthroughs these last decades. Its applications range from finance [107], mesh generation [44], PDE analysis [75] and imaging [114, 135] to machine learning and clustering [123, 50].

This paper is limited to the semi-discrete case, which consists in transporting discrete measures (Dirac masses) towards a background measure. However we allow more general background measures than the densities with respect to the Lebesgue measure that are usually found in the literature. In this setting, we prove second order differentiability of the optimal transport distance for an arbitrary cost with respect to the locations of the Dirac masses.

Precursors include De Goes [33] (Proposition 2.5.4) who has given the formula of the Hessian in the Euclidean setting. However he has given no proof of existence. We will make use of the framework developed by Kitagawa *et al.* [80] to overcome these restrictions.

As a by-product, we obtain the second order differentiability conditions for the so-called energy of a Voronoi diagram. The latter remark generalizes results presented in [42, 89] to higher dimensions and lower regularity of the background measure.

4.1.1 Semi-discrete optimal transport

The optimal transport [132] between two probability measures μ and ν defined respectively on the spaces X and Y with cost $c : Y \times X \rightarrow \mathbb{R}^+$ is the minimization problem :

$$\min_{\gamma \in \Pi(\nu, \mu)} \int_{Y \times X} c(y, x) d\gamma(y, x), \quad (4.1)$$

where $\Pi(\nu, \mu)$ is the set of positive measures on $Y \times X$ with marginal distributions on Y (resp. X) equal to ν (resp. μ), that is :

$$\gamma \in \Pi(\nu, \mu) \Leftrightarrow \begin{cases} \int_Y \phi(y) d\mu(y) = \int_{Y \times X} \phi(y) d\gamma(y, x) & \forall \phi \in L^1(\mu) \\ \text{and} \\ \int_X \psi(x) d\nu(x) = \int_{Y \times X} \psi(x) d\gamma(y, x) & \forall \psi \in L^1(\nu) \end{cases} .$$

Intuitively, a coupling γ may be seen as a way to transport the mass of μ to the mass of ν . Specifically $\gamma(B, A)$ is the mass moved from A to B .

Interpreting ϕ and ψ as Lagrange multipliers of the constraint $\Pi(\mu, \nu)$ and using a standard inf-sup inversion (see [132] for details), one derives the Kantorovitch dual problem :

$$\sup_{\phi(y)+\psi(x)\leq c(y,x)} \int_Y \phi(y)d\mu(y) + \int_X \psi(x)d\nu(x).$$

When ϕ is given, it can be explicitly solved in ψ and the problem turns into

$$\sup_{\phi} g(\phi) \quad \text{with} \quad g(\phi) = \int_Y \phi(y)d\mu(y) + \int_X \inf_{y \in Y} (c(y, x) - \phi(y))d\nu(x), \quad (4.2)$$

where the function g , as a dual function, is naturally concave.

Recall that a Lipschitz set is a set for which the boundary is, up to a partition and rotations, the graph of a Lipschitz function. Suppose that the support of ν is included in a bounded convex Lipschitz set $\Omega \subset \mathbb{R}^d$, and that μ is a discrete measure on \mathbb{R}^d , that is, given $n \in \mathbb{N}$, there exists $z = (z^i)_{i=1..n}$ with $z^i \in \mathbb{R}^d$ and $m = (m^i)_{i=1..n} \in \mathbb{R}^n$ such that

$$\mu = \sum_{i=1}^n m^i \delta_{z^i},$$

where δ_{z^i} is a Dirac measure located at z^i . In this case the set of test functions ϕ can be identified to \mathbb{R}^n , so that $\phi = (\phi^i)_{i=1..n} \in \mathbb{R}^n$.

Finally introducing the Laguerre tessellation [5] defined by its cells

$$\mathcal{L}_i(z, \phi) = \{x \in \Omega \text{ such that } c(z^i, x) - \phi^i \leq c(z^j, x) - \phi^j \quad \forall j \in \llbracket 1, n \rrbracket\}, \quad (4.3)$$

so that the final formulation of the optimal transport problem (4.1) in the semi-discrete setting is

$$\sup_{\phi} g(\phi, z, m) \quad \text{with} \\ g(\phi, z, m) = \sum_{i=1}^n \int_{\mathcal{L}_i(z, \phi)} (c(z^i, x) - \phi^i) \frac{1}{\#\mathcal{M}^{-1}(\{x\})} d\nu(x) + \sum_{i=1}^n \phi^i m^i, \quad (4.4)$$

where $\#\mathcal{M}^{-1}(\{x\})$ is defined in Section 4.2.1 as the counting factor of the number of Laguerre cells containing x .

The Laguerre cells $\mathcal{L}_i(z, \phi_{opt})$ associated to an optimal ϕ_{opt} in the maximization (4.4) are the ‘‘arrival’’ zones of the mass located at each z^i by a coupling γ of optimal transportation, namely :

$$\begin{aligned} \gamma(z^i, A) &= \nu(A) & \forall A \subset \mathring{\mathcal{L}}_i(z, \phi_{opt}), \\ \gamma(z^i, B) &= 0 & \text{if } B \cap \mathcal{L}_i(z, \phi_{opt}) = \emptyset. \end{aligned}$$

We aim at studying the differentiation properties up to the second order of $g(\phi, z, m)$ with respect to its parameters. The differentiation with respect to m is rather straightforward and will not be discussed hereafter. The second order differentiability of g with respect to ϕ is known [95, 34, 85] and proved in [80]. This proof mainly uses that the Laguerre cells $\mathcal{L}_i(z, \phi)$ are the intersection for all $j \neq i$ of the sub-level sets (with respect to the value of $\phi^i - \phi^j$) of the function $x \mapsto c(z^i, x) - c(z^j, x)$. Using the co-area formula, the authors are able to compute the differential of g with respect to ϕ . Differentiating with respect to z is more involved and is the main goal of the present paper. Note also that the authors of [80] prove $C^{2,\alpha}$ regularity with respect to ϕ , whereas we only deal with second order derivatives. Hence our set of hypotheses is slightly different.

Proposition 4.2.2 provides an easier version of the main theorem by restricting to the Euclidean case. Readers only interested in the standard L^2 transportation distance might want to skip directly to Proposition 4.2.2.

4.1.2 Link with Voronoi diagrams

The Voronoi diagram $\{\mathcal{V}_i(z)\}_i$ is the special case of the Laguerre tessellation when $\phi = 0$ and the cost is the square Euclidean distance, as can be seen from definition (4.3),

$$\mathcal{V}_i(z) = \mathcal{L}_i(z, 0) = \{x \in \Omega \text{ such that } \|z^i - x\|_2^2 \leq \|z^j - x\|_2^2 \quad \forall j \in \llbracket 1, n \rrbracket\}.$$

Moreover, as noticed by Aurenhammer *et al.* [5], for any choice of Lagrange multipliers ϕ , the vector of masses $m^i = \nu(\mathcal{L}_i(z, \phi))$ turns ϕ into a solution of (4.4). In particular, the Voronoi diagram is the Laguerre tessellation of the optimal transport problem associated to the choice of mass $\tilde{m}^i := \nu(\mathcal{V}_i(z))$. Now recall that ϕ is a Lagrange multiplier for the mass constraint in (4.1), so that setting $\phi = 0$ releases this constraint and the corresponding choice of \tilde{m} minimizes the optimal transportation cost :

$$\begin{aligned} g(0, z, \tilde{m}) &= \sup_{\phi} g(\phi, z, \tilde{m}) \\ &= \inf_m \sup_{\phi} g(\phi, z, m). \end{aligned} \tag{4.5}$$

Another more physical interpretation of (4.5) is that, without mass constraints, the best way to transport a measure ν to a finite number of points is to send

each part of ν to its closest neighbour. Hence we build the Voronoi diagram of the points.

The expression (4.5) has been coined as the energy of the Voronoi diagram :

$$G_S(z) = g(0, z, \tilde{m}) = \sum_i \int_{\mathcal{V}_i(z)} \|x - z^i\|^2 d\nu(x) \quad (4.6)$$

Finding critical points of this energy G_S is also known as the centroidal Voronoi tessellation (CVT) problem. Indeed, at a critical point \bar{z} of G_S , each \bar{z}^i is the barycenter of $\mathcal{V}_i(\bar{z})$ with respect to the measure ν :

$$\bar{z}^i \int_{\mathcal{V}_i(\bar{z})} d\nu(x) = \int_{\mathcal{V}_i(\bar{z})} x d\nu(x).$$

Results of second order differentiability of G_S with respect to z has been proven in [89] and inferred in many different previous papers [73, 3, 43]. However those papers do not tackle the question of the regularity of ν . Moreover the cost c in the Voronoi setting is the square of the Euclidean distance. Our work is a generalization of both of this points. Indeed differentiability of G_S stems from differentiability of g , see equations (4.17) and (4.19).

4.1.3 Organization of the paper

In Section 4.2, the main result is given. The hypotheses needed to ensure second order differentiability are given in 4.2.1, the result is stated in Section 4.2.2, Theorem 4.2.1 and is reformulated in the Euclidean case in Section 4.2.3. The rest of Section 4.2 is devoted to the proof of Theorem 4.2.1. Section 4.3 presents some numerical results.

4.2 Second order differentiability

The main goal of this section is to state, in Theorem 4.2.1, the sufficient conditions that ensure differentiability of second order of the function

$$\tilde{g}(\phi, z) = \int_{\Omega} \min_i (c(z^i, x) - \phi^i) d\nu(x),$$

which yields immediately the second order derivatives of $g = \tilde{g} + \phi \cdot m$ defined in (4.4).

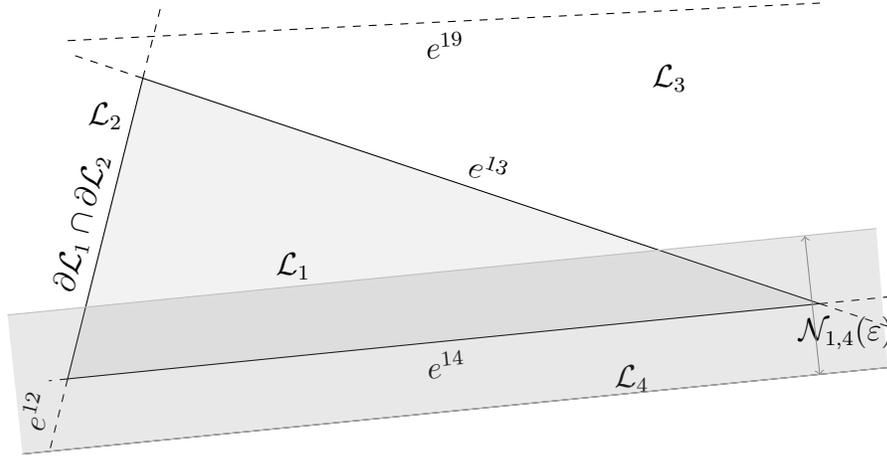


FIGURE 4.1 – Typical example of e^{ik} and the neighborhoods $\mathcal{N}_{ik}(\varepsilon)$ for $i = 1$.

4.2.1 Hypotheses and notation

In order to state the hypotheses required for Theorem 4.2.1, additional notation is required. For all $x \in \Omega$, denote $\mathcal{M}(x)$ the subset of $\llbracket 1, n \rrbracket$ given by

$$\mathcal{M}(x) = \left\{ i \in \llbracket 1, n \rrbracket \text{ s.t. } c(z^i, x) - \phi^i \leq c(z^j, x) - \phi^j \quad \forall j \in \llbracket 1, n \rrbracket \right\}. \quad (4.7)$$

The Laguerre cell $\mathcal{L}_i(z, \phi)$ is then exactly given by

$$x \in \mathcal{L}_i(z, \phi) \Leftrightarrow i \in \mathcal{M}(x).$$

For $i \neq k \in \llbracket 1, n \rrbracket$, denote

$$e^{ik} := \{x \in \mathbb{R}^d \text{ s.t. } c(z^i, x) - \phi^i = c(z^k, x) - \phi^k\}. \quad (4.8)$$

Note that $\mathcal{L}_i(z, \phi) \cap \mathcal{L}_k(z, \phi)$ is included in e^{ik} but the converse fails to be true. First notice that e^{ik} is not included in Ω whereas the Laguerre cells are included in Ω by definition. Second e^{ik} is only the “competition zone” between the i^{th} and the k^{th} Laguerre cells but it may (and will) happen that $x \in e^{ik}$ is included in $\mathcal{L}_j(z, \phi)$ for some other j and in neither $\mathcal{L}_i(z, \phi)$ nor $\mathcal{L}_k(z, \phi)$. Note also that the dependence of both e^{ik} and $\mathcal{M}(x)$ in (z, ϕ) is dropped to shorten the notation.

For all i and k , we denote the ε -neighborhood of e^{ik} by $\mathcal{N}_{ik}(\varepsilon)$. By convention, e^{i0} is the boundary of Ω and thus $\mathcal{N}_{i0}(\varepsilon)$ is an ε -neighborhood of $\partial\Omega$. Figure 4.1 illustrates these geometric objects.

In the following definitions σ denotes the $d - 1$ -Hausdorff measure. The sufficient geometric hypotheses on the Laguerre tessellation for second order differentiability are :

Definition 4.2.0.1 (Diff-2). *We say that hypothesis (Diff-2) holds at point (z_0, ϕ_0) iff*

- *for all $1 \leq i \leq n$, $(z, x) \mapsto c(z^i, x)$ is $W^{2,\infty}(B(z_0, r) \times \Omega)$, where $B(z_0, r)$ is a ball around the point z_0 .*
- *there exists $\varepsilon > 0$ such that for all $1 \leq k \neq i \leq n$, $\forall x \in e^{ik}$*

$$\|\nabla_x c(z_0^i, x) - \nabla_x c(z_0^k, x)\| \geq \varepsilon, \quad (\text{Diff-2-a})$$

- *for all i , there exists $s, C > 0$ such that for all $0 \leq k \neq j \leq n$, for all $\varepsilon, \varepsilon'$ in $]0, s[$, it holds*

$$|\mathcal{N}_{ik}(\varepsilon) \cap \mathcal{N}_{ij}(\varepsilon')| \leq C\varepsilon\varepsilon'. \quad (\text{Diff-2-b})$$

and

$$\lim_{\varepsilon \rightarrow 0} \sigma(e^{ik} \cap \mathcal{N}_{ij}(\varepsilon)) = 0, \quad (\text{Diff-2-c})$$

In the above definition, all the geometric objects $e^{ik}, \mathcal{N}_{ik}(\varepsilon), \mathcal{N}_{ij}(\varepsilon)$ are defined with parameters $(z, \phi) = (z_0, \phi_0)$.

The geometric hypothesis for continuity of second order derivatives is

Definition 4.2.0.2 (Cont-2). *We say that hypothesis (Cont-2) holds at point (z_0, ϕ_0) iff there exists $C > 0$ such that for all i, j*

$$\sigma(e^{ij} \cap \Omega) \leq C, \quad (\text{Cont-2})$$

where e^{ij} is defined in (4.8) with parameters $(z, \phi) = (z_0, \phi_0)$.

Remark 4.2.1. *At some point, we closely follow the ideas of [80]. Our hypotheses (Diff-2) and (Cont-2) are a consequence of more general geometric hypotheses in [80], in particular Loeper's condition. Using Loeper's condition requires an exponential mapping to reduce the problem to a convex one. Hence the proofs of this paper would be even more technical and hard to read.*

4.2.2 Main result

Directional derivative of \tilde{g} can be obtained using very mild assumptions on the cost function c and the approximated measure ν .

Proposition 4.2.1. *Set Ω a bounded Lipschitz convex set and z_0 a vector of positions of Dirac masses. Suppose that for all i and for ν -almost every x , the function $y \mapsto c(y, x)$ is differentiable around $y = z_0^i$ and that there exists $h \in L^1(\Omega, \nu)$ with $|\nabla_y c(z_0^i, x)| \leq h(x)$ ν -a.e. for all z around z_0 . Then \tilde{g} is directionally derivable at point (ϕ_0, z_0) with derivative given by :*

$$\begin{aligned} & \lim_{t \rightarrow 0^+} \frac{\tilde{g}((\phi_0 + td_\phi, z_0 + td_z)) - \tilde{g}(\phi_0, z_0)}{t} \\ &= \sum_{A \subset \llbracket 1, n \rrbracket} \int_{\mathcal{M}^{-1}(A)} \min_{i \in A} (\langle \nabla_y c(z_0^i, x), d_z^i \rangle - d_\phi^i) d\nu(x), \end{aligned}$$

where \mathcal{M} is defined in (4.7) with parameters $(z, \phi) = (z_0, \phi_0)$. If $\nu(\mathcal{M}^{-1}(A)) = 0$ for each A of cardinal ≥ 2 , it holds that \tilde{g} is differentiable and

$$\partial_z \tilde{g}(\phi_0, z_0) = \sum_{i=1}^n \int_{\mathcal{L}_i(z_0, \phi_0)} \nabla_y c(z_0^i, x) d\nu(x) \quad \text{and} \quad \frac{\partial \tilde{g}}{\partial \phi^i}(\phi_0, z_0) = -\nu(\mathcal{L}_i(z_0, \phi_0))$$

If in addition for ν -almost every x , the function $y \mapsto \nabla_y c(y, x)$ is continuous around z_0^i for all i , then \tilde{g} is C^1 around (ϕ_0, z_0) .

The theorem ensuring second order differentiability is :

Theorem 4.2.1. *If the hypotheses of Proposition 4.2.1 and (Diff-2) hold at point (ϕ_0, z_0) , and if ν admits a density with respect to the Lebesgue measure which is $W^{1,1}(\Omega) \cap L^\infty(\Omega)$ then \tilde{g} is twice differentiable at point (ϕ_0, z_0) . Let σ denote the $d - 1$ Hausdorff measure, m be the density of ν , and $k \neq l$ be in $\llbracket 1, n \rrbracket$, the formula for the Hessian at point (ϕ_0, z_0) is given by :*

$$\begin{aligned}
\frac{\partial^2 \tilde{g}}{\partial \phi^k \partial \phi^l} &= \int_{\partial \mathcal{L}_k \cap \partial \mathcal{L}_l} \frac{1}{\|\nabla_x c(z_0^k, x) - \nabla_x c(z_0^l, x)\|} m(x) d\sigma \\
\frac{\partial^2 \tilde{g}}{(\partial \phi^k)^2} &= \sum_{j \neq k} \int_{\partial \mathcal{L}_k \cap \partial \mathcal{L}_j} \frac{-1}{\|\nabla_x c(z_0^k, x) - \nabla_x c(z_0^j, x)\|} m(x) d\sigma \\
\frac{\partial^2 \tilde{g}}{\partial \phi^k \partial z^l} &= \int_{\partial \mathcal{L}_k \cap \partial \mathcal{L}_l} \frac{-\nabla_y c(z_0^l, x)}{\|\nabla_x c(z_0^k, x) - \nabla_x c(z_0^l, x)\|} m(x) d\sigma \\
\frac{\partial^2 \tilde{g}}{\partial \phi^k \partial z^k} &= \sum_{j \neq k} \int_{\partial \mathcal{L}_k \cap \partial \mathcal{L}_j} \frac{\nabla_y c(z_0^k, x)}{\|\nabla_x c(z_0^k, x) - \nabla_x c(z_0^j, x)\|} m(x) d\sigma \\
\frac{\partial^2 \tilde{g}}{\partial z^k \partial z^l} &= \int_{\partial \mathcal{L}_k \cap \partial \mathcal{L}_l} \frac{\nabla_y c(z_0^k, x) \otimes \nabla_y c(z_0^l, x)}{\|\nabla_x c(z_0^k, x) - \nabla_x c(z_0^l, x)\|} m(x) d\sigma \\
\frac{\partial^2 \tilde{g}}{(\partial z^k)^2} &= \int_{\mathcal{L}_k} \frac{\partial^2 c}{\partial y^2}(z_0^k, x) d\nu(x) - \sum_{j \neq k} \int_{\partial \mathcal{L}_k \cap \partial \mathcal{L}_j} \frac{\nabla_y c(z_0^k, x) \otimes \nabla_y c(z_0^j, x)}{\|\nabla_x c(z_0^k, x) - \nabla_x c(z_0^j, x)\|} m(x) d\sigma
\end{aligned}$$

Note that the Laguerre cells in consideration are computed for parameters $(z, \phi) = (z_0, \phi_0)$ and that for each $j \in \llbracket 1, n \rrbracket$, z^j is a vector in \mathbb{R}^d . As a consequence, differentiating \tilde{g} once with respect to z^j yields a vector in \mathbb{R}^d and differentiating \tilde{g} twice with respect to z^j yields a matrix. Recall that the derivatives in y of c refer to derivatives in the first variable. If in addition (Cont-2) holds at point (ϕ_0, z_0) and if the density m is $C^0(\Omega)$ then \tilde{g} is C^2 .

4.2.3 The Euclidean case

This section deals specially with the Euclidean case $c(z^i, x) = \frac{1}{2} \|z^i - x\|_2^2$. Then we have

Proposition 4.2.2. *If $c(y, x) = \frac{1}{2} \|y - x\|_2^2$, all the hypotheses of Theorem 4.2.1 are verified if ν admits a $C^0(\Omega) \cap W^{1,1}(\Omega)$ density with respect to the Lebesgue measure, and $z_0^i \neq z_0^j$ for $i \neq j$ and if the Laguerre tessellation for (z_0, ϕ_0) has no cell of zero Lebesgue volume. In this case the first order formulas are given by :*

$$\frac{\partial \tilde{g}}{\partial z_i}(\phi_0, z_0) = \int_{\mathcal{L}_i} (z_0^i - x) d\nu(x) \quad \text{and} \quad \frac{\partial \tilde{g}}{\partial \phi_i}(\phi_0, z_0) = -\nu(\mathcal{L}_i)$$

and the second order formula are given by :

$$\begin{aligned}
\frac{\partial^2 \tilde{g}}{\partial \phi^i \partial \phi^j}(\phi_0, z_0) &= \int_{\partial \mathcal{L}_i \cap \partial \mathcal{L}_j} \frac{m(x)}{\|z_0^i - z_0^j\|} d\sigma \quad \text{if } i \neq j \\
\frac{\partial^2 \tilde{g}}{(\partial \phi^i)^2}(\phi_0, z_0) &= - \sum_{j \neq i} \frac{\partial^2 \tilde{g}}{\partial \phi^i \partial \phi^j}(\phi_0, z_0) \\
\frac{\partial^2 \tilde{g}}{\partial \phi^i \partial z^j}(\phi_0, z_0) &= - \int_{\partial \mathcal{L}_i \cap \partial \mathcal{L}_j} \frac{(z_0^j - x)m(x)}{\|z_0^i - z_0^j\|} d\sigma \quad \text{if } i \neq j \\
\frac{\partial^2 \tilde{g}}{\partial \phi^i \partial z^i}(\phi_0, z_0) &= - \sum_{j \neq i} \frac{\partial^2 \tilde{g}}{\partial \phi^i \partial z^i} \\
\frac{\partial^2 \tilde{g}}{\partial z^i \partial z^j}(\phi_0, z_0) &= \int_{\partial \mathcal{L}_i \cap \partial \mathcal{L}_j} \frac{(z_0^j - x)(z_0^i - x)^T m(x)}{\|z_0^i - z_0^j\|} d\sigma \quad \text{if } i \neq j \\
\frac{\partial^2 \tilde{g}}{(\partial z^i)^2}(\phi_0, z_0) &= I_d \nu(\mathcal{L}_i) - \sum_{j \neq i} \int_{\partial \mathcal{L}_i \cap \partial \mathcal{L}_j} \frac{(z_0^i - x)(z_0^j - x)^T m(x)}{\|z_0^i - z_0^j\|} d\sigma,
\end{aligned}$$

where the Laguerre cells $\mathcal{L}_i, \mathcal{L}_j$ in consideration are computed with parameters (z_0, ϕ_0) and I_d is the identity matrix of dimension d .

Proof 4.2.1. In the Euclidean case the regularity assumption on c is trivially satisfied. Moreover e^{ik} is an hyperplane and Ω is bounded so that (Cont-2) is trivial.

A direct computation shows that

$$\|\nabla_x c(z_0^i, x) - \nabla_x c(z_0^j, x)\| = \|z_0^i - z_0^j\|,$$

which is non zero by hypothesis and hence uniformly non-zero, so that Hypothesis (Diff-2-a) is satisfied.

For $j, k \neq 0$, the sets $\mathcal{N}_{ik}(\varepsilon)$ are ε -neighbourhoods of the hyperplane e^{ik} , so that (Diff-2-b) and (Diff-2-c) are verified as soon as the hyperplanes e^{ik} and e^{ij} are different for $j \neq k$. On the other hand, it is impossible that $e^{ik} = e^{ij}$ for any $j \neq k$. Indeed, by the definition (4.8), it would mean that z_0^i, z_0^k and z_0^l are aligned and that the Laguerre cell corresponding to the point between the two others has empty interior, contradicting the hypotheses of the theorem.

Similarly, if $e^{ik} \cap \partial \Omega$ is not reduced to at most two points, by the convexity of Ω , the set Ω lies on one side of e^{ik} and one of the two Laguerre cells $\mathcal{L}_i(z_0, \phi_0)$ or $\mathcal{L}_k(z_0, \phi_0)$ is therefore empty. This final argument proves the case (Diff-2-b) and (Diff-2-c) for $j = 0$.

Now let A be of cardinal ≥ 2 . Let i and k belong to A , then $\mathcal{M}^{-1}(A)$ is included in e^{ik} , but e^{ik} is an hyperplane which is of zero Lebesgue measure, hence $\nu(\mathcal{M}^{-1}(A)) = 0$.

The rest of the hypotheses of Proposition 4.2.1 is trivial to prove.

As proved in [80], the constant C appearing in (Diff-2-b) depends on the minimal angle between the intersection of two competition zones e^{ik} and e^{il} . This constant is non-zero since there is a finite number of such intersections and it drives the $C^{2,\alpha}$ regularity of the function g .

4.2.4 Technical lemmas

This section is devoted to proving two technical lemmas, the first one ensures second-order differentiability of the function \tilde{g} and the second one ensures continuity. In this section, fix i , fix a C^∞ mapping $t \mapsto (z(t), \phi(t))$ that we aim at deriving at time $t = 0$. Set s small enough and consider only $t \in [0, s]$. Throughout this section the objects that depend on (z, ϕ) (say of the Laguerre cell $\mathcal{L}_j(z, \phi)$) will be written as depending on t (with the obvious notation $\mathcal{L}_j(t)$). Denote

$$u_t^{ik}(x) := c(z^i(t), x) - \phi^i(t) - (c(z^k(t), x) - \phi^k(t)).$$

Denote $(u_t^{ik})^{-1}(0) := \{x \in \mathbb{R}^d \text{ s.t. } u_t^{ik}(x) = 0\}$. Note that $(u_0^{ik})^{-1}(0) = e^{ik}$, where e^{ik} is defined in (4.8), Section 4.2.1.

Lemma 4.2.2. *Suppose that the Laguerre tessellation verifies (Diff-2). Let*

$$\xi : t \mapsto \int_{\mathcal{L}_i(t)} f(x, t) dx,$$

with f in $W^{1,1}(\Omega \times \mathbb{R}) \cap L^\infty(\Omega \times \mathbb{R})$ then ξ is derivable at time $t = 0$ with :

$$\partial_t \xi(0) = \sum_k \int_{\mathcal{L}_i(0) \cap \mathcal{L}_k(0)} \frac{\partial_t u_0^{ik}(x)}{\|\nabla_x u_0^{ik}(x)\|} f(x, 0) d\sigma + \int_{\mathcal{L}_i(0)} \partial_t f(x, 0) dx,$$

where σ is the $d - 1$ Hausdorff measure.

Lemma 4.2.3. *Suppose the Laguerre tessellation verifies (Cont-2). Let f be continuous, then $\xi : t \mapsto \int_{\mathcal{L}_i(t) \cap \mathcal{L}_k(t)} f(x, t) d\sigma$ is continuous.*

These lemmas are proven using tools of differential geometry via a bi-Lipshitz map θ that maps approximatively $\mathcal{L}_i(0)$ to $\mathcal{L}_i(t)$. The organization of this section is as follows : In Section 4.2.4.1 the map θ is built, and it is shown that $\theta(\mathcal{L}_i(0)) \simeq \mathcal{L}_i(t)$. The lemmas are then proven in Section 4.2.4.2.

4.2.4.1 Construction of the flow

For any $k \neq i, k > 0$, (Diff-2-a) ensures that $\|\nabla_x u_0^{ik}(x)\|$ is uniformly non-zero on $(u_0^{ik})^{-1}(0)$. By Lipschitz regularity of $\nabla_x u$, $\|\nabla_x u_t^{ik}(x)\|$ is uniformly non-zero for all $x \in \mathcal{N}_{ik}(s), t \in [0, s]$, provided s is chosen small enough. Hence the vector field defined as :

$$V_t^k(x) := -\partial_t u_t^{ik}(x) \frac{\nabla_x u_t^{ik}(x)}{\|\nabla_x u_t^{ik}(x)\|^2} \quad \forall x \in \mathcal{N}_{ik}(s), t \in [0, s],$$

is Lipschitz and can be extended as wanted outside $\mathcal{N}_{ik}(s)$.

The flow θ^k associated to V_t^k is defined as :

$$\theta_0^k(x) = x \text{ and } \partial_t \theta_t^k(x) = V_t^k(\theta_t^k(x)). \quad (4.9)$$

The flow θ^k preserves the level-sets of u_t^{ik} in the sense that for all x , the mapping $t \mapsto u_t^{ik}(\theta_t^k(x))$ is a constant as long as $\theta_t^k(x)$ remains in $\mathcal{N}_{ik}(s)$. Hence the flow θ_t^k preserves the competition zone between $\mathcal{L}_i(t)$ and $\mathcal{L}_k(t)$.

The objective is to build a flow θ which preserves the whole boundary of $\mathcal{L}_i(t)$. To that end, introduce :

$$\mathcal{N}_{ik}^*(\varepsilon) = \bigcup_{0 \leq j \neq k \leq n} \mathcal{N}_{ij}(\varepsilon),$$

and denote $\delta_k(x)$ the distance function to $\mathcal{N}_{ik}^*(0)$

$$\delta_k(x) = \inf_y \left\{ \|y - x\|, \text{ s.t. } y \in \partial\Omega \cup \left(\bigcup_{1 \leq j \neq k \leq n} e^{ij} \right) \right\}.$$

Set ζ a non-decreasing $C^\infty(\mathbb{R}^+, \mathbb{R})$ function equal to zero on $[0, 1/2]$ and to one on $[1, +\infty[$ and for all $x, 0 \leq t \leq s$ define

$$\tilde{V}_t^k(x) = \zeta\left(\frac{\delta_k(x)}{s}\right) V_t^k(x) \quad V_t(x) = \sum_{k=1}^n \tilde{V}_t^k(x)$$

Then V_t is equal to V_t^k on $\mathcal{N}_{ik}(s) \setminus \mathcal{N}_{ik}^*(s)$. One can safely interpret that $V_t = V_t^k$ on the edge of $\mathcal{L}_i(t)$ that is shared with $\mathcal{L}_k(t)$ and has been smoothed to zero on every corner of $\mathcal{L}_i(0)$.

Denote θ the flow associated to Lipschitz vector field V . We claim that

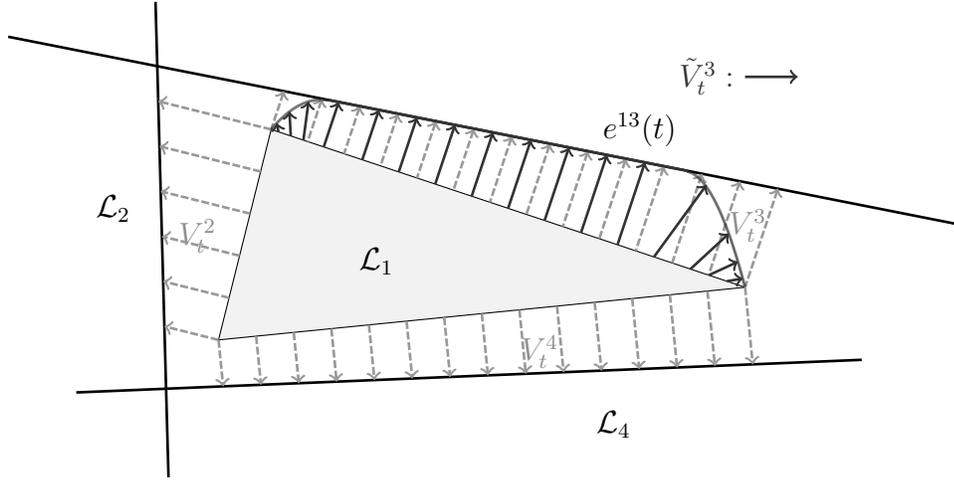


FIGURE 4.2 – Example of the vector fields V_t^3 and \tilde{V}_t^3 for $i = 1$

Proposition 4.2.3. *There exists $C, c > 0$ such that for all s small enough, for all k , and $0 \leq t \leq cs$, the symmetric difference between $\theta_t(\mathcal{L}_i(0))$ and $\mathcal{L}_i(t)$ has Lebesgue volume bounded by Cst .*

Proof Note first that the vector field V_t is always zero on $\partial\Omega$ so that $\theta_t(\Omega) = \Omega$ for all t . In the sequel C_v denotes an upper bound of the velocity of θ and θ^k . Set $c \leq 1/C_v$, then for all k :

$$\theta_t^k(e^{ik}) \subset \mathcal{N}_{ik}(s), \quad (4.10)$$

and then $\theta_t^k(e^{ik}) = (u_t^{ik})^{-1}(0)$. Let $x \in \theta_t(\mathcal{L}_i(0)) \Delta \mathcal{L}_i(t)$ and denote x_0 such that $x = \theta_t(x_0)$. We claim that there exists $k \in \llbracket 1, n \rrbracket$ and $0 \leq r_1, r_2 \leq t$ such that $u_{r_1}^{ik}(\theta_{r_1}(x_0)) = 0$ and $u_{r_2}^{ik}(\theta_{r_2}(x_0)) \neq 0$.

Indeed, if for instance $x \in \theta_t(\mathcal{L}_i(0))$ but $x \notin \mathcal{L}_i(t)$, then trivially $x_0 \in \mathcal{L}_i(0)$, meaning that for all k , $u_0^{ik}(x_0) \leq 0$. But $x \notin \mathcal{L}_i(t)$ means that there exists a k such that $u_t^{ik}(\theta_t(x_0)) > 0$. The continuity of the mapping $t \mapsto u_t^{ik}(\theta_t(x))$ ensures that for some r_1 we have $u_{r_1}^{ik}(\theta_{r_1}(x_0)) = 0$. The other case is done the same way.

Since θ has bounded velocity,

$$\forall r \leq t \quad \|\theta_r(x_0) - \theta_{r_1}(x_0)\| \leq C_v |r - r_1|.$$

By (4.10), since $\theta_{r_1}(x_0) \in (u_{r_1}^{ik})^{-1}(0)$, then $\theta_{r_1}(x_0) \in \mathcal{N}_{ik}(C_v r_1)$, then

$$\forall 0 \leq r \leq t, \quad \theta_r(x_0) \in \mathcal{N}_{ik}(2C_v t). \quad (4.11)$$

Upon reducing c by a factor 2, $\theta_r(x_0) \in \mathcal{N}_{ik}(s)$. We now claim that

$$\text{there exists } 0 \leq r_3 \leq t \text{ such that } \theta_{r_3}(x_0) \in \mathcal{N}_{ik}^*(s). \quad (4.12)$$

Indeed, if it is not the case, then for all r $V_r(\theta_r(x_0)) = V_r^k(\theta_r(x_0))$ and then $\theta_r(x_0) = \theta_r^k(x_0)$ and hence $u_r^{ik}(\theta_r(x_0))$ is a constant which is in contradiction with $u_{r_2}^{ik}(\theta_{r_2}(x_0)) \neq 0$ and $u_{r_1}^{ik}(\theta_{r_1}(x_0)) = 0$. Using the bounded velocity of θ , and (4.12) we conclude that $x = \theta_t(x_0) \in \mathcal{N}_{ik}^*(s + C_v t) \subset \mathcal{N}_{ik}^*(2s)$. Finally, using (4.11), we obtain

$$x \in \mathcal{N}_{ik}^*(2s) \cap \mathcal{N}_{ik}(C_v t) = \bigcup_{j \neq k} (\mathcal{N}_{ij}(2s) \cap \mathcal{N}_{ik}(C_v t)).$$

By hypothesis (Diff-2-b), the last set has volume bounded by Cst for some constant depending on C_v , the maximum velocity of θ^k and θ . Since C_v may be chosen independently of s when s is small enough, then C is independent of s and t . \square .

4.2.4.2 Proof of lemmas

We are now ready to tackle the proof of Lemmas 4.2.2 and 4.2.3 in this section.

Proof of Lemma 4.2.2 In this proof, the rate of convergence of $\frac{o(t)}{t}$ towards 0 depends on s (as s^{-1}). Let f in $L^\infty(\Omega \times \mathbb{R})$ with gradient in $L^1(\Omega \times \mathbb{R})$ and s small enough. For all $t \leq cs$, Proposition 4.2.3 asserts

$$\begin{aligned} \int_{\mathcal{L}_i(t)} f(x, t) dx &= \int_{\theta_t(\mathcal{L}_i(0))} f(x, t) dx + \mathcal{O}(st) \\ &= \int_{\mathcal{L}_i(0)} f(\theta_t(x), t) |det(J\theta_t(x))| dx + \mathcal{O}(st) \end{aligned}$$

Where $J\theta_t$ is the Jacobian matrix of θ_t .

Using $\theta_t(x) = x + tV_0(x) + o_{L^\infty}(t)$, we then have (see [68])

$$f(\theta_t(x)) = f(x, 0) + t\partial_t f(x, 0) + t\nabla_x f(x, 0) \cdot V_0(x) + o_{L^1}(t),$$

$$|det(J\theta_t(x))| = 1 + tdiv(V_0) + o_{L^\infty}(t),$$

where $o_{L^a}(t)$ is a time dependent function that, when divided by t goes towards zero in L^a norm as t goes to zero. The rate of convergence depends on the Lipschitz norm of V_t which scales as s^{-1} .

Then finally

$$\int_{\mathcal{L}_i(t)} f(x, t) dx - \int_{\mathcal{L}_i(0)} f(x, 0) dx = t\alpha_f(s) + o(t) + \mathcal{O}(st) \quad (4.13)$$

with $\alpha_f(s) = \int_{\mathcal{L}_i(0)} (\partial_t f(x, 0) + \nabla_x f(x, 0) \cdot V_0(x) + f(x, 0) \operatorname{div}(V_0)) dx$.

Recall that V_0 depends on s , hence α_f depends on s .

A Stokes formula yields

$$\alpha_f(s) = \int_{\mathcal{L}_i(0)} \partial_t f(x, 0) dx + \int_{\partial \mathcal{L}_i(0)} f(x, 0) V_0(x) \cdot n_i d\sigma$$

This formula is true for Lipschitz domain and $\mathcal{L}_i(0)$ is Lipschitz because each e^{ik} is Lipschitz as can be proven by a an implicit function theorem using (Diff-2-b).

Denote $Y(s) = \{x \in \partial \mathcal{L}_i(0) \cap \partial \mathcal{L}_k(0) \text{ s.t. } \zeta(\frac{\delta_k(x)}{s}) \neq 1\}$. Since

$$Y(s) \subset \bigcup_{j \neq k} (\mathcal{N}_{ij}(s) \cap e^{ik}),$$

we know $\lim_{s \rightarrow 0} \sigma(Y(s)) = 0$ by (Diff-2-c).

Since f is in $W^{1,1}$, its trace on e^{ik} is in $L^1(e^{ik})$ for the measure σ [24]. Hence, noticing that

$$\mathbb{1}_{\partial \mathcal{L}_i(0)} V_0(x) = \sum_{k \neq i} \mathbb{1}_{\partial \mathcal{L}_i(0) \cap \partial \mathcal{L}_k(0)} \zeta(\frac{\delta_k(x)}{s}) V_0^k(x),$$

the dominated convergence theorem asserts that $\alpha(s)$ converges as s goes to zero towards

$$\lim_{s \rightarrow 0} \alpha_f(s) := \alpha_f(0) = \int_{\mathcal{L}_i(0)} \partial_t f(x, 0) dx + \sum_{k \neq i} \int_{\partial \mathcal{L}_i(0) \cap \partial \mathcal{L}_k(0)} f(x, 0) V_0^k(x) \cdot n_i d\sigma.$$

Now denote,

$$r(t) = t^{-1} \left(\int_{\mathcal{L}_i(t)} f(x, t) dx - \int_{\mathcal{L}_i(0)} f(x, 0) dx \right).$$

Let t go to zero in (4.13), we have $\limsup_{0+} r(t) = \alpha_f(s) + \mathcal{O}(s)$ and $\liminf_{0+} r(t) = \alpha_f(s) + \mathcal{O}(s)$. Letting s goes to zero shows that $\lim_{0+} r(t)$ exists and is equal to $\alpha_f(0)$ which proves lemma 4.2.2.

Proof of lemma 4.2.3 The proof of this proposition owes so much to [80], proposition 3.2 that we even take the same notations. Consider the following partition of $\mathcal{L}_i(t) \cap \mathcal{L}_k(t)$:

$$\begin{aligned} A_t &= \left\{ \theta_t^k(x) \text{ s.t. } \theta_r^k(x) \in \mathcal{L}_i(r) \cap \mathcal{L}_k(r) \quad \forall r \in [0, s] \right\} \\ B_t &= \left\{ x \in \mathcal{L}_i(t) \cap \mathcal{L}_k(t) \text{ s.t. } x \notin A_t \right\} \end{aligned}$$

It is clear that for all $t \leq s$, we have

$$\xi(t) = \int_{A_t} f(x, t) d\sigma + \int_{B_t} f(x, t) d\sigma$$

In [80], in the first part of the proof of Proposition 3.2, the authors show that

$$\lim_{s \rightarrow 0^+} \int_{A_s} f(x, s) d\sigma = \int_{A_0} f(x, 0) d\sigma,$$

while actually controlling the convergence rate by the modulus of continuity of f . The reason is that θ_t^k is a bi-Lipschitz map between A_t and A_0 and that a change of variable allows to prove continuity. Note that no regularity assumption is made on the set A_t except that its $d - 1$ Hausdorff measure is bounded, which is exactly hypothesis (Cont-2). In order to prove that the sets B_t are small with respect to the measure $d\sigma$, we follow a slightly simpler and quicker path than [80] due to the fact that we use a stronger hypothesis in (Diff-2-c).

First if $x = \theta_t^k(x_0) \in B_t$, then there exists $r \in [0, s]$, such that :

$$\theta_r^k(x) \in \partial\Omega \cup \bigcup_{j \notin \{i, k\}} (u_r^{ij})^{-1}(0) \quad (4.14)$$

Indeed if $x \in B_t$ then $\theta_r^k(x)$ is in $\mathcal{L}_i(r) \cap \mathcal{L}_k(r)$ for $r = t$ and strictly outside this set for some $r = r_1$. Recalling that

$$\mathcal{L}_i(t) \cap \mathcal{L}_k(t) = \left\{ x \in \Omega \text{ s.t. } u_t^{ik}(x) = 0 \text{ and } u_t^{ij}(x) \leq 0 \text{ and } u_t^{kj}(x) \leq 0 \quad \forall j \notin \{i, k\} \right\},$$

and that θ_t^k preserves the level-set 0 of u_t^{ik} , then for some r , we must have by the intermediate value theorem either $\theta_r^k(x) \in \partial\Omega$ or $u_r^{ij}(\theta_r^k(x)) = 0$ or $u_r^{kj}(\theta_r^k(x)) = 0$. Finally if $u_r^{kj}(\theta_r^k(x)) = 0$, then implies that $u_r^{ij}(\theta_r^k(x)) = 0$, since $u_r^{ik}(\theta_r^k(x)) = 0$.

Suppose that we are in the case $u_r^{ij}(\theta_r^k(x_0)) = 0$ in (4.14), then $\theta_r^k(x_0)$ is in $(u_r^{ij})^{-1}(0)$, which, by finite velocity of θ^j , is at distance at most $C_v s$ of $e^{ij} = (u_0^{ij})^{-1}(0)$. By finite velocity of θ^k , x_0 is at distance at most $C_v s$ of $\theta_r^k(x_0)$, meaning that x_0 is at distance at most $2C_v s$ of e^{ij} . Since $u_0^{ik}(x_0) = 0$, we have that $x_0 \in C_0 := e^{ik} \cap \mathcal{N}_{ik}^*(2C_v s)$ which $d\sigma$ goes to zero as s goes to zero by (Diff-2-c). Finally

$$B_t \subset \theta_t^k(C_0),$$

since θ_t^k is a bi-Lipschitz map. Hence as s goes to zero, $d\sigma(B_t)$ goes to 0 and hence.

$$\lim_{s \rightarrow 0^+} \int_{B_t} f(x, t) d\sigma = 0 = \lim_{s \rightarrow 0^+} \int_{B_0} f(x, 0) d\sigma. \quad \square$$

4.2.5 Proof of the results of Section 4.2

The goal of this section is to prove the different results of Section 4.2. We begin by Proposition 4.2.1. Suppose first $z^i \mapsto c(z^i, x)$ is differentiable ν a.e. for all i and that ν is a positive Borelian measure of finite mass and rewrite \tilde{g} as

$$\tilde{g}(t) = \int_{\Omega} \psi_{\text{opt}}(z(t), \phi(t), x) d\nu(x)$$

where

$$\psi_{\text{opt}}(z, \phi, x) = \min_i \psi^i(z, \phi, x) \quad \text{with } \psi^i(z, \phi, x) = c(z^i, x) - \phi^i,$$

As the minimum of a finite number of differentiable functions, ψ_{opt} is measurable and is ν -a.e directionally derivable with formula

$$\psi_{\text{opt}}'(x) := \lim_{t \rightarrow 0^+} \frac{\psi_{\text{opt}}(z + td_z, \phi + td_\phi, x) - \psi_{\text{opt}}(z, \phi, x)}{t} = \min_{i \in \mathcal{M}(x)} \langle \nabla \psi^i(x), d \rangle,$$

with $d = (d_z, d_\phi)$ and ∇ is the gradient with respect to z and ϕ . Recall for that purpose that $\mathcal{M}(x)$ is exactly the argmin of $\psi^i(x)$. The function $(\psi_{\text{opt}})'$ is seen to be measurable when rewritten as :

$$\psi_{\text{opt}}'(x) = \sum_{A \subset \llbracket 1, n \rrbracket} \mathbb{1}_{\{\mathcal{M}^{-1}(A)\}}(x) \min_{i \in A} \langle \nabla \psi^i(x), d \rangle,$$

the set $\mathcal{M}^{-1}(A)$ being measurable since

$$(\mathcal{M}(x) = A) \Leftrightarrow (\psi_i(x) = \psi_{\text{opt}}(x) \quad \forall i \in A \text{ and } \psi_i(x) > \psi_{\text{opt}}(x) \quad \forall i \in A^c)$$

A standard dominated convergence theorem asserts that the directional derivative of \tilde{g} exists and is given by :

$$\tilde{g}' = \int_{\Omega} (\psi_{\text{opt}})'(x) d\nu(x)$$

and we retrieve

$$\tilde{g}' = \sum_{A \subset \llbracket 1, n \rrbracket} \int_{\mathcal{M}^{-1}(A)} \min_{i \in A} \langle \nabla \psi^i(x), d \rangle d\nu(x)$$

which is exactly the formula of Proposition 4.2.1. When one supposes that $\nu(\mathcal{M}^{-1}(A)) = 0$ as soon as the cardinal of A is strictly greater than 1, then \tilde{g}' is linear w.r.t d and hence differentiable. In this case, we have $\mathcal{M}^{-1}(\{i\}) = \mathcal{L}_i(z, \phi)$ up to a set of zero ν -measure, and hence

$$\nabla \tilde{g}(z, \phi) = \sum_{i=1}^n \int_{\mathcal{L}_i(z, \phi)} \nabla \psi_i(z, \phi, x) d\nu(x)$$

In order to prove the continuity of the gradient of \tilde{g} , we use the following technical lemma with $f = \nabla \psi_i(z, \phi, x)$.

Lemma 4.2.4. *Suppose that $\nu(\mathcal{M}^{-1}(A)) = 0$ if $\#(A) \geq 2$. If f is continuous with respect to z, ϕ for almost every x and if there exists $l \in L^1(\nu)$ such that $|f(z, \phi, x)| \leq l(x)$ ν -a.e. for all (z, ϕ) then*

$$F_i : (z, \phi) \mapsto \int_{\mathcal{L}_i(z, \phi)} f(z, \phi, x) d\nu$$

is continuous.

Proof of Lemma 4.2.4 First recall that

$$\mathcal{L}_i(z, \phi) = \{x \in \Omega \text{ s.t. } \psi^i(z, \phi, x) \leq \psi_{\text{opt}}(z, \phi, x)\}.$$

For a sequence (z_n, ϕ_n) that goes to (z, ϕ) , denote

$$h_n(x) = f(z_n, \phi_n, x) \mathbf{1}_{\psi^i(z_n, \phi_n, x) \leq \psi_{\text{opt}}(z_n, \phi_n, x)}(x) \mathbf{1}_{\Omega}(x),$$

and $h = \mathbf{1}_{\mathcal{L}_i} f$. Then

$$F_i(z_n, \phi_n) = \int_{\Omega} h_n d\nu, \quad F_i(z, \phi) = \int_{\Omega} h d\nu.$$

Moreover $h_n \leq l$ for all n .

If x is such that $i \notin \mathcal{M}(x)$, that is $\psi^i(z, \phi, x) > \psi_{\text{opt}}(z, \phi, x)$, then $h_n(x)$ converges to $h(x)$. If x is such that $\mathcal{M}(x) = \{i\}$, then by continuity of $\psi^j(z, \phi, x)$ for all $j \neq i$, $\psi^i(z_n, \phi_n, x) = \psi_{\text{opt}}(z_n, \phi_n, x)$ for n sufficiently large, hence $h_n(x)$ converges to $h(x)$. Then h_n converges to $h = \mathbf{1}_{\mathcal{L}_i} f$ except possibly on the sets where $\mathcal{M}^{-1}(A)$ is of cardinal greater or equal than 2, which is, by hypothesis, of zero ν -measure.

Since $h_n \leq l$ for all n , a dominated convergence theorem ensures the continuity of the integral with respect to (z, ϕ) . \square

Proof of Theorem 4.2.1 The proof of Theorem 4.2.1 is straightforward. We apply Lemma 4.2.2 to $f(z, x) = \partial_z c(z^i, x)m(x)$ or $f(z, x) = m(x)$, where m is the density of ν . Then we apply Lemma 4.2.3 to the formula of the second order derivative in order to prove second order continuity. We obtain the following formula for the second order derivatives of g , taken at point $(\phi, z) = (\phi_0, z_0)$:

$$\begin{aligned} \frac{\partial^2 \tilde{g}}{\partial a \partial b} &= \sum_{i \neq j} \int_{\partial \mathcal{L}_i \cap \partial \mathcal{L}_j} \frac{(\partial_a c(z^j, x) - \partial_a \phi^j - \partial_a c(z^i, x) + \partial_a \phi^i)(\partial_b c(z^i, x) - \partial_b \phi^i)}{\|\nabla_x c(z^i, x) - \nabla_x c(z^j, x)\|} m(x) d\sigma \\ &+ \sum_i \int_{\mathcal{L}_i} \frac{\partial^2 c}{\partial a \partial b}(z^i, x) d\nu(x), \end{aligned}$$

where σ is the $d-1$ Hausdorff measure, m is the density of ν and a or b have to be replaced by the coordinates of ϕ or of one of the point z^k . Replacing the a and b , yields the formulas stated in Theorem 4.2.1.

4.3 Numerical experiments

In this section we test a second order algorithm for the 2-Wasserstein distance, when c is the Euclidean cost. Two problems will be solved : *Blue Noise* and *Stippling*. In both cases, we optimize a measure μ of the form $\mu(z, m) = \sum_{i=1}^n m^i \delta_{z^i}$, so that the 2-Wasserstein distance $W_2(\mu(z, m), \nu)$ is minimal. The cost function for the Stippling problem is referred to as G_S , the one for the Blue Noise as G_B . We denote by $G_{B/S}$ the cost function when no difference is made between the two problems.

There are many applications for the *Blue Noise* and *Stippling* problems, from data compression in computer graphics to clustering in data science described in [43], or more recently in anisotropic meshing [86]. Note also that

these problems fall within the scope of optimum quantization, that is finding a set of supporting points that approximates a probability distribution, see [62, 65] and the references inside.

- *Blue Noise* : Here the weights m^i are fixed. Hence the functional to minimize reads as :

$$\inf_{z \in \mathbb{R}^{nd}} G_B(z) \text{ with } G_B(z) = W_2(\mu(z, m), \nu) = \max_{\phi} g(\phi, z, m). \quad (4.15)$$

- *Stippling* : This problem consists in optimizing in m and in z simultaneously :

$$\inf_{z \in \mathbb{R}^{nd}} \inf_{m \in \Delta_n} W_2(\mu(z, m), \nu),$$

where Δ_n is the canonical simplex.

The *Stippling* problem is actually easier than the *Blue Noise* problem. Following the discussion of Section 4.1.2, optimizing the mass amounts to set $\phi = 0$ and $\tilde{m}_i = \nu(\mathcal{V}_i)$:

$$\inf_{z \in \mathbb{R}^{nd}} G_S(z) \text{ with } G_S(z) = W_2(\mu(z, \tilde{m}), \nu) = g(0, z, \tilde{m}), \quad (4.16)$$

where G_S is the Voronoi energy defined in (4.6). Hence no optimization procedure is required in ϕ and \tilde{m} is merely given by computing the ν -mass of each Voronoi cells.

Note that $\partial_z G_{B/S}$, the differential of $G_{B/S}$ with respect to z is equal to :

$$\partial_z G_B = \partial_z g \quad \text{and} \quad \partial_z G_S = \partial_z g + \partial_m g \partial_z \tilde{m}$$

since $\partial_m g = 0$ when $\phi = 0$, we obtain

$$\partial_z G_{B/S} = \partial_z g \quad (4.17)$$

Formulas Recall that in the Euclidean case, the formulas for g boils down to

$$\begin{aligned} \frac{\partial g}{\partial z^i} &= M^i (z^i - \bar{z}^i) \\ \frac{\partial^2 g}{\partial z^i \partial z^j} &= \int_{\partial \mathcal{L}_i \cap \partial \mathcal{L}_j} \frac{(z^j - x)(z^i - x)^T m(x)}{\|z^i - z^j\|} d\sigma \text{ if } i \neq j \\ \frac{\partial^2 g}{\partial (z^i)^2} &= M^i - \sum_{i \neq j} \frac{\partial^2 g}{\partial z^i \partial z^j} \end{aligned}$$

where $M^i = \int_{\mathcal{L}_i} d\nu$ is the mass of the i^{th} Laguerre cell and $\bar{z}^i = \int_{\mathcal{L}_i} x d\nu / M^i$ is its barycenter.

4.3.1 Lloyd's algorithm

Algorithm 2 Lloyd's algorithm with Wolfe stabilization.

- 1: **Inputs :**
 - 2: Initial guess z^0
 - 3: target measure ν
 - 4: **Outputs :**
 - 5: An approximation of the solution of (4.15).
 - 6: **while** Until convergence **do**
 - 7: Compute $\phi_{\text{opt}}(z^k)$.
 - 8: Compute \bar{z}_i^k the barycenter of the i^{th} Laguerre cell $\mathcal{L}_i(z^k, \phi_{\text{opt}}(z^k))$.
 - 9: Set $d^k = \bar{z}_i^k - z_i^k$
 - 10: Set $\sigma^k = 1$ and $z_i^{k+1} = z_i^k + \sigma^k d^k$
 - 11: **while** The Wolfe conditions are not fulfilled **do**
 - 12: $\sigma^k = \sigma^k/2$ and $z_i^{k+1} = z_i^k + \sigma^k d^k$.
 - 13: **end while**
 - 14: $k = k + 1$
 - 15: **end while**
 - 16: Return z^k .
-

The gradient algorithm for computing the *Blue Noise* (resp. the *Stippling* problem) is to move each point in the direction of the barycenter of its Laguerre cell (resp. Voronoi cell). Taking the diagonal metric given by the mass of the cells $M = (M^i)_{i=1..n}$ (which is a decent approximation of the Hessian), yields the following formula for the gradient of $G_{B/S}$

$$\langle a, b \rangle_M = \sum_i M^i a^i b^i \implies \nabla G_{B/S}(z_k) = z^i - \bar{z}^i. \quad (4.18)$$

A fixed step gradient Algorithm with step 1 is to set each point z^i exactly at the location of the barycenter \bar{z}^i . This algorithm is well known as a Lloyd-like or a relaxation algorithm [90, 43]. An improvement of Lloyd's algorithm is to ensure a Wolfe step condition [13].

$$G_{B/S}(z_{k+1}) < G_{B/S}(z_k) + 10^{-4} \langle \nabla G_{B/S}(z_k), z_{k+1} - z_k \rangle_M \quad (\text{Wolfe})$$

This naturally leads to algorithm 2. The only difference between the *Stippling* and *Blue Noise* problems lies in the choice of $\phi_{\text{opt}}(z)$. It is chosen equal to 0 in the *Stippling* problem and to $\text{argmax}_\phi g$ in the *Blue Noise* problem.

Numerical experiment shows that it is not necessary to check for Wolfe's second condition which ensures that the step is not too small, indeed Lloyd's algorithm (and Newton's algorithm) have a natural step $\sigma^k = 1$.

4.3.2 Newton's Algorithm

The second algorithm is a Newton algorithm. Denoting by H the Hessian, in the *Stippling* case, we have by (4.17) :

$$H_{zz}G_S = H_{zz}g \quad (4.19)$$

The computation of $H_{zz}G_B$ for the *Blue Noise* case is more involved. A chain rule yields

$$H_{zz}G_B = H_{zz}g + H_{z\phi}g \cdot \nabla_z \phi_{\text{opt}}$$

The existence of $\nabla_z \phi_{\text{opt}}$ is given by an implicit function theorem, from

$$\nabla_{\phi}g(\phi_{\text{opt}}(z), z, m) = 0.$$

Differentiating the above equation with respect to z and applying the chain rule, we get

$$H_{z\phi}g + H_{\phi\phi}g \nabla_z \phi_{\text{opt}} = 0$$

and hence

$$H_{zz}G_B = H_{zz}g - H_{z\phi}g(H_{\phi\phi}g)^{-1}H_{\phi z}g.$$

The implicit function theorem that proves existence of $\nabla_z \phi_{\text{opt}}$ requires the matrix $H_{\phi\phi}g$ to be invertible. Note that constant ϕ are always part of the kernel of $H_{\phi\phi}g$ since g is invariant under the addition of constants to ϕ . Upon supposing that ϕ has zero average, the invertibility of $H_{\phi\phi}g$ is verified throughout the optimization procedure. In theory, $H_{\phi\phi}g$ can only be proven to be invertible generically [96, Theorem 18].

Once the Hessian is computed, the Newton algorithm with preconditioning by the matrix M amounts to changing in Algorithm 2 the descent direction d^k by a solution to the linear problem

$$AM^{1/2}d^k = -M^{1/2}\nabla G_{B/S}(z^k) \quad \text{with } A = (M^{-1/2}H_{zz}G_{B/S}M^{-1/2}), \quad (4.20)$$

and $\nabla G_{B/S}(z^k)$ is defined in (4.18) as the gradient with respect to the metric M . Newton's algorithm fails if the Hessian is not positive definite, hence we

propose a work-around based on the conjugate gradient method on the system (4.20). Recall that conjugate gradient method solve exactly the problem in the Krylov space and that the residues of the conjugate gradient method form an orthogonal basis of this Krylov space, hence are equal (up to a normalization procedure) to the Lanczos basis. Denote π_n the projection on the Krylov space at iteration n , the matrix $\pi_n A \pi_n$ is tridiagonal in the Lanczos basis hence the computation of its determinant is a trivial recurrence [115]. By monitoring the sign of the determinant throughout iterations one checks the positiveness of the matrix. The conjugate gradient algorithm is stopped whenever the matrix A stops being positive definite. The descent direction is then given by

$$(\pi_n A \pi_n) M^{1/2} d^k = -\pi_n M^{1/2} \nabla G_{B/S}(z^k),$$

By convention for $n = 0$, we solve $d^k = -\nabla G_{B/S}(z^k)$. If A is positive, then the problem (4.20) is solved exactly.

4.3.3 Other considerations

The computation of ϕ^* in the *Blue Noise* problem is a standard unconstrained concave maximization procedure with knowledge of second order derivatives. In order to compute ϕ_{opt} in a robust manner, we settled on a Levenberg-Marquardt type algorithm : denoting $H(\sigma) = H_{\phi\phi}g - \frac{1}{\sigma}Id$, we take as descent direction $-H(\sigma)^{-1}\nabla g(\phi)$, where σ is reduced until Wolfe's first order conditions are met. In the *Stippling* problem, the computation of $\phi^* = 0$ is trivial.

The Laguerre tessellation is computed by CGAL [128]. All the tests were performed using a standard Lena image as background measure ν which has been discretized as bilinear by pixel ($Q1$ finite element method). In the *Blue Noise* problem, the mass m is constrained to be equal to $\frac{1}{n}$ for all Dirac masses.

4.3.4 Numerical results

4.3.5 Direct comparison of the algorithms

For the first example, we search the optimal positions of the Dirac masses for either the *Blue Noise* or *Stippling* problem. Three methods are benchmarked, the Gradient method (Lloyd-like method), the Newton method discussed

in the previous section and a *LBFSGS* method with the memory of the 8 previous iterations. Tests are performed for 1K and 10K uniformly drawn points. The evolution of the cost functions and the L_2 norms of the gradient are displayed throughout iterations. Figure 4.3 displays the results obtained for the *Stippling* problem whereas Figure 4.4 displays the results for the *Blue Noise* problem. The tests are made with the same set of initial points, a different initialization leads to comparable results.

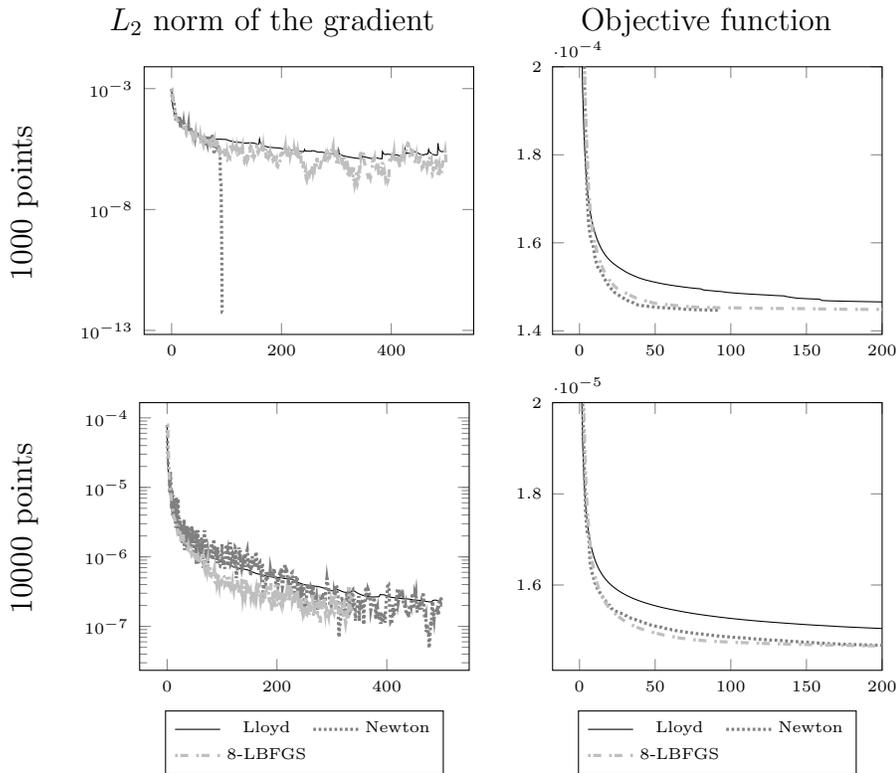


FIGURE 4.3 – *Stippling* problem with 1Kpts (Top) and 10 Kpts (Bottom). Left : norm of the gradient, Right : evolution of the cost function.

Our interpretation of Figure 4.3 is the following : in the 1K points problem, the best methods for finding critical points and minimum are, by decreasing order, Newton, 8-BFGS and Lloyd, which is coherent with theory. For the 10K points problems, the three different methods seem equivalent. Our interpretation of the 10K points behavior is the combination of two factors. First we believe that an augmentation of the number of points reduces

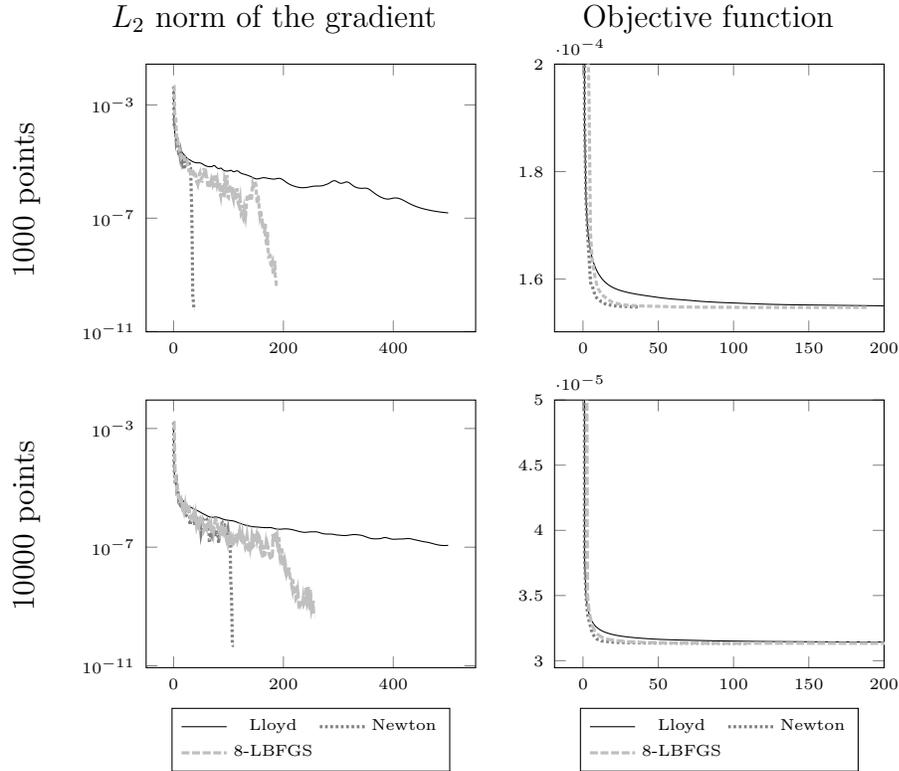


FIGURE 4.4 – *Blue Noise* problem with 1Kpts (Top) and 10 Kpts (Bottom). Left : norm of the gradient, Right : evolution of the cost function.

the basin of attraction of local minimum. Indeed, in our test the Newton method failed to attain locally convex points (the Hessian always had a negative eigenvalue throughout iterations). The second effect of the augmentation of the number of points is that numerical errors trickle down the algorithm, eventually preventing the Newton method to accurately find the minimum.

As a conclusion, we find that using second order derivative information in computing centroidal Voronoi tessellation (*Stippling* problem) is useful for a small number of points, which renders the application range of this method quite limited. Note that very similar tests have already been performed in [89]. The main conclusions of the tests in Figure 4.3 and Figure 4.4 is that the basin of attraction of the Newton method for the *Blue Noise* problem seems bigger than the one of the *Stippling* problem. Hence a second order method for the *Blue Noise* problem is of interest as the number of points

rises.

4.3.6 Adding a point

In order to exhibit the helpfulness of second order method for *Stippling*, we build an example where the classic gradient algorithm fails to converge. Empirically the main drawback of Lloyd algorithm is its lack of *globalisation*. Suppose one has optimized the position of n Dirac masses for the *Blue Noise* or *Stippling* problem and that one adds one mass at some random location and wants to optimize the position of the $n + 1$ Dirac masses. Lloyd's algorithm for the *Stippling* problem will converge slowly due to the fact that the new point will modify the Voronoi cells of its neighbours only, whereas the *Blue Noise* functional is global and every Laguerre cell will be modified at the first iteration. Hence Lloyd's algorithm for the *Stippling* problem has to wait for the information to propagate through each Voronoi cell, like the peeling of an onion, one layer at each iteration. The advantages of the second order method can then be seen, since the Hessian encodes the connectivity and propagates instantly the information. This effect should be less important for the *Blue Noise* case where information is propagated instantly. In Figure 4.5, we exhibit this effect for the *Blue Noise* and *Stippling* problem. We optimize with 1K pts with a second order method and then test either Lloyd's or Newton's method.

Our interpretation of Figure 4.5 lies mainly in the observation of the cost function. Newton's method converges for the *Blue noise* or *Stippling* case, where Lloyd's method convergence rate towards a critical point is way slower. Note however that the objective function for the *Stippling* problem decreases very slowly for Lloyd's method compared to the Newton method whereas the decrease of the objective function for the *Blue Noise* is comparable. We interpret this result as the "peeling layers" effect, only seen in the *Stippling* problem, described earlier.

4.4 Conclusion

In this paper we have studied the conditions under which second order differentiability of the semi-discrete optimal transport with respect to position of the Dirac masses holds for generic cost function c . This result encompasses the second order differentiability of the energy of a Voronoi diagram. We have

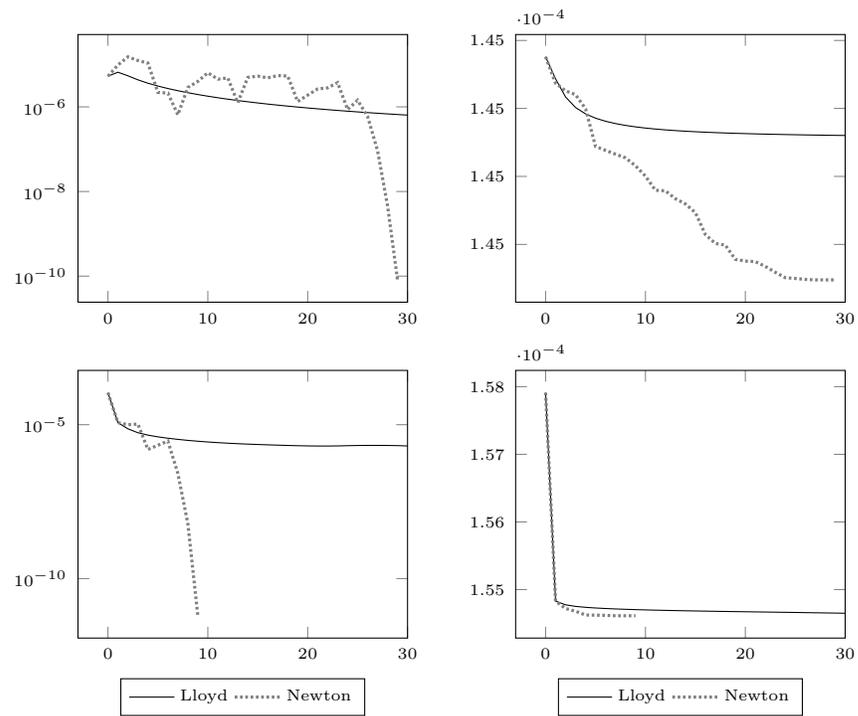


FIGURE 4.5 – Evolution of the cost function (Right) and the norm of the gradient (Left) for the *Stippling* problem (Top) and the *Blue Noise* problem (Bottom) when one point is added.

numerically implemented the second order procedure for both the *Blue Noise* and *Stippling* problem. In the *Stippling* problem, the numerical applications are limited by arithmetic precision and small basins of attraction. The *Blue Noise* problem is less sensitive to these effects. An interpretation of this fact is that the *Blue Noise* problem is global, a change in the position of a mass as an effect on the whole set of masses, whereas in the *Stippling* problem, a mass only sees its direct neighbours. The *Blue Noise* problem is then a more stable problem than the *Stippling* one.

It is then of the highest interest to understand the smallness of the basins of attraction and the disposition of local minima for the two corresponding problem. It is also the aim of future work to understand optimal transportation between Dirac masses and non-regular background measures (say measures supported by curves) and the corresponding *Blue Noise* problem. Such an application requires to differentiate the semi-discrete optimal transport with respect to parameters that describe the underlying background measure ν .

Chapter 5

Optimal Transport Approximation of 2-Dimensional Measures

Contents

5.1	Introduction	79
5.1.1	Contributions	80
5.1.2	Related works	80
5.1.3	Paper outline	83
5.2	The minimization framework	83
5.2.1	Discretization	83
5.2.2	Overall algorithm	84
5.3	Computing the Wasserstein distance F : ψ-step	85
5.3.1	Semi-discrete optimal transport	85
5.3.2	Solving the dual problem	87
5.3.3	Numerical integration	91
5.4	Optimizing the weights and the positions : w and x steps	92
5.4.1	Computing the optimal weights	92
5.4.2	Gradient $\nabla_{\mathbf{x}}F$ and the metric Σ_k	93
5.5	Links with other models	94
5.5.1	Special cases of the framework	94
5.5.2	Comparison with electrostatic halftoning	95

5.6	Projections on curves spaces	99
5.6.1	Discrete curves	99
5.6.2	Numerical projectors	102
5.6.3	Numerical examples	104
5.6.4	Multiresolution implementation	104
5.7	Applications	105
5.7.1	Non Photorealistic Rendering with curves	105
5.7.2	Path planning	108
5.8	Theoretical convergence of Algorithm 3	110
5.9	Outlook	111



source code


<https://github.com/lebrat/semiDiscreteCurvling>

Abstract

We propose a fast and scalable algorithm to project a given density on a set of structured measures defined over a compact 2D domain. The measures can be discrete or supported on curves for instance. The proposed principle and algorithm are a natural generalization of previous results revolving around the generation of blue-noise point distributions, such as Lloyd’s algorithm or more advanced techniques based on power diagrams. We analyze the convergence properties and propose new approaches to accelerate the generation of point distributions. We also design new algorithms to project curves onto spaces of curves with bounded length and curvature or speed and acceleration. We illustrate the algorithm’s interest through applications in advanced sampling theory, nonphotorealistic rendering and path planning.

This chapter was published in

F. de Gournay, J. Kahn, L. Lebrat, and P. Weiss

Optimal Transport Approximation of 2-Dimensional Measures. In *SIAM Journal on Imaging Sciences*, 12(2) : 762 :787, 2019.

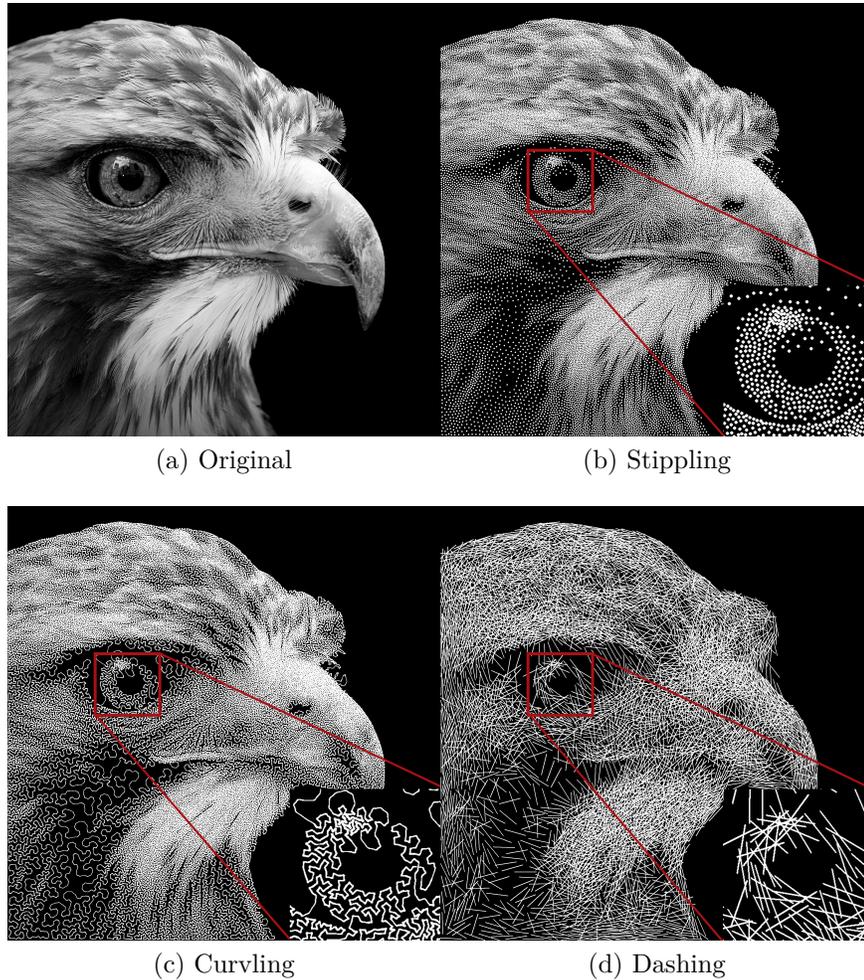


FIGURE 5.1 – Approximating an image with a measure supported on points (stippling, 100k, 202"), curve (curvling, 100k, 313") or segments (dashing, 33k, 237"). In each case, the iterative algorithm starts from a set of points drawn uniformly at random.

5.1 Introduction

The aim of this paper is to approximate a target measure μ with probability density function $\rho : \Omega \rightarrow \mathbb{R}_+$ with probability measures possessing some

structure. This problem arises in a large variety of fields including finance [107], computer graphics [122], sampling theory [21], and optimal facility location [57]. An example in nonphotorealistic rendering is shown in Figure 5.1, where the target image in Fig. 5.1a is approximated by an atomic measure in Fig. 5.1b, by a smooth curve in Fig. 5.1c and by a set of segments in Fig. 5.1d. Given a set of admissible measures \mathcal{M} (i.e. atomic measures, measures supported on smooth curves or segments), the best approximation problem can be expressed as follows :

$$\min_{\nu \in \mathcal{M}} D(\nu, \mu), \quad (5.1)$$

where D is a distance between measures.

5.1.1 Contributions

Our main contributions in this article are listed below.

- Develop a few original applications for the proposed algorithm.
- Develop a fast numerical algorithm to minimize problem (5.1), when D is the W_2 transportation distance and $\Omega = [0, 1]^2$.
- Show its connections to existing methods such as Lloyd’s algorithm [90] and optimal transport halftoning [34].
- Provide some theoretical convergence guarantees for the computation of the optimal semi-discrete transportation plan, especially for complicated point configurations and densities, for which an analysis was still lacking.
- Design algorithms specific to the case where the space of admissible measures \mathcal{M} consists of measures supported on curves with geometric constraints (e.g. fixed length and bounded curvature).
- Generate a gallery of results to show the versatility of the approach.

In the next section, we put our main contributions in perspective.

5.1.2 Related works

5.1.2.1 Projections on measure spaces

To the best of our knowledge, the generic problem (5.1) was first proposed in [26] with a distance D constructed through a convolution kernel. Similar problems were considered earlier, with spaces of measures restricted to a fixed support for instance [92], but not with the same level of generality.

Formulation (5.1) covers a large number of applications that are often not formulated explicitly as optimization problems. We review a few of them below.

Finitely supported measures Many approaches have been developed for cases when \mathcal{M} is the set of uniform finitely supported measures

$$\mathcal{M}_{f,n} = \left\{ \nu(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}[i]}, \mathbf{x} \in \Omega^n \right\}, \quad (5.2)$$

where n is the support cardinality or the set of atomic measures defined by

$$\mathcal{M}_{a,n} = \left\{ \nu(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^n \mathbf{w}[i] \delta_{\mathbf{x}[i]}, \mathbf{x} \in \Omega^n, \mathbf{w} \in \Delta_{n-1} \right\}, \quad (5.3)$$

where $\Delta_{n-1} = \{\sum_{i=1}^n \mathbf{w}[i] = 1, \mathbf{w}[i] \geq 0 \forall i\}$ is the canonical simplex.

For these finitely supported measure sets, solving problem (5.1) yields nice stippling results, which is the process of approximating an image with a finite set of dots (see Fig. 5.1b). This problem has a long history and a large number of methods were designed to find dots locations and radii that minimize visual artifacts due to discretization [52, 90, 131, 8]. Lloyd’s algorithm is among the most popular. We will see later that this algorithm is a solver of (5.1), with $\mathcal{M} = \mathcal{M}_{a,n}$. Lately, explicit variational approaches [118, 34] have been developed. The work of de Goes *et al* [34] is closely related to our paper since they propose solving (5.1), where D is the W_2 transportation distance and $\mathcal{M} = \mathcal{M}_{f,n}$. This sole problem is by no means limited to stippling and it is hard to provide a comprehensive list of applications. A few of them are listed in the introduction of [138].

Best approximation with curves Another problem that is met frequently is to approximate a density by a curve. This can be used for non-photorealistic rendering of images or sculptures [78, 2]. It can also be used to design trajectories of the nozzle of 3D printers [28]. It was also used for the generation of sampling schemes [21].

Apart from the last application, this problem is usually solved with methods that are not clearly expressed as an optimization problem.

Best approximation with arbitrary objects Problem (5.1) encompasses many other applications such as the optimization of networks [57], texture rendering or nonphotorealistic rendering [69, 70, 117, 79, 41], and sampling theory [22].

Overall, this paper unifies many problems that are often considered as distinct with specific methods.

5.1.2.2 Numerical optimal transport

In order to quantify the distance between the two measures, we use transportation distances [99, 76, 132]. In our work, we are interested mostly in the semi-discrete setting, where one measure is a density and the other is discrete. In this setting, the most intuitive way to introduce this distance is via Monge’s transportation plan and allocation problems. Given an atomic measure $\nu \in \mathcal{M}_{a,n}$ and a measure μ with density, a transport plan $T \in \mathcal{T}(\mathbf{x}, \mathbf{w})$ is a mapping $T : \Omega \rightarrow \{\mathbf{x}[1], \dots, \mathbf{x}[n]\}$ such that $\forall 1 \leq i \leq n, \mu(T^{-1}(\mathbf{x}[i])) = \mathbf{w}[i]$. In words, the mass at any point $x \in \Omega$ is transported to point $T(x)$. In this setting, the W_2 transportation distance is defined by

$$W_2^2(\mu, \nu) = \inf_{T \in \mathcal{T}(\mathbf{x}, \mathbf{w})} \int_{\Omega} \|x - T(x)\|_2^2 d\mu(x), \quad (5.4)$$

and the minimizing mapping T describes the optimal way to transfer μ to ν .

Computing the transport plan T and the distance W_2 is a challenging optimization problem. In the semi-discrete setting, the paper [7] provided an efficient method based on a “power diagram” or “Laguerre diagram”. This framework was recently further improved and analyzed in [34, 94, 85, 80]. The idea is to optimize a concave cost function with second-order algorithms. We will make use of those results in this paper and improve them by stabilizing them while keeping the second-order information.

5.1.2.3 Numerical projections on curve spaces

Projecting curves on admissible sets is a building block for many algorithms. For instance, mobile robots are subject to kinematic constraints (speed and acceleration), while steel wire sculptures have geometric constraints (length and curvature).

While the projection on kinematic constraints is quite easy, due to convexity of the underlying set [27], we believe that this is the first time projectors on

sets defined through intrinsic geometry have been designed. Similar ideas have been explored in the past. For instance, curve shortening with mean curvature motion [48] is a long-studied problem with multiple applications in computer graphics and image processing [139, 98, 124]. The proposed algorithms allow exploration of new problems such as curve lengthening with curvature constraints.

5.1.3 Paper outline

The rest of the paper is organized as follows. We first outline the overarching algorithm in Section 5.2. In Sections 5.3 and 5.4, we describe more precisely and study the theoretical guarantees of the algorithms used respectively for computing the Wasserstein distance, and for optimising the positions and weights of the points. We describe the relationships with previous models in Section 5.5. The algorithms in Sections 5.3 and 5.4 are enough for, say, halftoning, but do not handle constraints on the points. In Section 5.6, we add those constraints and design algorithms to make projections onto curve spaces with bounded speed and acceleration, or bounded length and curvature. Finally, some application examples and results are shown in Section 5.7.

5.2 The minimization framework

In this section, we show how to numerically solve the best approximation problem :

$$\inf_{\nu \in \mathcal{M}} W_2^2(\nu, \mu), \quad (5.5)$$

where \mathcal{M} is an arbitrary set of measures supported on $\Omega = [0, 1]^2$.

5.2.1 Discretization

Problem (5.5) is infinite-dimensional and first needs to be discretized to be solved using a computer. We propose approximating \mathcal{M} by a subset $\mathcal{M}_n \subseteq \mathcal{M}_{a,n}$ of the atomic measures with n atoms. The idea is to construct \mathcal{M}_n as

$$\mathcal{M}_n = \{\nu(\mathbf{x}, \mathbf{w}), \mathbf{x} \in \mathbf{X}_n, \mathbf{w} \in \mathbf{W}_n\}, \quad (5.6)$$

where the mapping $\nu : (\Omega^n \times \Delta_{n-1}) \rightarrow \mathcal{M}_{a,n}$ is defined by

$$\nu(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^n \mathbf{w}[i] \delta_{\mathbf{x}[i]}. \quad (5.7)$$

The constraint set $\mathbf{X}_n \subseteq \Omega^n$ describes interactions between points and the set $\mathbf{W}_n \subseteq \Delta_{n-1}$ describes the admissible weights.

We have shown in [26] that for any subset \mathcal{M} of the probability measures, it is possible to construct a sequence of approximation spaces $(\mathcal{M}_n)_{n \in \mathbb{N}}$ of the type (5.6), such that the solution sequence $(\nu_n^*)_{n \in \mathbb{N}}$ of the discretized problem

$$\inf_{\nu \in \mathcal{M}_n} W_2^2(\nu, \mu), \quad (5.8)$$

converges weakly along a subsequence to a global minimizer ν^* of the original problem (5.5). Let us give a simple example : assume that \mathcal{M} is a set of pushforward measures of curves parameterized by a 1-Lipschitz function on $[0, 1]$. This curve can be discretized by a sum of n Dirac masses with a distance between consecutive samples bounded by $1/n$. It can then be shown that this space \mathcal{M}_n approximates \mathcal{M} well, in the sense that each element of \mathcal{M}_n can be approximated with a distance $O(1/n)$ by an element in \mathcal{M} and vice versa [26]. We will show explicit constructions of more complicated constraint sets \mathbf{X}_n and \mathbf{W}_n for measures supported on curves in Section 5.6.

The discretized problem (5.8) can now be rewritten in a form convenient for numerical optimization :

$$\min_{\mathbf{x} \in \mathbf{X}, \mathbf{w} \in \mathbf{W}} F(\mathbf{x}, \mathbf{w}), \quad (5.9)$$

where we dropped the index n to simplify the presentation and where

$$F(\mathbf{x}, \mathbf{w}) = \frac{1}{2} W_2^2(\nu(\mathbf{x}, \mathbf{w}), \mu). \quad (5.10)$$

5.2.2 Overall algorithm

In order to solve (5.9), we propose using an alternating minimization algorithm : the problem is minimized alternately in \mathbf{x} with one iteration of a variable metric projected gradient descent and then in \mathbf{w} with a direct method. Algorithm 3 describes the procedure in detail.

A few remarks are in order. First notice that we are using a variable metric descent algorithm with a metric $\Sigma_k \succ 0$. Hence, we need to use a projector defined in this metric by :

$$\begin{aligned} \Pi_{\mathbf{X}}^{\Sigma_k}(\mathbf{x}_0) &:= \underset{\mathbf{x} \in \mathbf{X}}{\text{Argmin}} \|\mathbf{x} - \mathbf{x}_0\|_{\Sigma_k}^2 \quad \text{with} \\ \|\mathbf{x} - \mathbf{x}_0\|_{\Sigma_k}^2 &= \langle \Sigma_k(\mathbf{x} - \mathbf{x}_0), (\mathbf{x} - \mathbf{x}_0) \rangle. \end{aligned}$$

Second, notice that \mathbf{X} may be nonconvex. Hence, the projector $\Pi_{\mathbf{X}}^{\Sigma_k}$ on \mathbf{X} might be a point-to-set mapping. In the \mathbf{x} -step, the usual sign $=$ is therefore replaced by \in .

Below are listed five major difficulties in implementing this algorithm :

ψ step : How to compute efficiently $F(\mathbf{x}, \mathbf{w})$?

\mathbf{w} step : How to compute $\underset{\mathbf{w} \in \mathbf{W}}{\text{argmin}} F(\mathbf{x}, \mathbf{w})$?

\mathbf{x} step : How to compute the gradients $\nabla_{\mathbf{x}} F$ and the metric Σ_k ?

Π step : How to implement the projector $\Pi_{\mathbf{X}}^{\Sigma_k}$?

Generally : How to accelerate the convergence given the specific problem structure ?

The next four sections provide answers to these questions.

Note that if there are no constraints like in halftoning or stippling, there is no projection and the Π -step is trivial : $\mathbf{x}_{k+1} = \mathbf{y}_{k+1}$.

5.3 Computing the Wassertein distance F : ψ -step

5.3.1 Semi-discrete optimal transport

In this section, we review the main existing results about semi-discrete optimal transport and explain how it can be computed. Finally, we provide novel computation algorithms that are more efficient and robust than existing approaches. We work under the following hypotheses.

Assumption 5.3.1.

- *The space Ω is a compact convex polyhedron, typically the hypercube.*
- *μ is an absolutely continuous probability density function w.r.t. the Lebesgue measure.*

Algorithm 3 Alternating projected gradient descent to minimize (5.1).

Require: Oracle that computes F ▷ ψ -step.

Require: Projectors $\Pi_{\mathbf{X}}$ on \mathbf{X} .

1: **Inputs :**

2: Initial guess \mathbf{x}_0

3: Target measure μ

4: Number of iterations Nit .

5: **Outputs :**

6: An approximation $(\hat{\mathbf{x}}, \hat{\mathbf{w}})$ of the solution of (5.1).

7: **for** $k = 0$ to $Nit - 1$ **do**

8: $\mathbf{w}_{k+1} = \underset{\mathbf{w} \in \mathbf{W}}{\operatorname{argmin}}(F(\mathbf{x}_k, \mathbf{w}))$ ▷ **w**-step

9: Choose a positive definite matrix Σ_k , a step s_k .

10: $\mathbf{y}_{k+1} = \mathbf{x}_k - s_k \Sigma_k^{-1} \nabla_{\mathbf{x}} F(\mathbf{x}_k, \mathbf{w}_{k+1})$. ▷ **x**-step

11: $\mathbf{x}_{k+1} \in \Pi_{\mathbf{X}^k}^{\Sigma_k}(\mathbf{y}_{k+1})$ ▷ **Π**-step

12: **end for**

13: Set $\hat{\mathbf{x}} = \mathbf{x}_{Nit}$ and $\hat{\mathbf{w}} = \mathbf{w}_{Nit}$.

— ν is an atomic probability measure supported on n points.

Let us begin with a theorem stating the uniqueness of the optimal transport plan, which is a special case of Theorem 10.41 in the book [133].

Theorem 5.3.1. *Under Assumption 5.3.1, there is a unique optimal transportation plan μ -a.e. T^* for the solution of problem (5.4).*

Before further describing the structure of the optimal transportation plan, let us introduce a fundamental tool from computational geometry [6].

Definition 5.3.0.1 (Laguerre diagram). *Let $\mathbf{x} \in \Omega^n$ denote a set of point locations and $\boldsymbol{\psi} \in \mathbb{R}^n$ denote a weight vector. The Laguerre cell \mathcal{L}_i is a closed convex polygon set defined as*

$$\mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x}) = \{x \in \Omega, \forall 1 \leq j \leq n, j \neq i, \|x - \mathbf{x}[i]\|_2^2 - \boldsymbol{\psi}[i] \leq \|x - \mathbf{x}[j]\|_2^2 - \boldsymbol{\psi}[j]\}. \quad (5.11)$$

The Laguerre diagram generalizes the Voronoi diagram, since the latter is obtained by taking $\boldsymbol{\psi} = 0$ in (5.11).

The set of Laguerre cells partitions Ω into polyhedral pieces. It can be computed in $O(n \log(n))$ operations for points located in a plane [6]. In our

numerical experiments, we make use of the CGAL library to compute them [128]. We are now ready to describe the structure of the optimal transportation plan T^* , see [56, Example 1.9].

Theorem 5.3.2. *Under Assumption 5.3.1, there exists a vector $\boldsymbol{\psi}^* \in \mathbb{R}^n$, such that*

$$(T^*)^{-1}(\mathbf{x}[i]) = \mathcal{L}_i(\boldsymbol{\psi}^*, \mathbf{x}). \quad (5.12)$$

In words, $(T^*)^{-1}(\mathbf{x}[i])$ is the set where the mass located at point $\mathbf{x}[i]$ is sent by the optimal transportation plan. Theorem 5.3.2 states that this set is a convex polygon, namely the Laguerre cell of $\mathbf{x}[i]$ in the tessellation with a weight vector $\boldsymbol{\psi}^*$. More physical insight on the interpretation of $\boldsymbol{\psi}^*$ can be found in [87]. From a numerical point of view, Theorem 5.3.2 allows the transformation of the infinite-dimensional problem (5.4) into the following finite-dimensional problem :

$$W_2(\mu, \nu) = \max_{\boldsymbol{\psi} \in \mathbb{R}^n} g(\boldsymbol{\psi}, \mathbf{x}, \mathbf{w}), \quad (5.13)$$

where

$$g(\boldsymbol{\psi}, \mathbf{x}, \mathbf{w}) = \sum_{i=1}^n \int_{\mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x})} (\|\mathbf{x}[i] - x\|^2 - \boldsymbol{\psi}[i]) \, d\mu(x) + \sum_{i=1}^n \boldsymbol{\psi}[i] \mathbf{w}[i]. \quad (5.14)$$

Solving the problem (5.13) is the subject of numerous recent papers, and we suggest an original approach in the next section.

5.3.2 Solving the dual problem

In this section, we focus on the resolution of (5.13), i.e. computing the transportation distance numerically. The following proposition summarizes some concavity and differential properties of the functional g .

Proposition 5.3.1. *Under Assumption 5.3.1, the function g is concave with respect to the variable $\boldsymbol{\psi}$ and differentiable with a Lipschitz gradient. Its gradient is given by :*

$$\frac{\partial g}{\partial \boldsymbol{\psi}_i} = \mathbf{w}[i] - \mu(\mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x})). \quad (5.15)$$

In addition, if $\rho \in C^1(\Omega)$, then g is also twice differentiable w.r.t. $\boldsymbol{\psi}$ almost everywhere and - when defined - its second order derivatives are given by :

$$\frac{\partial^2 g}{\partial \boldsymbol{\psi}_i \partial \boldsymbol{\psi}_j} = \int_{\partial \mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x}) \cap \partial \mathcal{L}_j(\boldsymbol{\psi}, \mathbf{x})} \frac{d\mu(x)}{\|\mathbf{x}[i] - \mathbf{x}[j]\|} \quad \text{if } i \neq j. \quad (5.16)$$

The formula for the diagonal term $\frac{\partial^2 g}{\partial \psi_i \partial \psi_i}$ is given by the closure relationship

$$\forall 1 \leq i \leq n, \quad \sum_{j=1}^n \frac{\partial^2 g}{\partial \psi_i \partial \psi_j} = 0. \quad (5.17)$$

Proof 5.3.1. *Most of these properties have been proved in [80] and refined in [38]. The Lipschitz continuity of the gradient seems to be novel.*

Twice differentiability. If a Laguerre cell $\mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x})$ is empty, it remains so for small variations of $\boldsymbol{\psi}$, by the definition (5.11). It remains to prove that the set of $\boldsymbol{\psi}$ for which there exist nonempty Laguerre cells with zero measure is negligible. The fact that $\mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x})$ is nonempty of zero Lebesgue measure means that it is either a segment or a point. We consider the case, when the points \mathbf{x} are in generic position, meaning that any three distinct points are not aligned. This implies that $\mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x})$ is a singleton $\{x\}$ since the boundaries of the Laguerre cell cannot be parallel. We further assume that x belongs to the interior of Ω . Under those assumptions, x necessarily satisfies at least 3 equalities of the form

$$\|x - \mathbf{x}[i]\|_2^2 - \psi[i] = \|x - \mathbf{x}[j_k]\|_2^2 - \psi[j_k], \quad (5.18)$$

for some $j_k \neq i$ (i.e. $\mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x})$ is the intersection of at least 3 half spaces). The set of $\boldsymbol{\psi}$ allowing satisfaction of a system of equations of the form (5.18) is of codimension at least 1. Indeed, this system implies that x is the intersection of 3 lines, each perpendicular to one of the segments $[\mathbf{x}[i], \mathbf{x}[j_k]]$ and translated along the direction of this segment according to $\boldsymbol{\psi}$. Now, by taking all the finitely many sets of quadruplets (i, j_1, j_2, j_3) , we conclude that the set of $\boldsymbol{\psi}$ allowing us to make $\mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x})$ a singleton is of zero Lebesgue measure. It remains to treat the case of x belonging to the boundary of Ω . This can be done similarly, by replacing one or more equalities in (5.18), by the equations describing the boundary.

The case of points in nongeneric position can also be treated similarly, since for 3 aligned points at least 2 equations of the form (5.18) allow to turn $\mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x})$ into a line of zero Lebesgue measure. A solution of such a system is also of codimension 1.

Lipschitz gradient. In order to prove the Lipschitz continuity of the gradient, we first remark that the Laguerre cells defined in (5.11) are intersections of half spaces with a boundary that evolves linearly w.r.t. $\boldsymbol{\psi}$. The rate of variation is bounded from above by $\eta = \frac{1}{\min_{i \neq j} \|\mathbf{x}[i] - \mathbf{x}[j]\|_2}$. Hence, letting

$\Delta \in \mathbb{R}^n$ denote a unit vector, a rough bound on the variation of a single cell is :

$$\|\mu(\mathcal{L}_i(\boldsymbol{\psi} + t\Delta, \mathbf{x})) - \mu(\mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x}))\| \leq t(n-1)\|\rho\|_\infty \eta \text{diam}(\Omega).$$

Summing this inequality over all cells, we get that

$$\|\nabla_\psi g(\boldsymbol{\psi} + t\Delta, \mathbf{x}) - \nabla_\psi g(\boldsymbol{\psi}, \mathbf{x})\|_2 \leq tn^{3/2}\|\rho\|_\infty \eta \text{diam}(\Omega).$$

Notice that this upper bound is very pessimistic. For instance, applying Gershgorin's circle theorem shows that - when defined - the minimum eigenvalue of the Hessian matrix given in (5.16) is bounded below by $-n\eta\|\rho\|_\infty \text{diam}(\Omega)$.

In addition, the following proposition given in [96, Thm. 6] shows that the function g is well behaved around the minimizers.

Proposition 5.3.2. *If $\min_{x \in \Omega} \rho(x) > 0$ and the points $(\mathbf{x}[i])$ are pairwise disjoint, problem (5.13) admits a unique maximizer, up to the addition of constants. The function g is twice differentiable in the vicinity of the minimizers and strongly concave on the set of vectors with zero mean.*

Many methods have been proposed in the literature to compute the optimal vector $\boldsymbol{\psi}^*$, with the latest references providing strong convergence guarantees [7, 34, 94, 85, 80]. This may give the false impression that the problem has been fully resolved : in practice the conditions guaranteeing convergence are often unmet. For instance, it is well known that the convergence of first-order methods depends strongly on the Lipschitz constant of the gradient [102, Thm 2.1.7]. Unfortunately, this Lipschitz constant may blow up depending on the geometry of the point set \mathbf{x} and the regularity of the density ρ , see Remark 5.3.1. On the other hand, the second-order methods heavily depend on the Hölder regularity of g [74, 63]. Unfortunately, it can be shown that g is Hölder with respect to ψ only under certain circumstances. In particular, the mass of the Laguerre cells $\mu(\mathcal{L}_i(\boldsymbol{\psi}, \mathbf{x}))$ should not vanish [80, Remark 4.2]. Hence, only first-order methods should be used in the early steps of an optimization algorithm, and the initial guess should be well chosen due to slow convergence. Then, second-order methods should be the method of choice.

The Levenberg-Marquardt algorithm and the trust-region methods [137] are two popular solutions that interpolate between first- and second-order methods automatically. Unfortunately, to the best of our knowledge, the

existing convergence theorems rely on a global C^2 -regularity of the functional, which is not satisfied here. In this work, we therefore advocate the use of a regularized Newton method [111], which retains the best of first- and second-order methods : a global convergence guarantee and a locally quadratic convergence rate. The algorithm reads as follows :

$$\boldsymbol{\psi}_{k+1} = \boldsymbol{\psi} - t_k(A(\boldsymbol{\psi}) + \|\nabla_{\boldsymbol{\psi}}g(\boldsymbol{\psi}_k)\|_2\text{Id})^{-1}\nabla_{\boldsymbol{\psi}}g(\boldsymbol{\psi}_k), \quad (5.19)$$

where

$$A(\boldsymbol{\psi}) = \begin{cases} \nabla_{\boldsymbol{\psi}}^2g(\boldsymbol{\psi}_k) & \text{if } \nabla_{\boldsymbol{\psi}}^2g \text{ is defined at } \boldsymbol{\psi}_k, \\ 0 & \text{otherwise.} \end{cases} \quad (5.20)$$

The algorithm is implemented on the set of vectors with zero mean to ensure the uniqueness of a solution, see Proposition 5.3.2.

Without the term $\|\nabla_{\boldsymbol{\psi}}g(\boldsymbol{\psi}_k)\|_2\text{Id}$, the equation (5.19) would simplify to a pure Newton algorithm. The addition of this term makes the method (5.19) similar to a Levenberg-Marquardt algorithm, with the important difference that the regularization parameter is set automatically to $\|\nabla_{\boldsymbol{\psi}}g(\boldsymbol{\psi}_k)\|_2$. The rationale behind this choice is that the gradient vanishes close to the minimizer, making (5.19) similar to a damped Newton method and that the gradient amplitude should be large far away from the minimizer, making (5.19) closer to a pure gradient descent.

Following [111], we get the following proposition.

Proposition 5.3.3. *Algorithm (5.19) implemented with step-sizes t_k satisfying the Wolfe conditions converges globally to the unique maximizer of (5.13). In addition, the convergence is quadratic in the vicinity of the minimizer.*

To the best of our knowledge, this is the first algorithm to come with a global convergence guarantee. Up to now, the convergence was only local [80].

To further accelerate the convergence, the method can be initialized with the multiscale approach suggested in [94]. In practice, replacing (5.19) by a standard Levenberg-Marquardt method i.e $\boldsymbol{\psi}_{k+1} = \boldsymbol{\psi} - (A(\boldsymbol{\psi}) + c_k\text{Id})^{-1}\nabla_{\boldsymbol{\psi}}g(\boldsymbol{\psi}_k)$, yields a similar rate of convergence. In this case $c_k > 0$ is interpreted as the “step” of the descent method and it is decreased or increased following the Wolfe criterion.

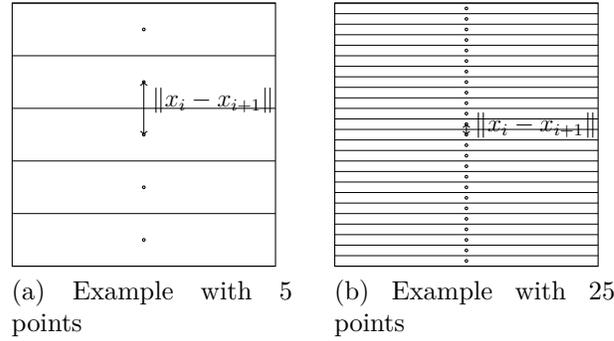


FIGURE 5.2 – Configurations of points generating a high Lipschitz constant for the gradient of g in ψ .

Remark 5.3.1 (High Lipschitz constant of the gradient). *In this example illustrated by Figure 5.2, we show that the Lipschitz constant of the gradient can be arbitrarily large. We consider that μ is the uniform measure on Ω and that ν is an atomic measure supported on n points aligned vertically and equispaced i.e. $\mathbf{x}[i] = \left(\frac{1}{2}, \frac{1+2i}{2n}\right)$ on $\Omega = [0, 1]^2$. In this case the Hessian is a multiple of the matrix of the 1d Laplacian with Neumann boundary conditions and the largest eigenvalue of H scales as $4n$. The Lipschitz constant hence blows up with the dimension. Notice that this situation is typical when it comes to approximating a density with a curve.*

5.3.3 Numerical integration

The algorithm requires computing the integrals (5.14) and (5.16). In all our numerical experiments, we use the following strategy. We first discretize the density ρ associated to the target measure μ using a bilinear or a bicubic interpolation on a regular grid. Then, we observe that the volume integrals in Equation (5.14) can be replaced by integrals of polynomials along the edges of the Laguerre diagram by using Green's formula. Hence computing the cost function, the Hessian, and the gradient all boil down to computing edge integrals.

Then, since the underlying density is piecewise polynomial, it is easy to see that only the *first moments* of the measure μ along the edges are needed to compute all formulas. We pre-evaluate the moments by using exact quadrature formulas and then use linear combinations of the moments to finish the evaluation.

To the best of our knowledge, this is a novel lightweight procedure. It significantly speeds up the calculations compared to former works [94, 34], which enables discretization of the density ρ over an arbitrary 3D mesh. After finishing this paper, we realized that the idea of using Green's formula was already suggested by [138], although not implemented. It is to be noted that this idea is particularly well suited to Cartesian grid discretization of the target density ρ . Indeed, in this case, we take advantage of the fact that the intersection of the Laguerre cells and the grid can be computed analytically without search on the mesh.

5.4 Optimizing the weights and the positions : \mathbf{w} and \mathbf{x} steps

5.4.1 Computing the optimal weights

In this section, we focus on the numerical resolution of the following subproblem

$$\operatorname{argmin}_{\mathbf{w} \in \mathbf{W}} F(\mathbf{x}, \mathbf{w}). \quad (5.21)$$

5.4.1.1 Totally constrained \mathbf{w}

When $\mathbf{W} = \{\mathbf{w}\}$ is reduced to a singleton, the solution of (5.21) is obviously given by $\mathbf{w}^* = \mathbf{w}$.

5.4.1.2 Unconstrained minimization in \mathbf{w}

When \mathbf{W} is the simplex, the unconstrained minimization problem (5.21) can be solved analytically.

Proposition 5.4.1. *If $\mathbf{W} = \Delta_{n-1}$, the solution \mathbf{w}^* of (5.21) is given for all $1 \leq i \leq n$ by*

$$\mathbf{w}^*[i] = \mu(\mathcal{L}_i(0, \mathbf{x})), \quad (5.22)$$

that is the volume (w.r.t. the measure μ) of the i -th Laguerre cell with zero cost ψ , i.e. the i -th Voronoi cell.

Proof 5.4.1. *In expression (5.14), the vector ψ can be interpreted as a Lagrange multiplier for the constraint*

$$\mu(T^{-1}(\mathbf{x}[i])) = \mathbf{w}[i].$$

Since the minimization in \mathbf{w} removes this constraint, the Lagrange multiplier might be set to zero.

5.4.2 Gradient $\nabla_{\mathbf{x}}F$ and the metric Σ_k

The following proposition provides some regularity properties of $\nabla_{\mathbf{x}}F$. It can be found in [38].

Proposition 5.4.2. *Let ψ^* denote the maximizer of (5.13). Assume that $\rho \in C^0 \cap W^{1,1}(\Omega)$, that $\mathbf{w} > 0$, and that the points in \mathbf{x} are separated. Then F is C^2 at (\mathbf{x}, \mathbf{w}) with respect to the variable \mathbf{x} and its gradient $\nabla_{\mathbf{x}}F(\mathbf{x}, \mathbf{w})$ is given by the following formula :*

$$\frac{\partial F(\mathbf{x}, \mathbf{w})}{\partial \mathbf{x}[i]} = \mathbf{w}[i] (\mathbf{x}[i] - \mathbf{b}[i]) \quad (5.23)$$

where $\mathbf{b}[i]$ is the barycenter of the i -th Laguerre cell $\mathcal{L}_i(\psi^*, \mathbf{x})$:

$$\mathbf{b}[i] = \mathbf{b}(\mathbf{x})[i] = \frac{\int_{\mathcal{L}_i(\psi^*, \mathbf{x})} x d\mu(x)}{\int_{\mathcal{L}_i(\psi^*, \mathbf{x})} d\mu(x)}. \quad (5.24)$$

Now, we discuss the choice of the metric (Σ_k) in Algorithm 3. In what follows, we refer to the “unconstrained case” as the case where there is no Π -step in Algorithm 3. The metric used in our paper is the following :

$$\Sigma_k = \text{diag}(\mu(\mathcal{L}_i(\psi_k^*, \mathbf{x}_k^*)))_{1 \leq i \leq n}. \quad (5.25)$$

We detail the rationale behind this choice below. First, with the choice (5.25), we have $\mathbf{x}_k - \Sigma_k^{-1} \nabla_{\mathbf{x}}F(\mathbf{x}_k) = \mathbf{b}(\mathbf{x}_k)$. In the unconstrained case, this particular choice of Σ_k amounts to moving the points \mathbf{x} towards their barycenters, which is the celebrated Lloyd’s algorithm.

In addition to this nice geometrical intuition, in the unconstrained case, the choice (5.25) leads to an alternating direction minimization algorithm. Indeed, given a set of points, this algorithm computes the optimal transport plan (ψ -step). Then, fixing this transport plan and the associated Laguerre tessellation, the mass of the point is moved to the barycenter of the Laguerre cell, which is the optimal position for a given tessellation. This algorithm is widespread because it does not require additional line-search.

Third, in the unconstrained case, the choice (5.25) leads to an interesting regularity property around the critical points. Assume that $\nabla_{\mathbf{x}}F(\mathbf{x}^*) = 0$,

i.e. that $\mathbf{x}^*[i] = \mathbf{b}(\mathbf{x}^*)[i]$ for all i , then the mapping $\mathbf{x} \mapsto \mathbf{b}(\mathbf{x})$ is locally 1-Lipschitz [43, Prop. 6.3]. This property suggests that a variable metric gradient descent with metric Σ_k and step-size 1 may perform well in practice for $\mathbf{X} = \Omega^n$, at least around critical points.

Fourth, this metric is the diagonal matrix with coefficients obtained by summing the coefficients of the corresponding line of $H_{\mathbf{xx}}[F]$, the Hessian of F with respect to \mathbf{x} see [38]. In this sense, Σ_k is an approximation of $H_{\mathbf{xx}}[F]$. In the unconstrained case Algorithm 3 can be interpreted as a quasi-Newton algorithm.

A safe choice of the step s_k in Algorithm 3 to ensure convergence could be driven by Wolfe conditions. In view of all the above remarks, it is tempting to use a gradient descent with the choice $s_k = 1$. In practice, it gives a satisfactory rate of convergence for Algorithm 3. For all the experiments presented in this paper, we therefore make this empirical choice. We provide some elements to justify the local convergence under a more conservative choice of parameters in Appendix 5.8.

5.5 Links with other models

5.5.1 Special cases of the framework

5.5.1.1 Lloyd's algorithm

Lloyd's algorithm [90] is well known to be a specific solver for problem (5.5), with $\mathbf{X} = \Omega$ and $\mathbf{W} = \Delta_{n-1}$, i.e. to solve the quantization problem with variable weights. We refer the reader to the excellent review by [43] for more details. It is easy to check that Lloyd's algorithm is just a special case of Algorithm 3, with the specific choice of metric

$$\Sigma_k = \text{diag}(\mu(\mathcal{L}_i(0, \mathbf{x}))). \quad (5.26)$$

5.5.1.2 Blue noise through optimal transport

The authors of [34] has proposed performing stippling by using optimal transport distance. This application can be cast as a special case of problem (5.5), with $\mathbf{X} = \Omega$ and $\mathbf{W} = \left\{ \frac{\mathbb{1}}{n} \right\}$. The algorithm proposed therein is also a special case of Algorithm 3 with

$$\Sigma_k = \text{diag}(\mu(\mathcal{L}_i(\phi^*(\mathbf{x}), \mathbf{x}))) = \frac{1}{n} \quad (5.27)$$

and the step-size τ_k is optimized through a line search. Note, however, that the extra cost of applying a line-search might not be worth the effort, since a single function evaluation requires solving the dual problem (5.13).

5.5.2 Comparison with electrostatic halftoning

In [118, 127, 61, 53, 26], an alternative to the W_2 distance was proposed, implemented and studied. Namely, the distance D in (5.1) is defined by

$$D(\nu, \mu) = \frac{1}{2} \|h \star (\nu - \mu)\|_{L^2(\Omega)}^2, \quad (5.28)$$

where h is a smooth convolution kernel and \star denotes the convolution product. This distance can be interpreted intuitively as follows : the measures are first blurred by a regularizing kernel to map them in $L^2(\Omega)$ and then compared using a simple L^2 distance. This appears in the literature under different names such as maximum mean discrepancies, kernel norms, and blurred SSD. In some cases, the two approaches are actually quite similar from a theoretical point of view. Indeed, it can be shown that the two distances are strongly equivalent under certain assumptions [109].

The two approaches however differ significantly from a numerical point of view. Table 5.1 provides a quick summary of the differences between the two approaches. We detail this table below.

- The theory of optimization is significantly harder in the case of optimal transport since it is based on a subtle mix of first- and second-order methods.
- The convolution-based algorithms require the use of methods from applied harmonic analysis dedicated to particle simulations such as fast multiple methods (FMM) [64] or non-uniform fast Fourier transforms (NUFFT) [112]. On their side, the optimal transport based approaches require the use of computational geometry tools such as Voronoi or Laguerre diagrams. The former has been parallelized efficiently on CPU and GPU and turnkey toolboxes are now available, while the latter seem to be less accessible for now and some implementations are intrinsically serial.
- The examples provided here are only two-dimensional. Many applications in computer graphics require dealing with 3D problems or larger dimensional problems (e.g. clustering problems). In that case, the numerical complexity of convolution based problems seems much better

	Convolution	Optimal transport
Optimization	1st order	Mix of 1st and 2nd
Computation	FMM/NUFFT	Power diagram
Scaling to d	Linear	Exponential
Speed in 2D	Slower	Faster
Quality	Good	Good

TABLE 5.1 – A comparison between convolution and optimal transport based approximation of measures.

controlled : it is only linear in the dimension d (i.e. $O(dn \log(n))$), while the exact computation of Laguerre diagrams requires on average $O(n^{[d/2]})$ operations, the worst case time complexity for $d = 2$ is $O(n \log n)$ [5]. Hence, for a large number of particles, the approach suggested here is mostly restricted to $d = 2$.

- In terms of computational speed for 2D applications, we observed that the optimal transport based approach was usually between 1 and 2 orders of magnitude faster. This is mostly due to the fact that the descent algorithm based on optimal transport converges in significantly fewer iterations than that based on convolution distances.
- Finally, we did not observe significant differences in terms of approximation quality from a perceptual point of view.

5.5.2.1 Benchmark with other methods

In this section we provide compare 4 methods : two versions of electrostatic halftoning [26], `ibnot` (a semi-discrete optimal transport toolbox [34]) and the code presented in this paper.

Choice of a stopping criterion The comparison of different methods yields the question of a stopping criterion. Following [118], the signal-to-noise ratio (SNR) of the original image and the stippled image convolved with a Gaussian function has been chosen. The standard deviation of the Gaussian is chosen as $1/\sqrt{n}$, which is the typical distance between points. Figure 5.3 shows different values of this criterion for an increasing quality of stippling. In all the forthcoming benchmarks, the algorithms have been stopped when the underlying measured reached a quality of 31dB.

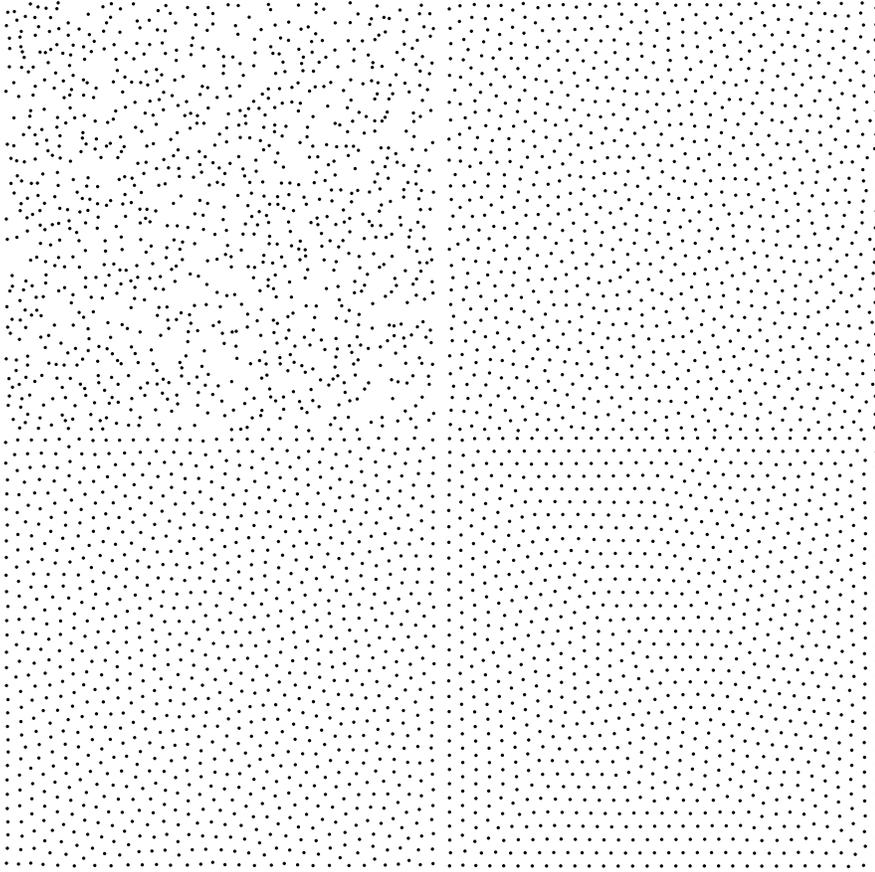


FIGURE 5.3 – Evolution of the PSNR through iteration : top left 8dB, top right 25dB, bottom left 31dB, bottom right 34dB

Benchmarks The first benchmark is illustrated in Table 5.2. For this test the background measure has a constant density and consists of 1024×1024 pixels. The number of Dirac masses increases from 2^{10} to 2^{18} . The initialization is obtained by a uniform Poisson point process.

In this case `ibnot` and our code present the same complexity and roughly the same number of \mathbf{x} -steps to achieve convergence. The time per iterations is significantly smaller in our code due to the use of Green’s formula for integration (see Section 5.3.3), which reduces the integration’s complexity from n to \sqrt{n} where n the number of pixels. We tested multiple versions of electrostatic halftoning, differing in the choice of the optimization algorithm. The code `Electro 20 cores` corresponds to a constant step-size gradient descent

# pts	Electro \equiv (20)	Electro BB \equiv (20)	ibnot \equiv (1)	our \equiv (1)
2^{10}	130.3 317	34.4 84	131.47 15	4.03 19
2^{11}	240.1 558	49.5 115	165.42 18	4.83 23
2^{12}	293.9 637	47.8 104	267.59 22	10.86 19
2^{13}	415.2 798	78.8 152	235.87 15	26.43 20
2^{14}	783.5 1306	106.3 177	344.77 17	47.90 21
2^{15}	1160.5 1319	156.8 178	598.60 18	99.63 18
2^{16}	4568.5 3286	569.2 410	1208.45 20	252.24 26
2^{17}	15875.0 4628	1676.3 489	2498.58 19	620.90 19
2^{18}	<i>TL</i>	12125.2 1103	5633.68 23	1136.51 21

TABLE 5.2 – Times in second and number of iterations to achieve convergence for the uniform background measure. TL means that the computing time was too long and that we stopped the experiment before reaching the stopping criterion. For the Electrostatic halftoning, two different methods are compared : a gradient descent with constant step-size (2nd column), and the Brazilai Borwein method (third column), numbers between brackets are number of physical core(s) used for the experiment.

as proposed in [118]. The code Electro BB 20 cores is a gradient descent with a Barzilai-Borwein step-size rule. In our experience, this turns out to be the most efficient solver (e.g. more efficient than an L-BFGS algorithm). The algorithms have been parallelized with Open-MP and evaluated on a 20-core machine using the NFFT to evaluate the sums [112].

In Table 5.2, the electrostatic halftoning algorithms are always slower than our optimal transport algorithms despite being multi-threaded on 20 cores.

The second test is displayed in Table 5.3. It consists in trying to approximate a nonconstant density $\rho(x, y)$ equal to 2 if $x < 0.5$ and 0 if $x > 0.5$. In this test we also start from a uniform point process.

In this test ibnot always fails to converge since the Hessian in ψ is not definite. In comparison with the previous test, our code is slightly slower since the first ψ -step requires to finding a ψ that maps Laguerre cells far away from the localization of their site (\mathbf{x}). After the first \mathbf{x} -step, the position of the site is decent and the optimization routine performs as well as the previous example. Again, the computing time of electro-static halftoning is significantly worse than that of optimal transport. Notice, in particular, how

# pts	Electro BB 20 cores	ibnot 1 core	our 1 core
2^{10}	40.2 99	NC	4.34 24
2^{11}	59.6 142	NC	9.70 21
2^{12}	84.5 190	NC	15.36 19
2^{13}	125.1 248	NC	29.76 25
2^{14}	177.7 282	NC	79.73 27
2^{15}	287.6 298	NC	113.56 18
2^{16}	746.3 424	NC	280.02 17
2^{17}	3052.9 712	NC	703.39 21
2^{18}	39546.1 2022	NC	1315.01 24

TABLE 5.3 – Times in seconds and number of iterations to achieve convergence for the non uniform setting. NC stands for does not converge. TL stands for too long (exceeding 4 hours of computations on 20 cores).

the number of iterations needed to reach the stopping criterion increases with the number of points, while it remains about constant for the optimal transport algorithm.

5.6 Projections on curves spaces

In this section, we detail a numerical algorithm to evaluate the projector $\Pi_{\mathbf{x}}$, for spaces of curves with kinematic or geometric constraints.

5.6.1 Discrete curves

A discrete curve is a set of points $\mathbf{x} \in \Omega^n$ with constraints on the distance between successive points. Let

$$A_1^a : \mathbf{x} \rightarrow \begin{pmatrix} \mathbf{x}[2] - \mathbf{x}[1] \\ \vdots \\ \mathbf{x}[n] - \mathbf{x}[n-1] \\ \mathbf{x}[1] - \mathbf{x}[n] \end{pmatrix} \quad (5.29)$$

and

$$A_1^b : \mathbf{x} \rightarrow \begin{pmatrix} \mathbf{x}[2] - \mathbf{x}[1] \\ \mathbf{x}[3] - \mathbf{x}[2] \\ \vdots \\ \mathbf{x}[n] - \mathbf{x}[n-1] \end{pmatrix} \quad (5.30)$$

denote the discrete first order derivatives operators with or without circular boundary conditions. From here on, we let A_1 denote any of the two operators. In order to control the distance between two neighboring points, we will consider two types of constraints : kinematic and geometrical.

5.6.1.1 Kinematic constraints

Kinematic constraints typically apply to vehicles : a car for instance has a bounded speed and acceleration. Bounded speeds can be encoded through inequalities of type

$$\|(A_1 \mathbf{x})[i]\|_2 \leq \alpha_1, \forall i. \quad (5.31)$$

Similarly, by letting A_2 denote a discrete second order derivative, which can for instance be defined by $A_2 = A_1^T A_1$, we may enforce bounded acceleration through

$$\|(A_2 \mathbf{x})[i]\|_2 \leq \alpha_2, \forall i. \quad (5.32)$$

The set \mathbf{X} is then defined by

$$\mathbf{X} = \{\mathbf{x} \in \Omega^n, \|A_1 \mathbf{x}\|_{\infty,2} \leq \alpha_1, \|A_2 \mathbf{x}\|_{\infty,2} \leq \alpha_2\}, \quad (5.33)$$

where, for $\mathbf{y} = (\mathbf{y}[1], \dots, \mathbf{y}[n])$, $\|\mathbf{y}\|_{\infty,p} = \sup_{1 \leq i \leq n} \|\mathbf{y}[i]\|_p$.

5.6.1.2 Geometrical constraints

Geometrical constraints refer to intrinsic features of a curve such as its length or curvature. In order to control those quantities using differential operators, we need to parameterize the curve with its arc length. Let $s : [0, T] \rightarrow \mathbb{R}^2$ denote a C^2 curve with arc length parameterization, i.e. $\|\dot{s}(t)\|_2 = 1, \forall t \in [0, T]$. Its length is then equal to T . Its curvature at time $t \in [0, T]$ is equal to $\kappa(t) = \|\ddot{s}(t)\|_2$.

In the discrete setting, constant speed parameterization can be enforced by imposing

$$\|(A_1 \mathbf{x})[i]\|_2 = \alpha_1, \forall i. \quad (5.34)$$

The total length of the discrete curve is then equal to $(n - 1)\alpha_1$.

Similarly, when (5.34) is satisfied, discrete curvature constraints can be captured by inequalities of type

$$\|(A_2\mathbf{x})[i]\|_2 \leq \alpha_2, \forall i. \quad (5.35)$$

Indeed, at a index $2 \leq i \leq n - 1$, we get :

$$\begin{aligned} \|(A_2\mathbf{x})[i]\|_2^2 &= \|(\mathbf{x}[i] - \mathbf{x}[i - 1]) - (\mathbf{x}[i + 1] - \mathbf{x}[i])\|_2^2 \\ &= \|\mathbf{x}[i] - \mathbf{x}[i - 1]\|_2^2 + \|\mathbf{x}[i + 1] - \mathbf{x}[i]\|_2^2 \\ &\quad - 2\langle \mathbf{x}[i] - \mathbf{x}[i - 1], \mathbf{x}[i + 1] - \mathbf{x}[i] \rangle \\ &= 2\alpha_1^2(1 - \cos(\theta_i)), \end{aligned}$$

where $\theta_i = \angle(\mathbf{x}[i] - \mathbf{x}[i - 1], \mathbf{x}[i + 1] - \mathbf{x}[i])$ is the angle between successive segments of the curve. Hence, by imposing (5.34) and (5.35), the angle θ_i satisfies

$$|\theta_i| \leq \arccos\left(1 - \frac{\alpha_2^2}{2\alpha_1^2}\right). \quad (5.36)$$

In order to fix the length and bound the curvature, we may thus choose the set \mathbf{X} as

$$\mathbf{X} = \{\mathbf{x} \in \Omega^n, \|(A_1\mathbf{x})[i]\|_2 = \alpha_1, \|A_2\mathbf{x}\|_{\infty,2} \leq \alpha_2\}. \quad (5.37)$$

Let us note already that this set is nonconvex, while (5.33) was convex.

5.6.1.3 Additional linear constraints

In applications, it may be necessary to impose other constraints such as passing at a specific location at a given time, closing the curve with $x_1 = x_n$ or having a specified mean value. All these constraints are of the form

$$B\mathbf{x} = \mathbf{b}, \quad (5.38)$$

where $B \in \mathbb{R}^{p \times 2n}$ and $\mathbf{b} \in \mathbb{R}^p$ are a matrix and vector describing the p linear constraints.

5.6.1.4 Summary

In this paper, we will consider discrete spaces of curves \mathbf{X} defined as follows :

$$\mathbf{X} = \{\mathbf{x} \text{ such that } A_i \mathbf{x} \in \mathbf{Y}_i, 1 \leq i \leq m, B\mathbf{x} = b\}, \quad (5.39)$$

The operators A_i may be arbitrary, but in this paper, we will focus on differential operators of different orders. The set \mathbf{Y}_i describes the admissible set for the i -th constraint. For instance, to impose a bounded speed (5.31), we may choose

$$\mathbf{Y}_1 = \{\mathbf{y} \in \mathbb{R}^{n \times 2}, \|\mathbf{y}_i\|_2 \leq \alpha_1, \forall i\}. \quad (5.40)$$

Throughout the paper, the set of admissible weights \mathbf{W} will be either the constant $\{1/n\}$ or the canonical simplex Δ_{n-1} .

5.6.2 Numerical projectors

The Euclidean projector $\Pi_{\mathbf{X}} : \mathbb{R}^n \rightarrow \mathbf{X}$ is defined for all $\mathbf{z} \in \Omega^n$ by

$$\begin{aligned} \Pi_{\mathbf{X}}(\mathbf{z}) &= \underset{\mathbf{x} \in \mathbf{X}}{\text{Argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 \\ &= \underset{\substack{A_k \mathbf{x} \in \mathbf{Y}_k, 1 \leq k \leq m \\ B\mathbf{x} = \mathbf{b}}}{\text{Argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 \end{aligned} \quad (5.41)$$

When \mathbf{X} is convex, $\Pi_{\mathbf{X}}(\mathbf{z})$ is a singleton. When it is not, there exists \mathbf{z} such that $\Pi_{\mathbf{X}}(\mathbf{z})$ contains more than one element. The objective of this section is to design an algorithm to find critical points of (5.41).

The specific structure of (5.41) suggests using splitting-based methods [30], able to deal with multiple constraints and linear operators. The sparse structure of differential operator makes the alternating direction method of multipliers (ADMM, [60]), particularly suited for this problem. Let us turn (5.41) into a form suitable for the ADMM.

Let $\gamma_1, \dots, \gamma_m$ denote positive reals used as preconditioners. Define

$$A = \begin{pmatrix} \gamma_1 A_1 \\ \vdots \\ \gamma_m A_m \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_m \end{pmatrix}, \quad (5.42)$$

and

$$\mathbf{Y} = \gamma_1 \mathbf{Y}_1 \times \dots \times \gamma_m \mathbf{Y}_m. \quad (5.43)$$

Problem (5.41) then becomes

$$\begin{aligned}\Pi_{\mathbf{X}}(\mathbf{z}) &= \underset{\substack{B\mathbf{x}=\mathbf{b} \\ A\mathbf{x}=\mathbf{y} \\ \mathbf{y}\in\mathbf{Y}}}{\text{Argmin}} \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 \\ &= \underset{A\mathbf{x}=\mathbf{y}}{\text{Argmin}} f_1(\mathbf{x}) + f_2(\mathbf{y}),\end{aligned}\tag{5.44}$$

where $f_1(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 + \iota_{\mathbf{L}}(\mathbf{x})$, $f_2(\mathbf{y}) = \iota_{\mathbf{Y}}(\mathbf{y})$, $\mathbf{L} = \{\mathbf{x}, B\mathbf{x} = \mathbf{b}\}$ denotes the set of linear constraints and the indicator $\iota_{\mathbf{Y}}$ of \mathbf{Y} is defined by :

$$\iota_{\mathbf{Y}}(\mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{y} \in \mathbf{Y}, \\ +\infty & \text{otherwise.} \end{cases}\tag{5.45}$$

The ADMM for solving (5.44) is given in Algorithm 4. Specialized to our problem, it yields Algorithm 5. The linear system can be solved with a linear conjugate gradient descent.

Algorithm 4 Generic ADMM.

Inputs :

functions f_1 and f_2 , matrix A , initial guess $(\mathbf{x}_0, \boldsymbol{\lambda}_0)$, parameter $\beta > 0$.

1: **while** Stopping criterion not met **do**

$$\mathbf{y}_{k+1} = \underset{\mathbf{y}}{\text{Argmin}} f_2(\mathbf{y}) + \frac{\beta}{2}\|A\mathbf{x} - \mathbf{y}_{k+1} + \boldsymbol{\lambda}_k\|_2^2.$$

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\text{Argmin}} f_1(\mathbf{x}) + \frac{\beta}{2}\|A\mathbf{x} - \mathbf{y}_{k+1} + \boldsymbol{\lambda}_k\|_2^2.$$

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + A\mathbf{x}_{k+1} - \mathbf{y}_{k+1}.$$

2: **end while**

Convergence issues The convergence and rate of convergence of the ADMM is a complex issue that depends on the properties of functions f_1 and f_2 and on the linear transform A . In the convex setting (5.33), the sequence $(\mathbf{x}_k)_k$ converges to $\Pi_{\mathbf{X}}(\mathbf{z})$ linearly (see Corollary 2 in [59]). The behavior in a non-convex setting (5.37) is still mostly open despite recent advances in [88]. Nevertheless, we report that we observed convergence empirically towards critical points of Problem (5.41).

Algorithm 5 ADMM to solve the projection problem.

Inputs :

Vector to project \mathbf{z} , initial guess $(\mathbf{x}_0, \boldsymbol{\lambda}_0)$, matrices A and B , projector $(\Pi_{\mathbf{Y}})$, $\beta > 0$.

1: **while** Stopping criterion not met **do**

$\mathbf{y}_{k+1} = \Pi_{\mathbf{Y}}(A\mathbf{x}_k + \boldsymbol{\lambda}_k)$.

Solve

$$\begin{bmatrix} \beta A^T A + I & B^T \\ B & 0 \end{bmatrix} \begin{pmatrix} \mathbf{x}_{k+1} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \beta A^T (\mathbf{y}_{k+1} - \boldsymbol{\lambda}_k) + \mathbf{z} \\ b \end{pmatrix}.$$

$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + A\mathbf{x}_{k+1} - \mathbf{y}_{k+1}$.

2: **end while**

Choosing the coefficients β and (γ_i) Despite recent advances [104], a theory to select good values of β and (γ_i) still seems lacking. In this paper, we simply set $\gamma_i = \|A_i\|_2$, the spectral norm of A_i . In practice, it turns out that this choice leads to stable results. The parameter β is set manually to obtain a good empirical behavior. Notice that for a given application, it can be tuned once for all.

5.6.3 Numerical examples

To illustrate the proposed method, we project the silhouette of a cat onto spaces of curves with fixed length and bounded curvature in Fig. 5.4. In the middle, we see how the algorithm simplifies the curve by making it smaller and smoother. On the right, we see how the method is able to make the curve longer, by adding loops of bounded curvature, while still keeping the initial cat's shape.

5.6.4 Multiresolution implementation

When \mathbf{X} is a set of curves, the solution of (5.9) can be found more efficiently by using a multiresolution approach. Instead of optimizing all the points simultaneously, it is possible to only optimize a down-sampled curve, allowing to get cheap warm start initialization for the next resolution.

In our implementation, we use a dyadic scaling. We up-sample the curve by adding mid-points in between consecutive samples. The weights from one

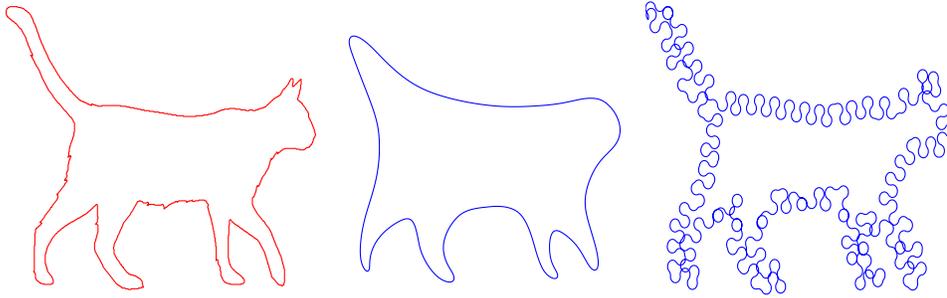


FIGURE 5.4 – Examples of projections of a curve (in red) on spaces of curves with constraints (in blue). Center : projection on sets of curves with smaller length and bounded curvature. Right : projection on sets of curves with longer length and bounded curvature.

resolution to the next are simply divided by a factor of 2.

5.7 Applications

5.7.1 Non Photorealistic Rendering with curves

In the following subsections we exhibit a few rendering results of images using different types of measures sets \mathcal{M} .

5.7.1.1 Gray-scale images

A direct application of the proposed algorithm allows to approximate an arbitrary image with measures supported on curves. An example is displayed in Fig. 5.5 with curves satisfying different kinematic constraints.

5.7.1.2 Color images

There are different ways to render color images with the proposed idea. We refer the reader, for instance, to [136, 25] for two different examples. In this section, we propose a simple alternative idea to give a color to the dots or curves. Given a target vectorial density $\rho = (\rho_R, \rho_G, \rho_B) : \Omega \rightarrow [0, 1]^3$, the algorithm we propose simply reads as follows :

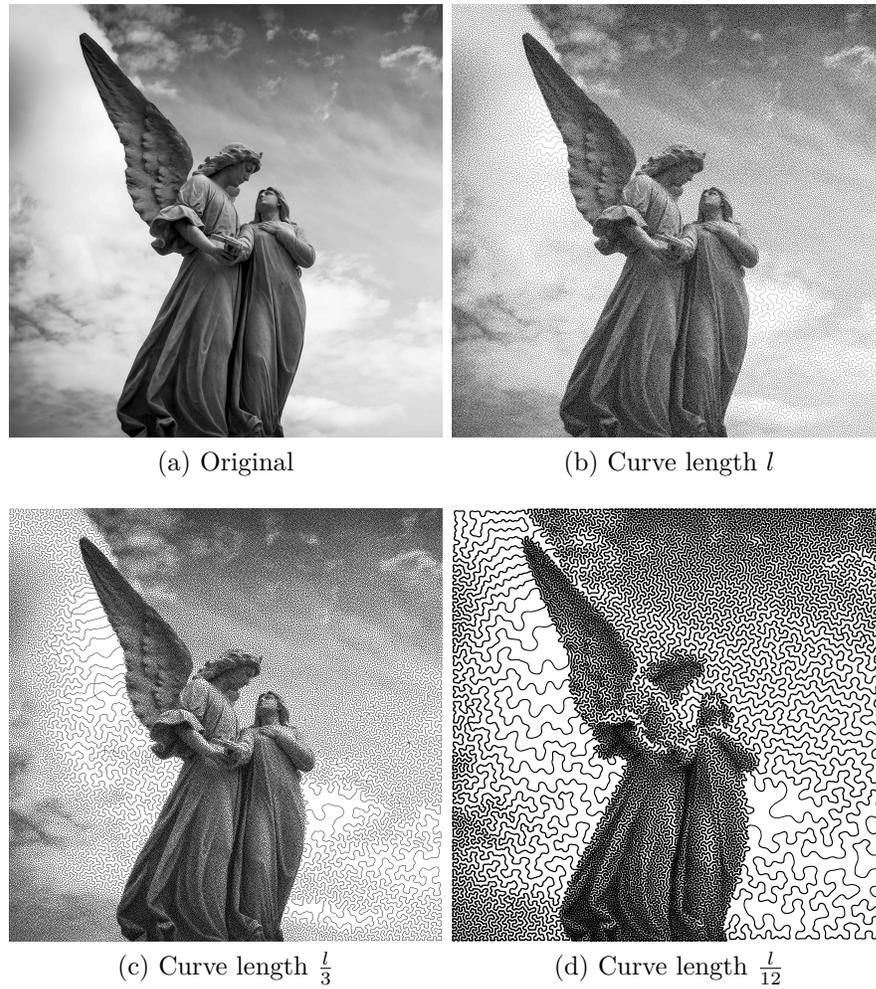
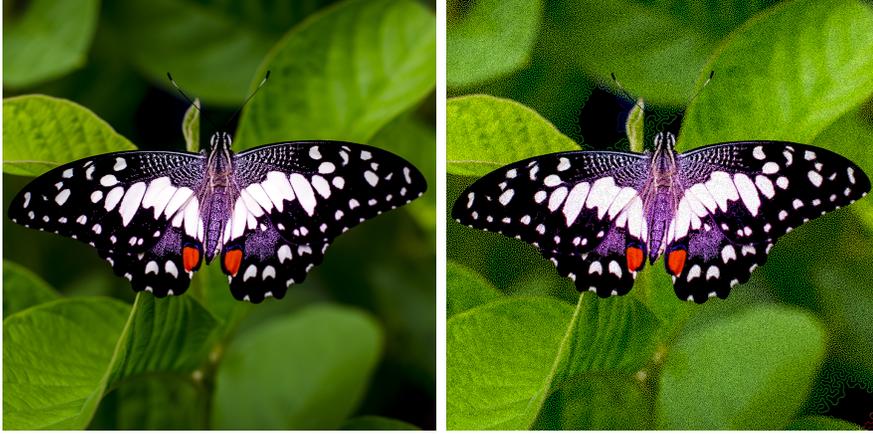


FIGURE 5.5 – Examples of curvling (stippling + curve projection, 256k, $\approx 10'$),



(a) Target color image

(b) Approximate color measure

FIGURE 5.6 – Examples of color curvling, 512k, $\approx 24'$),

- 1) We first construct a gray level image defined by :

$$\bar{\rho} = (\rho_R + \rho_G + \rho_B)/3. \quad (5.46)$$

- 2) Then, we project the density $\bar{\rho}$ onto the set of constraints \mathcal{M} with Algorithm 3. This yields a sequence of points $\mathbf{x} \in \Omega^n$.
- 3) Then, for each point $\mathbf{x}[i]$ of the discretized measure, we select a color as $\frac{\rho(\mathbf{x}[i])}{\bar{\rho}(\mathbf{x}[i])}$.

We use only saturated colors, explaining the division in step 3). The parallel for gray-scale images, is that we represent stippling results with disks taking only the maximal intensity. Then, the mean in step 1) is used to attract the curve towards the regions of high luminance of the image. An example of result of the proposed algorithm is shown in Fig. 5.6.

5.7.1.3 Dynamic examples

The codes can also be used to approximate videos. The principle is simple : first we approximate the first sequence of the frame with our projection algorithm starting from an arbitrary initial guess. Then, the other frames are obtained with the projection algorithm, taking as an initial guess, the result of the previous iteration. This ensures some continuity of the dots or curves between consecutive frames. Some videos are given in the supplementary material.

5.7.2 Path planning

In this section, we provide two applications of the proposed algorithm to path planning problems.

5.7.2.1 Videodrone

Drone surveillance is an application of increasing interest to cities, companies, and even private individuals. In this paragraph, we show that the proposed algorithms can be used to plan the drone trajectories for surveillance applications. We use the criminal data provided by [29] to create a density map of crime in Philadelphia, see Fig. 5.7a. We give different weights to different types of crimes. By minimizing (5.1), we can design an optimal path, in the sense that it satisfies the kinematic constraints of the drone and passes close to dangerous spots more often than in the remaining locations. In this example, we impose a bounded speed and a maximal yaw angular velocity and also pass at a given location at a given time to recharge the drone to satisfy autonomy constraints.

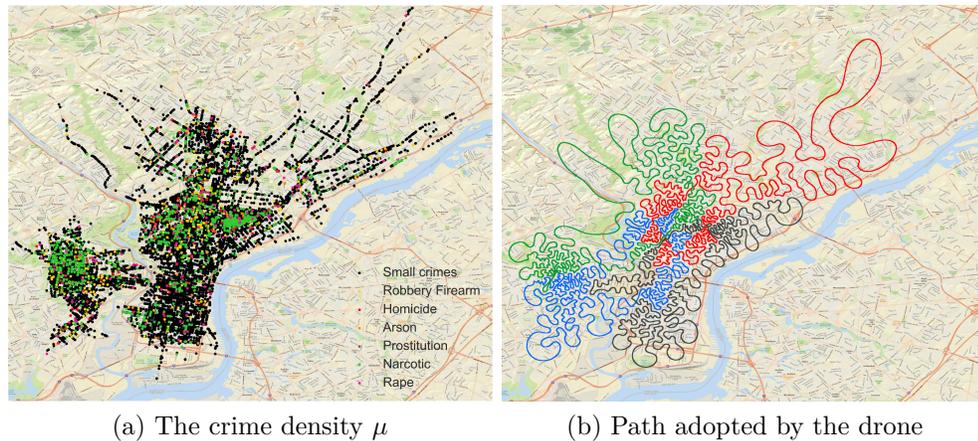


FIGURE 5.7 – The data superimposed on a map of Philadelphia. A possible drone trajectory made. In this example, the drone passes 4 times to its recharging location, explaining the different colors of the trajectory. In this example, the trajectory was discretized with 8k points and optimized in 30”.

5.7.2.2 Laser engraving

In Fig. 5.8, we gave a trajectory to a laser engraving machine in order to reproduce a landscape with a continuous line. We suspect that the same techniques could be used to optimize the nozzle and the flow of matter trajectory of 3D printers.



(a) Laser engraving machine

(b) The resulting wood engraved trajectory

FIGURE 5.8 – Example of wood engraving. Left : a laser burning the wood by following an input trajectory. Right : the final result.

5.7.2.3 Sampling in MRI

Following [21], we propose generating compressive sampling schemes in MRI (Magnetic Resonance Imaging), using the proposed algorithm.

In MRI, images are probed indirectly through their Fourier transform. Fourier transform values are sampled along curves with bounded speed and bounded acceleration, which exactly corresponds to the set of constraints defined in (5.33). The latest compressed sensing theories suggest that a good way of subsampling the Fourier domain, consists in drawing points independently at random according to a certain distribution μ , that depends on the image sparsity structure in the wavelet domain [21, 1]. Unfortunately, this

strategy is impractical in MRI due to physical constraints. To simulate such a sampling scheme, we therefore propose projecting μ onto the set of admissible trajectories.

Let $u : [0, 1]^2 \rightarrow \mathbb{R}$ denote a magnetic resonance image. The sampling process yields a set of Fourier transform values $\mathbf{y}[i] = \hat{u}(\mathbf{x}[i])$. Given this set of values, the image is then reconstructed by solving a nonlinear convex programming problem :

$$\min_{v, v|_{\mathbf{x}=\mathbf{y}}} \frac{1}{2} \|\hat{v}(\mathbf{x}) - \mathbf{y}\|_2^2 + \lambda \|\Psi u\|_1, \quad (5.47)$$

where Ψ is a linear sparsifying transform, such as a redundant wavelet transform.

5.8 Theoretical convergence of Algorithm 3

The following result is a direct application of standard convergence results, see e.g. [101].

Theorem 5.8.1. *Suppose that $X \subset \mathbb{R}^n$ is closed and convex, that $\Sigma_k = \Sigma$ is a constant positive definite matrix. In addition, suppose that F is a C^1 function with Lipschitz continuous gradient :*

$$\forall(\mathbf{x}_1, \mathbf{x}_2), \|\nabla F(\mathbf{x}_1) - \nabla F(\mathbf{x}_2)\|_{\Sigma^{-1}} \leq L \|\mathbf{x}_1 - \mathbf{x}_2\|_{\Sigma}. \quad (5.48)$$

Finally suppose that either X is compact or F is coercive. Then Algorithm 3 converges to a critical point of F for step-size $s_k = \frac{1}{L}$.

Applying Theorem 5.8.1 requires Σ_k to be constant, hence the mass \mathbf{w} to be prescribed. We make this assumption in this section.

Theorem 5.8.1 shows that it is critical to evaluate - if it exists - the Lipschitz constant of $\nabla_{\mathbf{x}} F$. By equation (5.23), we need to evaluate the variations of the Laguerre cells barycenter \mathbf{b} with respect to \mathbf{x} . Unfortunately, following the Hessian computation in [38], the Lipschitz constant scales as $\min_{i \neq j} \|\mathbf{x}[i] - \mathbf{x}[j]\|^{-1}$ and cannot be proven to be uniform in \mathbf{x} . Hence, we can only hope for a local result describing the Lipschitz constant.

Hence, if the hypotheses of existence of the Hessian of F are met (see [38]), an estimation of the Lipschitz constant of F by its Hessian yields a theory of local convergence of F in a vicinity $\mathcal{V}(\mathbf{x}^*)$ of a local minimizer

\mathbf{x}^* , for small enough steps s_k . Without these assumptions, local Lipschitz continuity of the gradient of F cannot be enforced.

If in addition there is no Π -step, that is $X = \Omega^n$, the gradient of F is 1-Lipschitz around critical points (the so-called centroidal tessellation), see [43, Prop. 6.3]. Hence, convergence can be proven in a vicinity of \mathbf{x}^* for step choice $s_k = 1$ and the metric Σ_k . However, the size of the vicinity $\mathcal{V}(\mathbf{x}^*)$ relies on the geometrical properties of the “optimal” Laguerre tessellation. The quality of such a local minimum could be very far from the global minimizer, nevertheless numerical experiments tend to indicate that is it not the case. For the same problem, hundreds of random initializations converge to a set of stationary points with a nice visual property.

5.9 Outlook

All the methods described in this chapter can be carried to $\Omega \subset \mathbb{R}^3$. The complexity for building a Laguerre tessellation in 3D is increased but affordable (as shown in Chapter 6). The use of Green’s formula that converts volume integration into surface integration is possible although its implementation is tedious. Methods to compute the 3D optimal transport distance exist and we refer the interested reader to **Geogram**,¹ a toolbox developed by Bruno Lévy.

1. <http://alice.loria.fr/index.php/software/4-library/75-geogram.html>

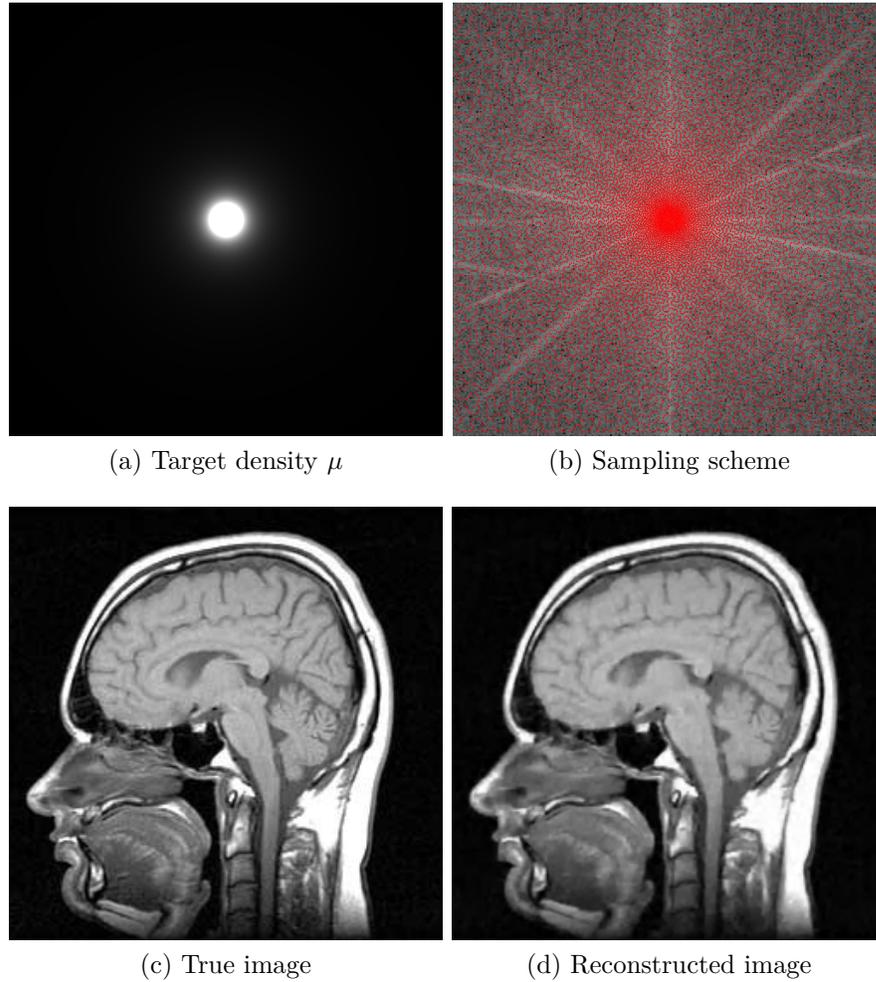


FIGURE 5.9 – Example of sampling scheme generation and image reconstruction in MRI. The target density μ is shown in 5.9a. The sampling scheme generated by our algorithm is shown in 5.9b. The background shows the Fourier transform of u in log-scale. It contains one fourth of the total number of Fourier transform values. The true image and the reconstructed image are shown in Fig. 5.9c and 5.9d.

Chapter 6

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

Contents

6.1	Introduction	115
6.2	Setting	118
6.2.1	Reminders on 2-Wasserstein distance	118
6.2.2	Setting	119
6.3	Derivatives of the cost function	121
6.3.1	First order derivative with respect to ϕ	121
6.3.2	Computation of the second order derivative with respect to ϕ	121
6.3.3	Computation of the first order derivative with respect to P and ρ	123
6.4	Numerical implementation	127
6.4.1	Integration computation	127
6.4.2	Parallelism	129
6.5	Computation of the optimal transport	130
6.5.1	Choice of optimization method	132
6.6	Numerical examples	137
6.6.1	Representation of picture	137
6.7	Conclusion	139



source code

<https://github.com/lebrat/3forthOptimalTransport>

Abstract

This paper deals with the $\frac{3}{4}$ -discrete 2-Wasserstein optimal transport between two measures, where one is supported on a set of line segments and the other one is supported on a set of points. We select the most suitable optimization procedure that computes the optimal transport. Then we address the problem of projecting point clouds on the set of measures supported on segments for the optimal transportation distance. We provide numerical examples of approximation of point clouds by segments.

This chapter is a preprint

F. de Gournay, J. Kahn, L. Lebrat

$3/4$ -discrete optimal transport.

arXiv preprint arXiv :1806.09537, 2018.

6.1 Introduction

The numerical computation of optimal transport between two measures in the sense of the 2-Wasserstein distance has seen several breakthroughs in the last decade. One can distinguish three kinds of methods : The first method is based on the PDE theory [9] and is only available when the measures are absolutely continuous with respect to the Lebesgue measure. The second method handles discrete measures and is known as the Sinkhorn algorithm [32, 10]. The main idea of this latter method is to add an entropic regularization in the Kantorovitch formulation.

The third method is known as semi-discrete [94, 85, 34] optimal transport and is limited to the case where one measure is atomic and the other is absolutely continuous. This method uses tools of computational geometry, the Laguerre tessellation which is an extension of the Voronoi diagram. The aim of this paper is to develop an efficient method to approximate a discrete measure (a point cloud) by a measure carried by a curve.

In order to solve this problem, we discretize the curve by a sequence of line segments and we solve the optimal transport between a discrete measure and a measure carried by a sequence of line segments. Since this setting lies between the fully-discrete setting and the semi-discrete setting, we coined this problem as the 3/4-discrete optimal transportation problem.

Pre-existing methods and choice of a 3/4 optimal transport. We discuss in this section the reasons that lead us to preferring 3/4 discrete optimal transport to the other methods when computing the optimal transport between a point cloud and a curve.

The first method, the PDEs formulation, requires the measures to be absolutely continuous. Even if it is always possible to mollify the considered measures, we discarded the idea of accurately meshing the space to recover a regularization of a curve.

If the Sinkhorn algorithm is to be applied to this problem, then the curve has to be sampled by points. But to the best of our knowledge [39], taking Dirac masses along anisotropic "objects", here, a curve may dwindle the efficiency of this algorithm. Indeed, in this particular case the parameter of regularization of the Sinkhorn algorithm has to be chosen smaller than the curve-sampling precision, which causes numerical issues. Since regularization has a detrimental effect on this particular type of problem, one could consider exact methods such as the Hungarian algorithm [82], the network simplex

formulation [106] or the auction algorithm [12, 15]. However time complexity of these methods is polynomial and prohibitive for a large number of points.

The semi-discrete optimal transport is the more favorable setting for such problem. In a previous paper [40] we described a method to approximate an image by a curve. We also detailed numerous applications of this method that range from non-photo realistic rendering to compress sampling schemes. This method relies on sampling of the curve by points and computes the Wasserstein distance between a density (the image) and a point cloud (the discretization of the curve). The semi-discrete approach suffers from several drawbacks, the first one is that solving the optimal transport problem is more and more difficult as discretization step of the curve decreases. Indeed the closer the Dirac masses are taken along the curve the more stretched the Laguerre tessellation is. The second flaw is the poor quality of approximation of a curve by a sum of Dirac masses. We develop this discussion in [37], to cut a long story short, an object of Hausdorff dimension one approximates more effectively a curve of Hausdorff dimension one than 0 dimensional object (Dirac masses). Finally having a plain piecewise linear curve is preferable since it eliminates a casting operation (from points to curve). Note however that if the image is understood as a probability measure with density, the approximation of an image by a curve in the 3/4-discrete setting is made at the expense of projecting this image on a sum of Dirac masses. One goal of this paper is to weight the pros and cons of the 3/4-discrete setting.

Lately, progresses have been made to extend the semi discrete optimal transport algorithm to measures supported by simplices of dimension at least 2. In this context [96] showed that the cost function is \mathcal{C}^2 under standard genericity conditions of the point cloud and connectedness of the simplices. Unfortunately, these results are not applicable to our framework since the measure has a one-dimensional support and the cost function is \mathcal{C}^1 .

Using ideas from the semi-discrete optimal transport, a solution to the above objections is to compute the optimal transport between an absolutely continuous measure and a measure carried by segments. The main tool of such computations is an extension of the Laguerre tessellation with conic boundaries. The difficulty arises when it comes to integrates the continuous density over theses Laguerre cells. In fact, a robust algorithm with exact integration is complex to develop. This is the reason why computer graphic community [72, 70] implements numerical approximations for those Voronoi cells (Laguerre cells with equals weights) using shape primitives and rasterisation with graphic hardware. However numeric precision is intrinsically

tied to grid (pixel). Hence, refinement for this method scales badly when the dimension increases.

In order to overpass these limitations, we prefer to approximate the measure supported by curve by a measure supported by segments and to compute the optimal transport between an atomic measure and a measure supported on a set of segments.

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

The ideas of this paper owes a much to [34, 94, 85] and the semi-discrete approach but it considers a measure supported by a set of segments instead of a measure with regular density. This modification annihilates the convergence theory developed in [80]. The optimal transport plan is no longer unique and Kantorovitch functional is no longer \mathcal{C}^2 . But under slight geometric condition (see (H)) one can show that the dual functional is \mathcal{C}^1 with respect to the dual variables.

Outline of the paper. Section 6.2 is devoted to set up the notations and the known results used in this paper. In (6.5) of Section 6.2, we introduce the computation of the Wasserstein distance as the maximization of a dual function g . In Section 6.3 we compute the different derivatives of the function g with respect to its parameters. In Section 6.4 we discuss technical details about the numerical implementation of the computation of the function g . In Section 6.5 we settle for the optimization procedure that solves the optimal transport, and in Section 6.6 we showcase numerical approximations of cloud data by segments.

6.2 Setting

6.2.1 Reminders on 2-Wasserstein distance

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

Given $\Omega \subset \mathbb{R}^d$, $\mu \in \mathcal{P}(\Omega)$ and $\nu \in \mathcal{P}(\Omega)$ two probability measures on Ω , the 2-Wasserstein distance between μ and ν , $W_2(\nu, \mu)$ is given by :

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

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

$$\gamma \in \Pi(\nu, \mu) \Leftrightarrow \begin{cases} \int_{\Omega} \psi(x) d\nu(x) = \int_{\Omega^2} \psi(x) d\gamma(x, y) & \forall \psi \in L^1(\nu) \\ \text{and} \\ \int_{\Omega} \phi(y) d\mu(y) = \int_{\Omega^2} \phi(y) d\gamma(x, y) & \forall \phi \in L^1(\mu) \end{cases} \quad (6.1)$$

An elegant way [133] to solve (MK) is via its dual

$$\begin{aligned} & \sup_{\psi \in L^1(\nu), \phi \in L^1(\mu)} \int_{\Omega} \psi d\nu + \int_{\Omega} \phi d\mu \\ \text{s.t. } & \forall (x, y) \in \Omega^2 : \quad \psi(x) + \phi(y) \leq \|x - y\|_2^2, \end{aligned} \quad (6.2)$$

where ϕ and ψ are the Lagrange multipliers for (6.1), the marginals constraints of (MK). Introducing the c -transform of ϕ as :

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

The problem (6.2) can be rewritten as :

$$\sup_{\phi \in L^1(\mu)} \int_{\Omega} \phi^c d\nu + \int_{\Omega} \phi d\mu. \quad (6.3)$$

Consider now the case where the measure μ is atomic with n atoms : $\mu(\mathbf{x}, m) = \sum_{i=1}^n m_i \delta_{\mathbf{x}_i}$, $\mathbf{x}_i \in \mathbb{R}^d$, $m_i \in \mathbb{R}^{+*}$. In this case $L^1(\mu)$ can be identified with \mathbb{R}^n and if $\phi = (\phi_i)_{i=1..n}$, then $\phi^c(x) = \min_{i \in \llbracket 1, n \rrbracket} \|x - \mathbf{x}_i\|_2^2 - \phi_i$. This naturally leads to the definition of the i -th Laguerre cell [5] :

$$\mathcal{L}_i(\phi, \mu) = \{x \in \Omega \text{ such that } \|x - \mathbf{x}_i\|_2^2 - \phi_i \leq \|x - \mathbf{x}_j\|_2^2 - \phi_j \quad \forall j \in \llbracket 1, n \rrbracket\}. \quad (6.4)$$

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

$$W_2^2(\nu, \mu) = \sup_{\phi \in \mathbb{R}^n} g(\phi, \nu, \mu), \text{ with } g(\phi, \nu, \mu) = \sum_i \int_{\mathcal{L}_i(\phi, \mu)} (\|x - \mathbf{x}_i\|_2^2 - \phi_i) d\nu + m_i \phi_i. \quad (6.5)$$

Assuming that a $\phi^* \in \mathbb{R}^n$, a maximum of (6.5) exists, then the physical interpretation of the Laguerre cell $\mathcal{L}_i(\phi^*, \mu)$ is that the Dirac mass located at \mathbf{x}_i is transported to $\text{supp}(\nu) \cap \mathcal{L}_i(\phi^*, \mu)$.

6.2.2 Setting

In this paper μ denotes a n -atomic probability measure on \mathbb{R}^d :

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

By contrast, ν is a probability measure supported by a polyline. A polyline is a sequence of p consecutive line segments $[P_\alpha, P_{\alpha+1}]$, where $P_\alpha \in \mathbb{R}^d$ and α runs from 1 to $p+1$. The measure ν carries a non negative mass ρ_α on the segment $[P_\alpha, P_{\alpha+1}]$. In order to define mathematically ν , we introduce l_α , the α -th segment line by :

$$\begin{aligned} l^\alpha : [0, 1] &\rightarrow \mathbb{R}^d \\ t &\mapsto (1-t)P_\alpha + tP_{\alpha+1}. \end{aligned} \quad (6.6)$$

If $\lambda_{[0,1]}$ is the Lebesgue measure on $[0, 1]$, we denote $l_{\#}^\alpha \lambda_{[0,1]}$ the push-forward through l^α of $\lambda_{[0,1]}$. In this case, ν can be written as :

$$\nu = \sum_{\alpha=1}^p \rho_\alpha l_{\#}^\alpha \lambda_{[0,1]}. \quad (6.7)$$

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

In order to ensure the regularity of function g defined in (6.5) we enforce hypothesis (H) throughout the paper.

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

Moreover, under (H), for all ϕ we have $\nu(\mathcal{L}_i(\phi, \mu) \cap \mathcal{L}_j(\phi, \mu)) = 0$. Indeed if \mathcal{L}_i and \mathcal{L}_j are two Laguerre cells with a common boundary $\mathcal{L}_i \cap \mathcal{L}_j$, the boundary has to be orthogonal to $\mathbf{x}_i - \mathbf{x}_j$. Hypothesis (H) prevents tiny perturbation of ϕ to harshly shift the association of the segment's mass from one Dirac mass to another, see Figure 6.1. Under this assumption g is a \mathcal{C}^1 function of ϕ , see [38] and Section 6.3.1. Note however than even if it is easy to check Hypothesis (H) for fixed families of $(\mathbf{x}_i)_i$ and $(P_\alpha)_\alpha$, when trying to approximate data cloud by polylines and hence optimizing the optimal transport distance w.r.to $(P_\alpha)_\alpha$, there is no way to enforce Hypothesis (H) throughout the optimisation procedure.

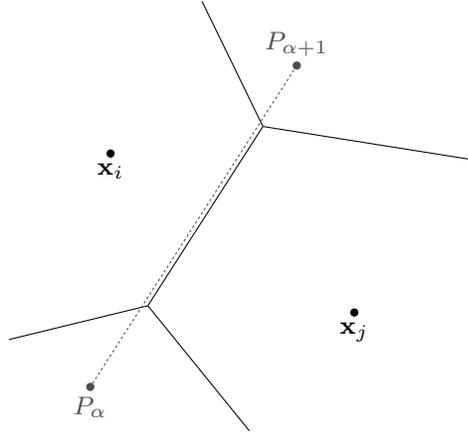


FIGURE 6.1 – When the hypothesis (H) is violated one can cook up a ϕ such that the functional g in (6.5) is no longer \mathcal{C}^1 with respect to ϕ . Here a small perturbation on ϕ_i or ϕ_j induces a discontinuity on $\nabla_\phi g$.

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

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

If the space Ω is convex then the Laguerre cells defined in (6.4) are convex and $\mathcal{L}_i \cap [P_\alpha, P_{\alpha+1}]$ is a segment. For every $(i, \alpha) \in \mathcal{I}$ denote $t_s^{i\alpha}, t_e^{i\alpha}$ the starting (respectively ending) time of the α^{th} segment $[P_\alpha, P_{\alpha+1}]$ in the i^{th} Laguerre cell :

$$0 \leq t_s^{i\alpha} \leq t_e^{i\alpha} \leq 1 \text{ and } \mathcal{L}_i \cap [P_\alpha, P_{\alpha+1}] = [l^\alpha(t_s^{i\alpha}), l^\alpha(t_e^{i\alpha})]. \quad (6.8)$$

Then the cost function $g(\phi, \nu, \mu)$ defined in (6.5) can be re-written as :

$$g(\phi, \nu, \mu) = \sum_{(i,\alpha) \in \mathcal{I}} \int_{t_s^{i\alpha}}^{t_e^{i\alpha}} r^{i\alpha}(t) dt + \sum_{i=1}^n \phi_i m_i \quad (6.9)$$

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

6.3 Derivatives of the cost function

6.3.1 First order derivative with respect to ϕ

Hypothesis (H) ensures that $\nu(\mathcal{L}_i(\phi, \mu) \cap \mathcal{L}_j(\phi, \mu)) = 0, \forall i \neq j$. A direct application of [38] shows that g is differentiable with respect to ϕ and :

$$\frac{\partial g}{\partial \phi_i}(\phi, \nu, \mu) = m_i - \int_{\mathcal{L}_i(\phi, \mu)} d\nu. \quad (6.10)$$

6.3.2 Computation of the second order derivative with respect to ϕ

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

Denote by $(e_i)_i$ the canonical basis on \mathbb{R}^n , following the calculation of the first derivative given in (6.10), the second order derivative is given by :

$$\frac{\partial^2 g}{\partial \phi_i \partial \phi_j} = - \lim_{\varepsilon \rightarrow 0} \frac{\int_{\mathcal{L}_i(\phi + \varepsilon e_j, \mu)} d\nu - \int_{\mathcal{L}_i(\phi, \mu)} d\nu}{\varepsilon}. \quad (6.11)$$

Denote n_{il} the outer normal of \mathcal{L}_i on the facet $\mathcal{L}_i \cap \mathcal{L}_l$, that is $n_{il} = \frac{x_l - x_i}{\|x_l - x_i\|}$. Denote by δ_{il} the first order approximation of the evolution of the facet $\mathcal{L}_i \cap \mathcal{L}_l$ in the normal direction n_{il} when we change the Lagrange multiplier of the j -th cell, that is, when we change ϕ into $\phi + \varepsilon e_j$, see Figure 6.2 (left). If both i and l are different from j then $\delta_{il} = 0$. For $j = l \neq i$ δ_{ij} is given by :

$$y + \varepsilon \delta_{ij} n_{ij} \in (\mathcal{L}_i \cap \mathcal{L}_j)(\phi + \varepsilon e_j, \mu)$$

$$\begin{aligned} \Leftrightarrow \|y + \varepsilon \delta_{ij} n_{ij} - \mathbf{x}_j\|_2^2 - \phi_j - \varepsilon &= \|y + \varepsilon \delta_{ij} n_{ij} - \mathbf{x}_i\|_2^2 - \phi_i & (6.12) \\ \Rightarrow \delta_{ij} &= \frac{-1}{2\|\mathbf{x}_i - \mathbf{x}_j\|}. \end{aligned}$$

Now if $i = j \neq l$, the i -th Laguerre cell increases its size, again with the same rate $\delta_{ij} = (2\|\mathbf{x}_i - \mathbf{x}_l\|)^{-1}$.

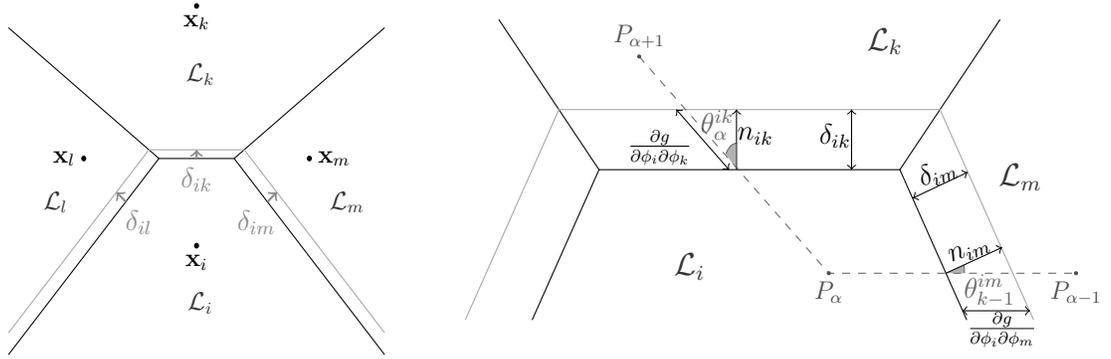


FIGURE 6.2 – (Left) Normal displacements $(\delta_{il}, \delta_{ik}, \delta_{im})$ of the boundary of the Laguerre cell when ϕ_i increases and (ϕ_l, ϕ_k, ϕ_m) remain constant. (Right) geometrical interpretation of the Hessian

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

$$\Theta_{ij} = \{\alpha \text{ such that, } [P_\alpha, P_{\alpha+1}] \cap (\mathcal{L}_i(\phi, \mu) \cap \mathcal{L}_j(\phi, \mu))\}.$$

For every α in Θ_{ij} denote θ_α^{ij} the angle between the segment $[P_\alpha, P_{\alpha+1}]$ and the outer normal n_{ij} , see Figure 6.2 (right). The second derivative is given by the variation of the first derivative (6.10), hence to the variation of the mass seen by each Laguerre cell. Now using the expression of δ_{ij} and some trigonometric formulas one can compute this variation, this leads to the following expression of the Hessian :

$$\frac{\partial g}{\partial \phi_i \partial \phi_j}(\phi, \mathbf{x}) = \begin{cases} \sum_{\alpha \in \Theta_{ij}} \frac{\rho_\alpha}{2\|\mathbf{x}_j - \mathbf{x}_i\| \cos(\theta_\alpha^{ij})} & \text{if } i \neq j, \\ -\sum_{k \neq i} \frac{\partial g}{\partial \phi_i \partial \phi_k} & \text{otherwise.} \end{cases} \quad (6.13)$$

6.3.3 Computation of the first order derivative with respect to P and ρ

In this section we compute the derivatives of the 2-Wasserstein distance with respect to the parameters of the measure ν . We denote generically as ∂_a the derivative with respect to one of the parameters that define ν , that is either one of the positions P_α or one of the densities ρ_α . Let ϕ^* be an optimal Lagrange multiplier in (6.5) defined up to an additive constant. The first step is to ensure the differentiability of ϕ^* w.r.to the considered parameter. In order to do so, additional hypotheses may be required. For instance, work in the subspace of zero-mean function in order to remove the additive constant of ϕ and suppose that g admits a continuous invertible the Hessian with respect to ϕ . An implicit function theorem will then ensure existence of $\partial_a \phi^*$. It seems difficult however to cook up a geometric hypothesis in the spirit of (H) that ensures existence of $\partial_a \phi^*$. We have to resort to assume this differentiability ϕ^* . An application of the chain rule formula yields :

$$W_2^2(\mu, \nu) = g(\phi^*, \nu, \mu) \Rightarrow \frac{\partial W_2^2}{\partial a} = \frac{\partial g}{\partial \phi} \frac{\partial \phi^*}{\partial a} + \frac{\partial g}{\partial a} = \frac{\partial g}{\partial a}, \quad (6.14)$$

since ϕ^* is a solution of (6.5), the derivative of g with respect to ϕ is zero at ϕ^* . Hence computing the differential of W_2^2 consists in differentiating the cost function g while keeping ϕ fixed at ϕ^* . Differentiating (6.9) with respect to ν we obtain :

$$\partial_a W_2^2 = \sum_{(i, \alpha) \in \mathcal{I}} (\partial_a t_e^{i\alpha}) r^{i\alpha}(t_e^{i\alpha}) - (\partial_a t_s^{i\alpha}) r^{i\alpha}(t_s^{i\alpha}) + \int_{t_s^{i\alpha}}^{t_e^{i\alpha}} \partial_a r^{i\alpha}(t) dt$$

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

1. If P_α belongs to the interior of the i th Laguerre cell then, $t_s^{i\alpha} = 0$ and $\partial_a t_s^{i\alpha} = 0$.
2. If $P_{\alpha+1}$ belongs to the interior of the i th Laguerre cell then, $t_e^{i\alpha} = 1$ and $\partial_a t_e^{i\alpha} = 0$.
3. For all i there exists exactly one j such that $t_s^{i\alpha} = t_e^{j\alpha}$. The segment $[P_\alpha, P_{\alpha+1}]$ intersects $\mathcal{L}_i \cap \mathcal{L}_j$ at point $l^\alpha(t_s^{i\alpha})$. For all x in $\mathcal{L}_i \cap \mathcal{L}_j$, by definition of the Laguerre cell (6.4), the following equality holds $\|x - \mathbf{x}_i\|_2^2 - \phi_i = \|x - \mathbf{x}_j\|_2^2 - \phi_j$ then $r^{i\alpha}(t_e^{i\alpha}) = r^{j\alpha}(t_s^{j\alpha})$.

It follows that

$$\partial_a W_2^2 = \sum_{(i,\alpha) \in \mathcal{I}} \int_{t_s^{i\alpha}}^{t_e^{i\alpha}} \partial_a r^{i\alpha}(t) dt \quad (6.15)$$

Note that equation (6.15) can be obtained directly by a formal argument. Assuming that $\partial_\nu \phi^*$ exists and define

$$h(\phi, \nu) = \int_{\Omega} \phi d\mu + \int_{\Omega} \phi^c d\nu.$$

We then have, at a formal level :

$$W_2^2(\mu, \nu) = \sup_{\phi \in L^1(\mu)} h(\phi, a) \Rightarrow \frac{\partial W_2^2}{\partial a} = \frac{\partial h}{\partial \phi} \frac{\partial \phi^*}{\partial a} + \frac{\partial h}{\partial a} = \frac{\partial h}{\partial a} = \int_{\Omega} (\phi^*)^c (\partial_a d_\nu), \quad (6.16)$$

where the term $\partial_a d_\nu$ is the variation of the measure d_ν with respect to the parameter a . Despite being even more formal, this latter argument allows to prove differentiability w.r.to the mass ρ^α . Indeed in this case, h is linear with respect to ρ^α , and then W_2^2 is convex with respect to ρ^α . Up to translating the formula (6.16) in the language of subgradients, is indeed true. Note finally that this does not help at all to differentiate W_2 with respect to P_α .

6.3.3.1 Derivative with respect to P

We are first interested in differentiating the Wasserstein distance with respect to the position of the endpoints $(P_\alpha)_{\alpha \in \llbracket 1, p+1 \rrbracket}$ of the polyline. A direct application of (6.15) yields :

$$\frac{\partial W_2^2}{\partial P_\alpha} = \sum_{i \in \mathcal{K}(\alpha-1)} \int_{t_s^{i(\alpha-1)}}^{t_e^{i(\alpha-1)}} \partial_{P_\alpha} r^{i(\alpha-1)}(t) dt + \sum_{i \in \mathcal{K}(\alpha)} \int_{t_s^{i\alpha}}^{t_e^{i\alpha}} \partial_{P_\alpha} r^{i\alpha}(t) dt$$

The differential of $r^{i\alpha}(t)$ with respect to P_α amounts to differentiate $\|l^\alpha(t) - \mathbf{x}_i\|^2$, we have that :

$$\left\{ \begin{array}{l} \frac{\partial \|l^\alpha(t) - \mathbf{x}_i\|^2}{\partial P_\alpha} = 2(1-t) (l^\alpha(t) - \mathbf{x}_i) \\ \frac{\partial \|l^{(\alpha-1)}(t) - \mathbf{x}_i\|^2}{\partial P_\alpha} = 2t (l^{(\alpha-1)}(t) - \mathbf{x}_i) . \end{array} \right.$$

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

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

We add the two previous equations together and notice that $P_{\alpha-1}, P_\alpha, P_{\alpha+1}$ are integrated between 0 and 1 :

$$\begin{aligned} \frac{\partial W_2^2}{\partial P_\alpha} &= \left(P_\alpha + \frac{P_{\alpha-1} - P_\alpha}{3} - 2 \sum_{i \in \mathcal{K}(\alpha-1)} \int_{t_s^{i(\alpha-1)}}^{t_e^{i(\alpha-1)}} t \mathbf{x}_i dt \right) \rho_{\alpha-1} \\ &\quad + \left(P_\alpha + \frac{P_{\alpha+1} - P_\alpha}{3} - 2 \sum_{i \in \mathcal{K}(\alpha)} \int_{\dagger t_s^{i\alpha}}^{\dagger t_e^{i\alpha}} t \mathbf{x}_i dt \right) \rho_\alpha \end{aligned} \quad (6.17)$$

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

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

We may interpret (6.17) as the sum of the torques of the surrounding segments $[P_{\alpha-1}, P_\alpha]$ and $[P_\alpha, P_{\alpha+1}]$ around the point P_α .

6.3.3.2 Derivative with respect to ρ

The functional g defined in Equation (6.9) is linear in ρ_α , and its derivative is therefore trivial. Throughout this paper, we consider that the density is constant by segment. As a consequence

$$\rho_\alpha = \frac{\|P_\alpha - P_{\alpha+1}\|}{\sum_\beta \|P_\beta - P_{\beta+1}\|}. \quad (6.18)$$

Set $S = \sum_\beta \|P_\beta - P_{\beta+1}\|$, the derivative of ρ with respect to P is given by :

$$\frac{\partial \rho_\alpha}{\partial P_\alpha} = \frac{1}{S} \frac{(P_\alpha - P_{\alpha+1})}{\|P_\alpha - P_{\alpha+1}\|} + \frac{1}{S^2} \left(P_{\alpha+1} - P_\alpha + (P_{\alpha-1} - P_\alpha) \frac{\|P_\alpha - P_{\alpha+1}\|}{\|P_{\alpha-1} - P_\alpha\|} \right)$$

6.3.3.3 Optimization algorithm

The aim of this section is to describe the algorithm which optimizes the Wasserstein distance W_2^2 with respect to the position of the endpoints P_α of

the polyline. The proposed algorithm is a gradient descent method with a metric Σ , we recall that the gradient of W_2^2 with respect to P is given by :

$$\nabla_P W_2^2 = \frac{\partial W_2^2}{\partial P} + \frac{\partial W_2^2}{\partial \rho} \frac{\partial \rho}{\partial P}, \quad (6.19)$$

The differential of W_2^2 with respect to P and ρ are discussed in Section 6.3.3.1 and Section 6.3.3.2 respectively. The main goal of this section is to discuss the choice of the metric Σ .

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

$$\frac{1}{2} \left(\frac{\partial W_2^2}{\partial P_\alpha} + \frac{\partial W_2^2}{\partial P_{\alpha+1}} \right) = \left(\frac{1}{2} P_\alpha + \frac{1}{2} P_{\alpha+1} - \sum_{i \in \mathcal{K}_\alpha} \int_{t_s^{i\alpha}}^{t_e^{i\alpha}} \mathbf{x}_i dt \right) \rho_\alpha = \rho_\alpha (c_\alpha - \bar{\mathbf{x}}_\alpha) \quad (6.20)$$

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

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

Equation (6.20) is reminiscent of the formula of the derivative of W_2^2 in the semi-discrete setting, see [85, 34, 94]. In this setting, μ is a measure with density and the approximating measure ν is a sum of Diracs :

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

In this case, the derivative of W_2^2 with respect to the position of the points is given by :

$$\frac{\partial W_2^2}{\partial \mathbf{y}_\alpha} = m_\alpha (\mathbf{y}_\alpha - \mathbf{b}_\alpha), \quad (6.21)$$

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

In the semi-discrete setting, the most commonly used algorithm when minimizing W_2^2 with respect to \mathbf{y} is to update the point position \mathbf{y}_α to their barycenters \mathbf{b}_α . This procedure is known as Lloyd's algorithm [34, 94, 85]. In view of the formula of the gradient (6.21), this method is a gradient descent with metric $\Sigma = \text{diag}(m_\alpha)$. Indeed, in Lloyd's algorithm, the update formula

for \mathbf{y}_α is $\mathbf{y}_\alpha \rightarrow \mathbf{y}_\alpha - \frac{1}{\mathbf{m}} \frac{\partial \mathbf{W}_2^2}{\partial \mathbf{y}_\alpha}$. In our case it is natural to consider a metric Σ defined via ρ_α as an analogy to the semi-discrete setting. We define :

$$\Sigma = \text{diag}\left(\frac{\rho_{\alpha-1} + \rho_\alpha}{2}\right). \quad (6.22)$$

The main algorithm that minimizes W_2^2 with respect to P is given by algorithm 6 below

Algorithm 6 Optimization polyline position

```

1: procedure OPTIMIZATION IN P
2:    $\nabla_P W_2^2 \leftarrow \mathbf{0}$ 
3:   do
4:      $P \leftarrow P - \Sigma^{-1} \cdot \nabla_P W_2^2$   $\triangleright$  with  $\Sigma$  defined in (6.22),  $\nabla_P W_2^2$  defined
       in (6.19)
5:      $\rho \leftarrow \text{normalizedDensity}(P)$   $\triangleright$  as in (6.18)
6:      $\phi^*, \nabla_P W_2^2 \leftarrow \text{computeOptimalTransport}(\mathbf{x}, m, P, \rho)$   $\triangleright$  see
       Algorithm 7
7:   while  $\|\nabla P\|_\infty \geq 10^{-3}$ 
8: end procedure

```

6.4 Numerical implementation

The Laguerre cells are computed using the computational geometry library CGAL [128]. In 2D the algorithm is fast and scalable, the average complexity for n sites randomly drawn is linear in time and memory. For 3D triangulation the worst case complexity is quadratic, but for random point configurations the complexity is observed to be almost linear [46]. In this section we discuss the computation of g and its parallelization.

6.4.1 Integration computation

The main issue when computing g and its derivative is to evaluate the intersections (6.8) between the polylines and the Laguerre cells. The intersections are computed by following each segment $l^\alpha(t)$ with t increasing. We first focus our attention on computing the exit time of the Laguerre cell j knowing the starting time $t_s^{i\alpha}$. To clarify things suppose that at the known

time $t_e^{i\alpha} = t_s^{j\alpha}$ the segment exits the Laguerre cell \mathcal{L}_i and enters the Laguerre cell \mathcal{L}_j . The objective is to compute the time $t_e^{j\alpha} = t_s^{k\alpha}$ and the index k such that the segment exits the Laguerre cell \mathcal{L}_j at time $t_e^{j\alpha}$ and enters the Laguerre cell \mathcal{L}_k . Such a time is computed by solving the following minimization problem

$$k = \underset{m \text{ s.t. } m \neq i, t_s^{j\alpha} < t_m \leq 1}{\operatorname{argmin}} t_m, \quad (6.23)$$

$$\text{with } t_m = \frac{2\langle P_\alpha, \mathbf{x}_j - \mathbf{x}_m \rangle + \|\mathbf{x}_m\|^2 - \phi_m + \phi_j - \|\mathbf{x}_j\|_2^2}{2\langle P_\alpha - P_{\alpha+1}, \mathbf{x}_j - \mathbf{x}_m \rangle}.$$

First notice that t_m is the time when the α -th segment hits the intersection of the cells i and j if all the other Laguerre cells are ignored. Second, the constraints $m \neq i$ and $t_s^{j\alpha} < t_m$ are deliberately redundant, it is a numerical safeguard that prevent the code from endless loops.

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

In the case when $t_s^{j\alpha} = 0$, that is the segment starts in the Laguerre cell \mathcal{L}_j , we consider the same minimization problem as (6.23) where the constraint set is replaced with $\{m, 0 \leq t_m \leq 1\}$.

We still have to compute the index of the Laguerre cell where the segment $[P_\alpha, P_{\alpha+1}]$ begins. If $\alpha > 0$, it is obviously the index of the ending Laguerre cell of $[P_{\alpha-1}, P_\alpha]$. For the case $\alpha = 0$, we add a dummy segment $[\mathbf{x}_l, P_\alpha]$, where \mathbf{x}_l is the position of the l -th Dirac mass corresponding to the largest multiplier $\phi_l = \max_k \phi_k$. By definition of the Laguerre diagram (6.4) the point \mathbf{x}_l belongs to \mathcal{L}_l .

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

There might exist several solutions k to the minimization problem (6.23). In this case, the segment encounters a corner, the intersection of at least 3 Laguerre cells. In practice this case never occurs thanks to floating point arithmetic but the rounding error can elect a non-suitable candidate. In this case, the algorithm assigns a segment of negligible length to the candidate, see Figure 6.3.

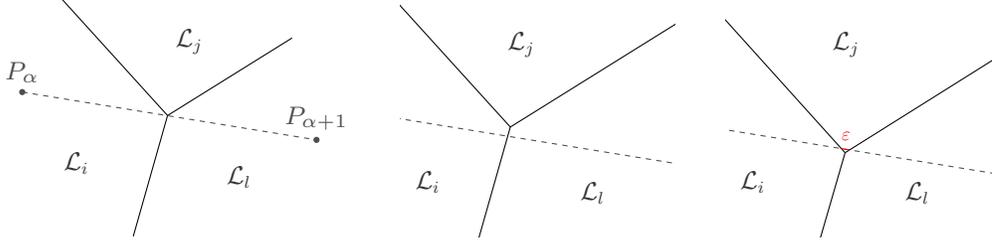


FIGURE 6.3 – Example of non-uniqueness to the minimization problem (6.23). At the left the polyline $[P_\alpha, P_{\alpha+1}]$ intersects the junction of 3 Laguerre cells (i, j, l) . In the center $k = l$ chosen as expected. At the right $k = j$ chosen and $t_e^{j\alpha} - t_s^{j\alpha} = \varepsilon$, a small error occurs assigning a bit of the segment to the j th Laguerre cell.

6.4.2 Parallelism

The evaluation of $g(\phi, \nu, \mu)$ and its derivative with respect to ϕ are discussed in Section 6.4.1. For each segment of the polyline the intersection times are computed sequentially by (6.23) and the integration is performed on the fly. The parallelism is unfurled at the highest level of the algorithm for the segments. Remember that the computation of the intersection of the polyline and the Laguerre tessellation requires for each segment the knowledge of the cell of its starting point. This cell can be inferred from the calculation of the previous segment. Hence, we slice the polyline into contiguous chunks of segments with equivalent size, each worker dealing with one chunk. For each of these chunks the sequential integration procedure is launched and then the results are merged. We carry out both OpenMP and `c++11` thread class implementation. For a standard chunk size and for a regular processor the `c++11` performance overtakes the OpenMP's one. We suppose that this difference of performance (up to a factor two) is due to OpenMP overhead.

In Figure 6.4, the evolution of the computation time of the cost function g and its gradient with respect to ϕ is benchmarked. The speedup unit is defined as the ratio of the execution time of the non-parallelized task over the parallelized task. OpenMP and `c++11` thread class are both implementations of shared memory parallelization, the memory is simultaneously accessible for every thread. The performance of a process depends intrinsically on how close the data is. If the data is on processor cache the latency is a dozens

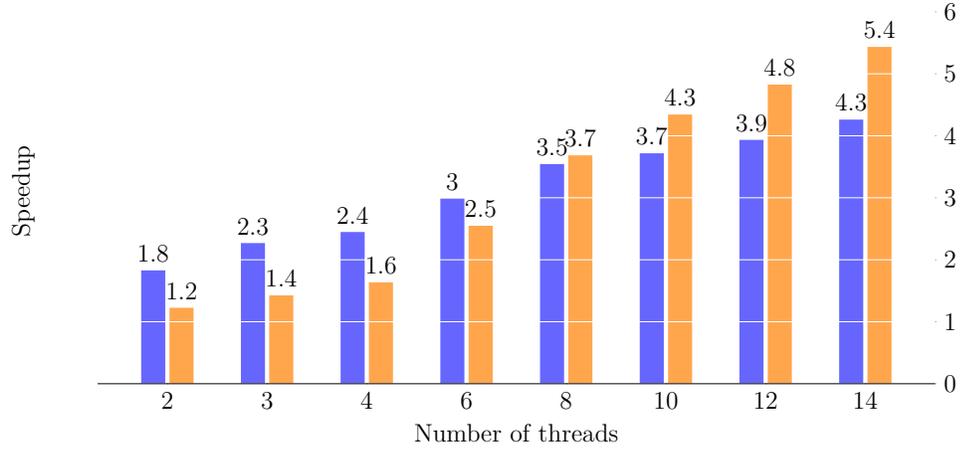


FIGURE 6.4 – Speedup for the computation of the cost function g with `c++11` thread class in blue, and `OpenMP` in orange . The parallelism is placed over the segment integration for $n = 200K$ points and $p = 80K$ segments for an increasing number of threads with a super-calculator equipped with Intel Xeon[®] E5-2680

of CPU cycles, if the data is located on RAM, the latency is $40ns$. In our experiment, for large number of threads, 30

6.5 Computation of the optimal transport

The goal of this section is to compare the available methods to optimize g with respect to ϕ (6.5). The functional g is concave with respect to ϕ since it is a dual formulation of the problem (MK) see [133]. Under the hypothesis (H) the functional g is \mathcal{C}^1 with respect to ϕ . Note that a higher level of regularity cannot be established as argued in Figure 6.5. This lack of regularity precludes the use of the convergence framework for second order methods established by Mériçot *et al.* [80].

However, we may apply classical convergence results of concave analysis to functional the g . To trigger those arguments, the Lipschitz constant of the gradient L has to be bounded. Under Hypothesis (H), using the notation of Section 6.3.2 summarized in Figure 6.2 and the expression of the Hessian matrix (6.13), we establish a finite upper bound for L using the Gershgorin circle theorem :

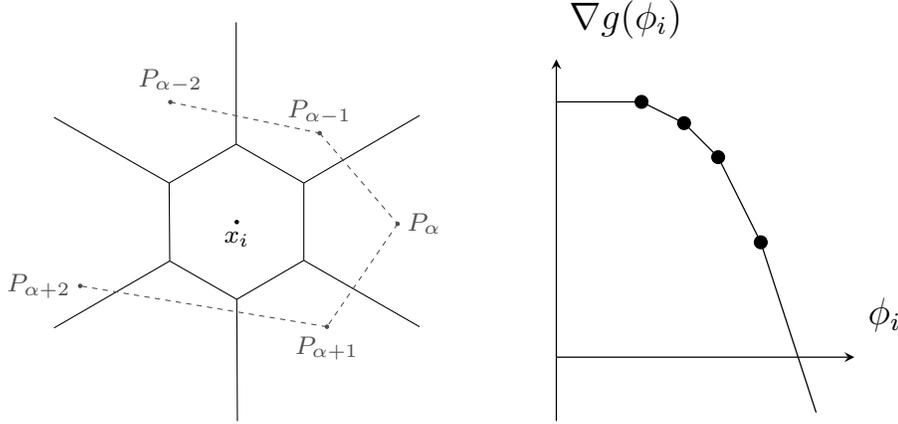


FIGURE 6.5 – Counter-example to the smoothness of ∇g . Consider the a Laguerre tessellation and a set of segment as displayed in the left and increase ϕ_i . The Laguerre cell \mathcal{L}_i increases and the gradient of g with respect to ϕ exhibits kinks. Those kinks happen each time a Laguerre cell meets a line segment for the first time. Under Hypothesis (H) boundary segments of the Laguerre cell are never parallel to a line segment. If Hypothesis (H) is not met ∇g may fail to exist.

$$|L| \leq 2 \max_i \sum_{j \neq i} \left| \frac{\partial^2 g}{\partial \phi_i \partial \phi_j} \right| \leq \frac{p(n-1) \max_{\alpha}(\rho_{\alpha})}{\min_{l \neq m} (\|\mathbf{x}_l - \mathbf{x}_m\|_2) \min_{i,j,\alpha, i \neq j} |\cos(\theta_{\alpha}^{ij})|}, \quad (6.24)$$

where p is the number of segments composing the polyline and n the number of Dirac masses composing the discrete measure, ρ_{α} is the density associated to each segment and θ_{α}^{ij} is the angle between the vector $P_{\alpha+1} - P_{\alpha}$ and $(\mathbf{x}_j - \mathbf{x}_i)$. For fixed measures ν and μ the evaluation of the bound (6.24) is costly as it involves combinatorial quantities and is not evaluated in practice.

The gradient Lipschitz condition (6.24) is sufficient to ensure convergence of ascent methods. Note that g is bounded from above as μ and ν are compactly supported measures with the same total mass. The gradient method with step $s < \frac{2}{L}$ converges to a stationary point ϕ^* see [110, 13, 102]. The same holds for variant step-size method with line search such as Global Bralazai Borwein algorithm [113, 51].

Because of the counter-example in Figure 6.5, and the lack of regularity of the gradient of g , quadratic convergence cannot be guaranteed for second

order methods. Note also that in a generic setting, the Newton method is impractical since the Hessian matrix defined in equation (6.13) fails to be invertible, see Section 6.5.1.3.

6.5.1 Choice of optimization method

The different methods are tested against the same benchmark. It consists in drawing uniformly 10K points and 500 segments in 2D. Several realizations of the optimization procedure are performed and plotted in the corresponding figures 6.6, 6.7, 6.8 and 6.9. The methods are benchmarked in Table 6.1.

6.5.1.1 First order method

We first implement a gradient ascent method. The generic convergence history is displayed in Figure 6.6. As explained in Section 6.5, the Lipschitz constant of the gradient cannot be satisfactorily computed, so the step-size is chosen according to strong Wolfe conditions. In practice the algorithm settles for a constant step of 0.05. As it can be observed in Figure 6.6 the rate of convergence of the gradient towards zero is too slow for this method to be used in practice.

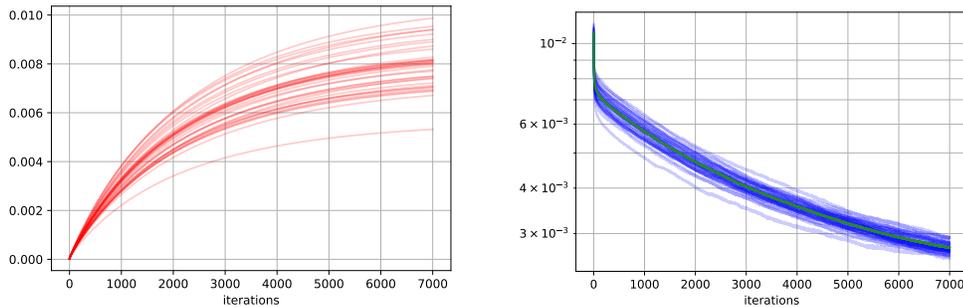


FIGURE 6.6 – Gradient ascent method : cost function history (left) and norm of the gradient (right). Method stopped by the max iteration criterion.

We have tried other first order methods including Polak-Ribière, Fletcher-Reeves, Barzalai Borwein and Nesterov acceleration. Only Nesterov acceleration has behaved differently from the other methods and it is displayed in Figure 6.7. Note however, that Nesterov acceleration requires an estimation of L , the Lipschitz constant of the gradient. Figure 6.7 was obtained with

an estimation of L that promotes quick decreases of gradient norm at the beginning of the algorithm. Note however that the estimation of L seems too optimistic since the Nesterov algorithm does not converge. A more pessimistic choice of L leads to a convergence rate similar to the gradient algorithm, at least during the first 7K iterations.

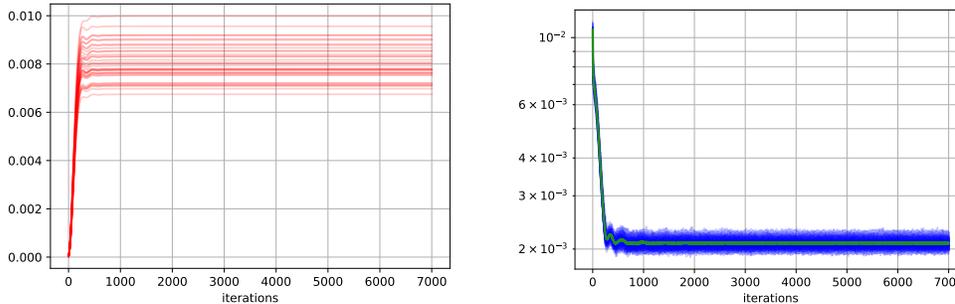


FIGURE 6.7 – Nesterov acceleration of gradient ascent : cost function history (left) and norm of the gradient (right). Method stopped by the max iteration criterion.

6.5.1.2 Quasi-Newton Method

The main idea behind quasi-Newton methods is to build at the iteration k an approximation of the Hessian matrix of g . We choose the limited memory BFGS method, which only stores a limited amount of vectors determined by the user. The result is displayed in Figure 6.8. This method converges faster than first order method, the L-BFGS algorithm reaches the desired gradient tolerance 5.10^{-5} within 1500 iterations. By contrast first order methods failed to converge in 7000 iterations. However when the method is close to a critical point the convergence speed tends to decrease.

6.5.1.3 Newton and quasi-Newton hybridization

In the light of the previous section, the flaw of the quasi-Newton method is its lack of briskness when it reaches the vicinity of a maximizer. Second-order methods are known to converge quadratically in the basin of attraction of a maximizer. The idea of the hybrid algorithm is to start with a quasi-Newton method, and then switch to a second order Newton algorithm when

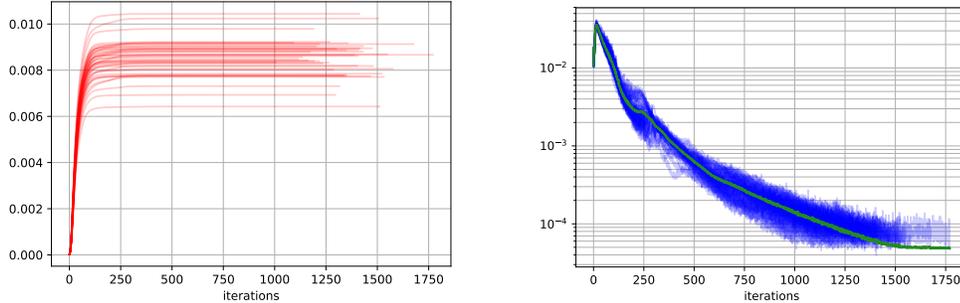


FIGURE 6.8 – L-BFGS method : cost function history (left) and norm of the gradient (right). Method converges to the targeted gradient norm 5.10^{-5} in a average of 1500 iterations.

the basin of attraction is reached. The quandary is to determine when to start the Newton method in preference to the L-BFGS method. Indeed far from the basin of attraction the direction given by the Newton method is poor and the natural step 1 is truncated by the line search which causes extra computation time.

We propose to switch from the L-BFGS algorithm to the Newton algorithm as soon as there is no empty Laguerre cell. A Laguerre cell \mathcal{L}_i is called empty as long as there is no mass assigned to its centroid \mathbf{x}_i that is $\nu(\mathcal{L}_i) = 0$. This criterion is informally motivated by the fact that the second order information relates the competition and the connectivity between Laguerre cells, hence if one of them is empty its information is inconsequential. From a practical point of view, this condition ensures that the Hessian has a kernel of dimension 1 (the dual variable ϕ is defined up to an additive constant). The condition $\nu(\mathcal{L}_i) \neq 0$ is checked throughout the algorithm via the variable **hiddenNumber** which totals the number of empty Laguerre cells. In order to invert the Hessian, it is first shifted with a small multiple of the identity in order to prevent the 0 eigenvalue and a Cholesky procedure is performed. This procedure is described in Algorithm 7 and the result are displayed in Figure 6.9.

In the numerical tests, the switch between the two Newton method happens approximatively 30 iterations before termination of the algorithm. During the Newton phase of the hybrid algorithm, the algorithm typically undergoes two stages. During the twenty first iterations of the Newton method the

Algorithm 7 Computation of optimal transport

Ensure: Dirac positions \mathbf{x} and masses m .**Ensure:** Nodes P and density ρ of the polyline.**Require:** ϕ_{init} a starting ϕ for the computation of g **Require:** gradTol L2 tolerance on the gradient norm**Require:** outerMax maximum number of iterations**Require:** wolfeProcedure standard line search with strong Wolfe condition and initial step $s = 1$. [13]

```

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

```

algorithm stabilizes around the maximum and the Wolfe line-search prevents picking $s = 1$. During the second stage, quadratic convergence is achieved in few iterations.

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

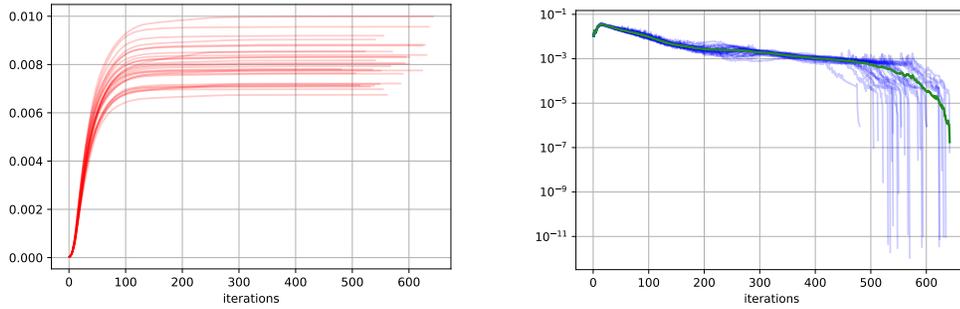


FIGURE 6.9 – Hybrid Newton method : cost function history (left) and norm of the gradient (right). The method converges to the targeted gradient norm 10^{-6} in 600 iterations on average.

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

TABLE 6.1 – Comparison of algorithms for solving the optimal transport problem for 10^4 points and 500 lines. Time required for an iteration in millisecond and the gradient norm after 1000 iterations. The time per iterations are averaged over 10^4 iterations, the standard deviation of this mean is below 0.5%. The BFGS/Newton method converges up to numerical error.

6.6 Numerical examples

6.6.1 Representation of picture

The approximation of a measure by a curve has multiple applications of which some are described in [40]. In this section we discuss the representation of a picture by a polyline. The first step is the discretization of the picture, here a landscape, by a sum of Dirac masses of weights m . Several approaches are possible but the most intuitive is to take the Dirac positions \mathbf{x} on a Cartesian grid and their weights equal to the pixels intensities. The polyline is then initialized randomly, and Algorithm 6 is launched. With no speed or curvature constraints, the gradient method in P described in Algorithm 6 empirically gives a stationary point. This solution is highly non-smooth, the length of the segments and the angle between them are disparate see Figure 6.10 top right. We implement the projection step on kinematical constraints (Π -step of Algorithm 3) described in Chapter 5 after computing the optimal transport's gradient step. This step allows to cancel scibbled result depicted in the top right image of Figure 6.10.

This scibbled solution is, in practice, difficult to carry out with laser engravers. Indeed for the same number of segments p , the engraving time of uncontrolled trajectories can be up to five times longer. In order to get around this problem, we follow the method described in [40], that is projecting the polyline after the gradient step of Algorithm 6 on a set of kinematic constraints. The constraints imposed on the speed and the acceleration polyline read as :

$$\mathcal{K}(K_1, K_2) = \left\{ (P_\alpha)_{\alpha \in \llbracket 1, p \rrbracket} \text{ s.t. } \left\{ \begin{array}{ll} \|P_{\alpha+1} - P_\alpha\| \leq K_1, & \forall \alpha \in \llbracket 1, p-1 \rrbracket \\ \|2P_\alpha - P_{\alpha-1} - P_{\alpha+1}\| \leq K_2, & \forall \alpha \in \llbracket 2, p-1 \rrbracket \end{array} \right. \right\}, \quad (6.25)$$

with K_1 the constant controlling the speed of the polyline and K_2 its acceleration. The projection on the set $\mathcal{K}(K_1, K_2)$ is performed using the Alternating Direction Method of Multipliers (ADMM) [60].

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

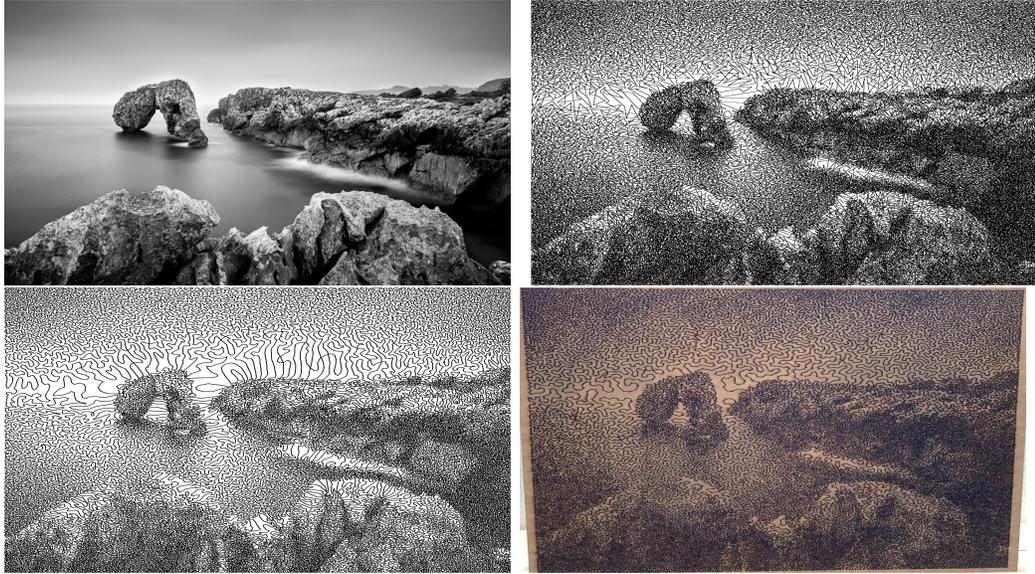


FIGURE 6.10 – Curvling by $\frac{3}{4}$ -discrete optimal transport. Original image (top left), approximation by a polyline with 400K segments without any constraints (top right), approximation by a polyline with kinematic constraints and 250K segments (bottom left), final rendering after wood engraving, the polyline is composed of 80K segments (bottom right). In all tests measure μ is represented by 320K Dirac masses.

6.6.1.1 Galaxy filaments

Galaxies are known to cluster along filaments and other low dimensional structures [125, 16]. However mathematical extraction of these filaments is a challenging task. We might try to apply refinements of our algorithm to find those filaments. As a naive proof of concept, we have applied our method while setting $\rho_{2k+1} = 0$ for all k , so that the lines are disjoint. We use the data of [126] available at : <https://github.com/etempel/bisous>. The galaxies are represented by Dirac masses. Note that in our tests their masses m are arbitrarily set to $\frac{1}{n}$ but the code support other values. In Figure 6.11, computations are performed for $n = 180\text{K}$ galaxies and a decreasing number of filaments 8K,1K,500.

Despite the fact that the representation of this 3D data is difficult to interpret, we believe that searching directly solution of the galaxy filaments by our method is not a reasonable thing to do. Indeed our problem is prone

to be stuck at local minima. We are currently working on using our method as a post-processing algorithm which is fed by accurate initialization given by state-of-the-art algorithms.

6.7 Conclusion

We study the approximation of point clouds by curves for the Wasserstein distance. We first discuss the choice of discretization and then settle for discretizing curves by line segments. We use Laguerre cells in order to compute the optimal transport problem. We derive formally the formulas for the differentiation of the Wasserstein distance with respect to the discretization parameters of the curve. We discuss technical points of the numerical implementation and we illustrate our method by numerical experiments in $2D$ and $3D$. The main drawback of this work is that we assume existence of differential of the Wasserstein distance with respect to the discretization parameters of the curve. An interesting future work is to find simple and plausible hypotheses that allow this differential to exist.

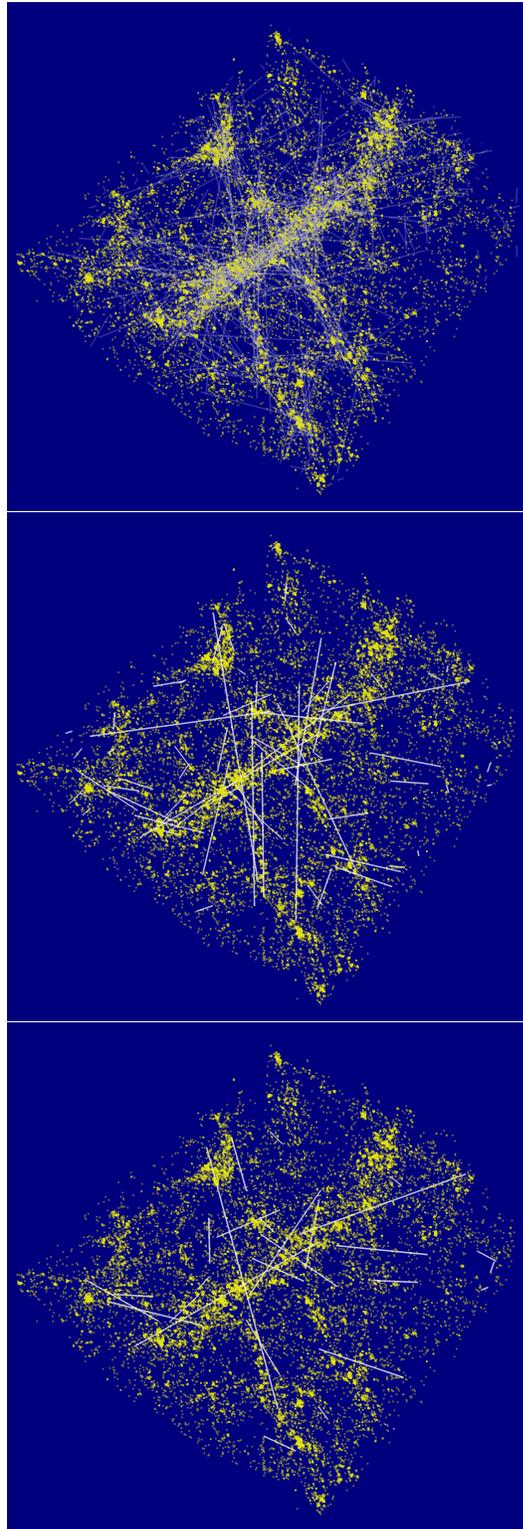


FIGURE 6.11 – A thin slice of a solution for locating galaxy filaments (white), and galaxies (yellow).

Chapter 7

Approximation of curves with piecewise constant or piecewise linear functions

Contents

7.1	Introduction	143
7.2	Notation	144
7.3	Main result	147
7.4	Proof of the theorems	151
7.4.1	Notations and technical lemmas	151
7.4.2	Approximation of function by splines	153
7.4.3	Approximation of splines by functions	156
7.4.4	Wasserstein distance	159
7.4.5	Proof of theorems	163
7.5	Numerical implementation	165
7.6	Outlook	165



source code

<https://github.com/lebrat/eulerianApproximation>

Abstract

In this paper we compute the Hausdorff distance between sets of continuous curves and sets of piecewise constant or linear discretizations. These sets are Sobolev balls given by the continuous or discrete L^p -norm of the derivatives. We detail the suitable discretization or smoothing procedure which are preservative in the sense of these norms. Finally we exhibit the link between Eulerian numbers and the uniformly space knots B-spline used for smoothing.

This chapter is a preprint

F. de Gournay, J. Kahn, L. Lebrat

Approximation of curves with piecewise constant or piecewise linear functions.

arXiv preprint arXiv :1909.04582, 2019.

7.1 Introduction

In this paper we compute the Hausdorff distance between sets of continuous curves and sets of piecewise constant or linear discretizations. These sets are Sobolev balls given by the continuous or discrete L^p -norm of the derivatives. We detail the suitable discretization or smoothing procedure which are preservative in the sense of these norms. Finally we exhibit the link between Eulerian numbers and the uniformly space knots B-spline used for smoothing.

Introduction

This article focuses on a widespread problem of approximation which consists in approaching a curve by a set of points or by a piecewise linear function (line segments or polyline). We also analyze the reverse operation called smoothing, which amounts to, given a set of points or polyline find an approaching curve with an higher level of regularity. These two approaches yields the instinctive question : how well can we approximate a particular space of curves with a particular set of points sets or polyline sets.

This subject has been thoroughly studied, especially by the computer vision community [45, 130, 100]. The most common approach is to find minimal length objects controlling some approximation error and the limit error. Our view is different, since we want to approximate in a Hausdorff sense, that is to approximate each curve by a set of points or polylines, and each set of points or polylines by a curve, so that all approximations are close for an appropriate distance.

In practice, the authors have encountered this question when trying to computationally project a measure on a space of pushforward measures of curves [21, 26, 36, 40] : the implementation needs a discretization, and it is guaranteed to work only if both directions of approximation are small for the transportation distance \mathcal{W}^1 . It is likely that this kind of results may be useful in other contexts in computer science.

In this article, we prove that Sobolev balls and similar spaces may be approximated by discretized Sobolev spaces, where the norm is given by discrete derivatives. As explained in the notations, the Hausdorff distance comes from the transportation distance on both time and space, giving a very robust meaning to the approximation.

An ingredient in the proof is exhibiting a Sobolev curve that approximates a given set of points. The construction makes use of Eulerian numbers. Given their known connection to B-splines [67, 134], this might not be so surprising.

7.2 Notation

Throughout the paper, κ_a will denote a constant depending only on a that might change from line to line.

Curves are in \mathbb{R}^d and we identify discretization of curves with families of vectors $\mathbf{p} = (\mathbf{p}_0, \dots, \mathbf{p}_{n-1})$. Even if we tackle both periodic and non-periodic cases, the notations are tailored for the periodic case, which allows the abuse of notation $\mathbf{p}_i = \mathbf{p}_{i \pmod n}$ for each $i \in \mathbb{Z}$. In this setting, the discrete convolution product for a family of vectors $\mathbf{p} \in \mathbb{R}^{n \times d}$ and $K \in \mathbb{R}^n$ reads as :

$$(K \star \mathbf{p})_i = \sum_{j=0}^{n-1} K_j \mathbf{p}_{i-j},$$

Given any norm $\|\bullet\|$ in \mathbb{R}^d , the discrete renormalized ℓ^q norm is defined as

$$\|\mathbf{p}\|_{\ell^q} = \left(\frac{1}{n} \sum_{i=0}^{n-1} \|\mathbf{p}_i\|^q \right)^{1/q}.$$

Note that this special choice of renormalization of the ℓ^q norm turns Young's convolution inequality into :

$$\|K \star \mathbf{p}\|_{\ell^q} \leq n \|K\|_{\ell^1} \|\mathbf{p}\|_{\ell^q} \quad \forall q \geq 1. \quad (7.1)$$

The convolutional discrete derivative operator, Δ is defined by

$$(\Delta \star \mathbf{p})_i = \mathbf{p}_i - \mathbf{p}_{i-1}.$$

Similarly for any $m \in \mathbb{N}$, the m -order discrete convolutional derivative operator $\Delta^{\star m} \in \mathbb{R}^n$ is defined by the recursion formula $\Delta^{\star(m+1)} = \Delta \star \Delta^{\star m}$ with $\Delta^{\star 1} = \Delta$. Its closed form is given by :

$$\Delta_i^{\star m} = (-1)^i \binom{m}{i}. \quad (7.2)$$

Let us also define $\mathbb{1}$ as the identity for the convolution, and T the shift operator :

$$\mathbb{1} = \begin{cases} \mathbb{1}_0 = 1 \\ \mathbb{1}_i = 0 \quad i \neq 0 \end{cases} \quad \text{and} \quad T = \begin{cases} T_1 = 1 \\ T_i = 0 \quad i \neq 1. \end{cases}$$

The discrete derivative operator can be written as $\Delta = \mathbb{1} - T$.

Given $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_0, \dots, \boldsymbol{\alpha}_m)$ with $\boldsymbol{\alpha}_i \in \mathbb{R}^{+*}$, we consider the periodic Sobolev multiballs $W_{\#}^{m,q}(\boldsymbol{\alpha})$ and their discrete counterparts $\mathcal{P}_{\#,n}^{m,q}(\boldsymbol{\alpha})$ defined as :

$$\begin{aligned} W_{\#}^{m,q}(\boldsymbol{\alpha}) &= \left\{ f \in L_{\#}^1([0,1] \rightarrow \mathbb{R}^d) \text{ s.t. } \|f^{(r)}\|_{L_{\#}^q([0,1])} \leq \boldsymbol{\alpha}_r \quad \forall r, 0 \leq r \leq m \right\}, \\ \mathcal{P}_{\#,n}^{m,q}(\boldsymbol{\alpha}) &= \left\{ \mathbf{p} \in \mathbb{R}^{n \times d} \text{ s.t. } n^r \|\Delta^{*r} \star \mathbf{p}\|_{\ell^q} \leq \boldsymbol{\alpha}_r \quad \forall r, 0 \leq r \leq m \right\}, \end{aligned}$$

where $L_{\#}^1$ is the set of periodic functions in L^1 and $f^{(r)}$ denotes the derivative of order r of f .

In the non-periodic case we define the Sobolev multiballs as :

$$\begin{aligned} W^{m,q}(\boldsymbol{\alpha}) &= \left\{ f \in L^1([0,1] \rightarrow \mathbb{R}^d) \text{ s.t. } \|f^{(r)}\|_{L^q([0,1])} \leq \boldsymbol{\alpha}_r \quad \forall r, 0 \leq r \leq m \right\}, \\ \mathcal{P}_n^{m,q}(\boldsymbol{\alpha}) &= \left\{ \mathbf{p} \in \mathbb{R}^{n \times d} \text{ s.t. } n^r \left(\sum_{i=r}^{n-1} \frac{1}{n} \|(\Delta^{*r} \star \mathbf{p})_i\|^q \right)^{1/q} \leq \boldsymbol{\alpha}_r \quad \forall r, 0 \leq r \leq m \right\}. \end{aligned}$$

We consider two different discretizations of curves : given a family of points \mathbf{p} the 0-spline discretization is defined by :

$$s^0(\mathbf{p}) : t \mapsto \mathbf{p}_{[nt]},$$

which simply amounts to considering the piecewise constant function with plateaus on the intervals $[\frac{i}{n}, \frac{i+1}{n}]$, $0 \leq i \leq n-1$. On the other hand the 1-spline discretization is the linear interpolation between the points, it is defined by :

$$s^1(\mathbf{p}) : t \mapsto \mathbf{p}_{[nt]} + \{nt\} (\mathbf{p}_{[nt]} - \mathbf{p}_{[nt]}),$$

where $\lfloor \bullet \rfloor$, $\lceil \bullet \rceil$ are respectively the floor and ceiling function ; we denote the decimal part of a number as : $\{nt\} = nt - [nt] \in [0,1]$. We introduce the Sobolev multiballs of 0-splines and of 1-splines as :

$$\mathcal{S}_n^{m,q}(\boldsymbol{\alpha}) = \left\{ s^0(\mathbf{p}) \quad \text{with} \quad \mathbf{p} \in \mathcal{P}_n^{m,q}(\boldsymbol{\alpha}) \right\},$$

$$\mathcal{L}_n^{m,q}(\boldsymbol{\alpha}) = \left\{ s^1(\mathbf{p}) \quad \text{with} \quad \mathbf{p} \in \mathcal{P}_n^{m,q}(\boldsymbol{\alpha}) \right\},$$

with of course $\mathcal{S}_{\sharp,n}^{m,q}(\boldsymbol{\alpha})$ and $\mathcal{L}_{\sharp,n}^{m,q}(\boldsymbol{\alpha})$ their periodic counterparts.

Finally, we specify a metric between curves. Given curves f and g from $[0, 1]$ to \mathbb{R}^d , the distance between f and g is defined as :

$$d(f, g) = \int_0^1 \|f(t) - g(t)\| dt. \quad (7.3)$$

The distance (7.3) enforces that the set of values of f and g are similar but also that their time parameterizations are close. The distance (7.3) is related to the 1-Wasserstein distance, if one considers :

$$\tilde{f}(t) = (f(t), t) \quad \text{and} \quad c((x_1, t_1), (x_2, t_2)) = \|x_1 - x_2\| + |t_1 - t_2|.$$

Denote $d_\lambda \tilde{f}$ the push-forward of the Lebesgue measure λ of $[0, 1]$ on \mathbb{R}^{d+1} , that is :

$$\text{For any Borel set } A \subset \mathbb{R}^{d+1}, \quad d_\lambda \tilde{f}(A) = \lambda(\tilde{f}^{-1}(A)).$$

Note that, for instance

$$d_\lambda \tilde{s}^0(A, B) = \sum_{i=0}^{n-1} \frac{1}{n} \delta_{\mathbf{p}_i}(A) \lambda(B \cap [\frac{i}{n}, \frac{i+1}{n}]) \quad \forall A \subset \mathbb{R}^p, B \subset \mathbb{R}.$$

Introduce the 1-Wasserstein distance between the corresponding measures $d_\lambda \tilde{f}$ and $d_\lambda \tilde{g}$. We have

$$\mathcal{W}^1(d_\lambda \tilde{f}, d_\lambda \tilde{g}) = \inf_{\gamma \in \pi} \int_{\mathbb{R}^{d+1} \times \mathbb{R}^{d+1}} c((x, t_x), (y, t_y)) d\gamma(x \times t_x, y \times t_y),$$

where π is the set of measures on $\mathbb{R}^{d+1} \times \mathbb{R}^{d+1}$ whose first and second marginals are given by $d_\lambda \tilde{f}$ and $d_\lambda \tilde{g}$ respectively. One such coupling γ is given by the time parameter of the curve, so that :

$$\mathcal{W}^1(d_\lambda \tilde{f}, d_\lambda \tilde{g}) \leq \int_0^1 \|f(t) - g(t)\| dt. \quad (7.4)$$

7.3 Main result

Introduce the Hausdorff distance between two sets of functions A and B by :

$$d_{\mathcal{H}}(A, B) = \sup_{f \in A} \inf_{g \in B} d(f, g) + \sup_{g \in B} \inf_{f \in A} d(f, g),$$

where d is defined in (7.3). Our main theorems are stated as follows,

Theorem 7.3.1. *If $m \geq 1$, the Hausdorff distance between the multi-balls of radii α of zero-order periodic splines and the multi-balls of radii α of periodic Sobolev functions is bounded by $\frac{\kappa\alpha}{n}$.*

Theorem 7.3.2. *If $m \geq 1$, the Hausdorff distance between the multi-balls of radii α of zero-order non-periodic splines and the multi-balls of radii α of non-periodic Sobolev functions is bounded by $\frac{\kappa\alpha}{n}$.*

Theorem 7.3.3. *If $m \geq 2$, the Hausdorff distance between the multi-balls of radii α of first-order periodic splines and the multi-balls of radii α of periodic Sobolev functions is bounded by $\frac{\kappa\alpha}{n^2}$.*

Remark 7.3.1. *More precisely, Theorem 7.3.3 states that, if $m \geq 2$ for any $f \in W_{\sharp}^{m,q}(\alpha)$, there exists $\mathbf{p} \in \mathcal{P}_{\sharp,n}^{m,q}(\alpha)$ such that*

$$\mathcal{W}_1(f, s^1(\mathbf{p})) \leq \frac{\kappa\alpha}{n^2}, \quad (7.5)$$

and for any $\mathbf{p} \in \mathcal{P}_{\sharp,n}^{m,q}(\alpha)$, there exists $f \in W_{\sharp}^{m,q}(\alpha)$ such that (7.5) holds.

We first describe the approximant, for the discrete to continuous case, in the following proposition.

Proposition 7.3.1. *Given a sequence of points \mathbf{p} , define the function $f_{\mathbf{p}}$ by*

$$\forall t \in [0, 1[, \quad f_{\mathbf{p}}(t) = \sum_{i=0}^{n-1} g_i(nt - i), \quad (7.6)$$

$$\text{with } g_i(x) = \sum_{k=0}^m (C^{m-k} \star \Delta^{\star k} \star \mathbf{p})_i \frac{x^k}{k!} \chi_{0 \leq x < 1},$$

with χ_A the indicator function of the set A . Then the two following properties are equivalent :

— For each r and i the coefficient C_i^r satisfies

$$\forall r \geq 1, \quad C_i^r = \frac{E_{i-1}^r}{r!} \quad \text{and} \quad C^0 = \mathbf{1} \quad (7.7)$$

where E_i^k , $k \geq 1$ is the i -th Eulerian number of degree k .

— The curve $f_{\mathbf{p}}$ is a spline of order m , $m - 1$ time continuously differentiable whose m^{th} order derivative is given by

$$f_{\mathbf{p}}^{(m)}(t) = n^m (\Delta^{\star m} \star \mathbf{p})_i \quad \text{for } t \in \left] \frac{i}{n}, \frac{i+1}{n} \right[. \quad (7.8)$$

In the course of the proof of Proposition 7.3.1 we prove the following seemingly new recurrence relationship between the Eulerian numbers.

Proposition 7.3.2. *The Eulerian numbers are solution to each of the two recurrence equations :*

$$E_i^m = \sum_{k=1}^m \binom{m}{k} \sum_{l=0}^{k-1} (-1)^l \binom{k-1}{l} E_{i-1}^{m-k} \quad (7.9)$$

$$E_{i-1}^m = \sum_{k=0}^m \binom{m}{k} \sum_{l=0}^k (-1)^{l-1} \binom{k-1}{l-1} E_{i-l}^{m-k} \quad (7.10)$$

Propositions 7.3.1 and 7.3.2. Let $f_{\mathbf{p}}$ be the function defined in Equation (7.6). It is trivial to see that Equation (7.8) is true if C^0 is the convolution identity kernel. It remains to check the regularity at the connections, indeed the l -th derivatives on the right and on the left of the spline $f_{\mathbf{p}}$ have to be equal at each connection, that is, for each $i \in \llbracket 0, n-1 \rrbracket$ and $l \in \llbracket 0, m-1 \rrbracket$

$$\lim_{t \rightarrow 0} g_{i+1}^{(l)}(t) = \lim_{t \rightarrow 1} g_i^{(l)}(t).$$

This gives the following equation :

$$\begin{aligned} (C^{m-l} \star \Delta^{\star l} \star \mathbf{p})_{i+1} &= \sum_{k=l}^m \frac{1}{(k-l)!} (C^{m-k} \star \Delta^{\star k} \star \mathbf{p})_i \\ &= \sum_{k=0}^{m-l} \frac{1}{k!} (C^{m-l-k} \star \Delta^{\star(k+l)} \star \mathbf{p})_i, \end{aligned} \quad (7.11)$$

Since \mathbf{p} is arbitrary, it can be removed, this yields :

$$\sum_{k=0}^s \frac{1}{k!} (T \star C^{s-k} \star \Delta^{\star k}) = C^s, \quad 0 \leq s \leq m.$$

Subtracting the first term of the sum from the right hand side one has

$$\sum_{k=1}^s \frac{1}{k!} \left(T \star C^{s-k} \star \Delta^{\star k} \right)_i = (C^s - T \star C^s)_i = C_i^s - C_{i-1}^s = (\Delta \star C^s)_i. \quad (7.12)$$

Finally, $f_{\mathbf{p}}$ is $m-1$ continuously differentiable if and only if the coefficient C verify the recursive formula :

$$\forall 1 \leq s \leq m, \quad \sum_{k=1}^s \frac{1}{k!} \left(T \star C^{(s-k)} \star \Delta^{\star k-1} \right) = C^s. \quad (7.13)$$

We turn our attention to solving Equation (7.13) and to obtain that for all s ,

$$C_i^s = \frac{E_i^s}{s!} \quad \text{with} \quad E_i^s = \sum_{k=0}^i (-1)^k \binom{s+1}{k} (i-k)^s \quad \text{and} \quad E_i^0 = 1 \text{ iff } i = 0,$$

where E_{i+1}^k , $k \geq 1$ is the i -th Eulerian number [31] of degree k . Suppose the formula for C^r is valid up to $r = s-1$, replacing C^{s-k} by its value in Equation (7.13) and replacing $\Delta^{\star k}$ by its expression (7.2) one obtains :

$$s!C_i^s = \sum_{k=1}^s \binom{s}{k} \sum_{l=0}^{k-1} (-1)^l \binom{k-1}{l} \sum_{r=0}^{i-1-l} (-1)^r \binom{s+1-k}{r} (i-1-l-r)^{s-k}.$$

Changing the index r by $q = r + l$ and extending the summation of q from l to 0 one gets

$$s!C_i^s = \sum_{q=0}^{i-1} (-1)^q \sum_{k=1}^s \binom{s}{k} (i-1-q)^{s-k} \sum_{l=0}^{k-1} \binom{k-1}{l} \binom{s+1-k}{q-l}.$$

Summing in l the right hand side, one has

$$s!C_i^s = \sum_{q=0}^{i-1} (-1)^q \binom{s}{q} \sum_{k=1}^s \binom{s}{k} (i-q-1)^{s-k}.$$

Now using the binomial theorem,

$$s!C_i^s = \sum_{q=0}^{i-1} (-1)^q \binom{s}{q} \left((i-q)^s - (i-(q+1))^s \right),$$

an Abel transform gives

$$s!C_i^s = \sum_{q=1}^{i-1} (-1)^q (i-q)^s \left[\binom{s}{q} + \binom{s}{q-1} \right] + 1.$$

Finally, Pascal's rule yields formula $s!C_i^s = E_i^s$ and the proof of Proposition 7.3.1 is finished. \square

In order to prove Proposition 7.3.2, we rewrite Equations (7.11),(7.12) where we substitute C with the corresponding Eulerian number E using formula (7.7). \square

Proposition 7.3.3. *The periodic spline $f_{\mathbf{p}}$ satisfying Equalities (7.7),(7.8) can be expressed in the B-Spline basis. It turns out that the control points are exactly the \mathbf{p}_i :*

$$f_{\mathbf{p}}(t) = \sum_{i \in \mathbb{Z}} B^m(nt - i) \mathbf{p}_i \quad (7.14)$$

$$\text{with } B^m(x) = \frac{1}{m!} \sum_{k=0}^{m+1} (-1)^k \binom{m+1}{k} (x-k)_+^m,$$

where $(a)_+^m$ denotes the m -th power of the positive part of a . Note that the formula defined is periodic and the support of B^m is $[0, m+1[$, so that, for a fixed t , $f_{\mathbf{p}}$ is a finite sum.

Proposition 7.3.3. The function which satisfies condition (7.8) is defined up to the addition of a polynomial of degree $m-1$. This polynomial has to be periodic then it is reduced to a constant. Finally the homogeneity of $f_{\mathbf{p}}$ in \mathbf{p} yields its uniqueness.

It remains to show that the function defined in (7.14) satisfies condition (7.8). Using the differentiation formula of the equispaced B-spline [120], the k -th derivative of $f_{\mathbf{p}}(t)$ is given by :

$$f_{\mathbf{p}}^{(k)}(t) = \sum_{i \in \mathbb{Z}} \mathbf{p}_i n^k \sum_{j=0}^k (-1)^j \binom{k}{j} B^{m-k}(nt - i - j)$$

Since $B^0(t) = \chi_{t \in [0,1[}$ m -th derivative reads as :

$$f_{\mathbf{p}}^{(m)}(t) = \sum_{i \in \mathbb{Z}} \mathbf{p}_i n^m \sum_{j=0}^m (-1)^j \binom{m}{j} \chi_{t \in [\frac{i+j}{n}, \frac{i+1+j}{n}[}$$

$$f_{\mathbf{p}}^{(m)}(t) = \sum_{i \in \mathbb{Z}} n^m \sum_{a=0}^m (-1)^a \binom{m}{a} \mathbf{p}_{i-a} \chi_{t \in [\frac{a}{n}, \frac{a}{n}]}.$$

For t in $[\frac{i}{n}, \frac{i+1}{n}]$ one has

$$f_{\mathbf{p}}^{(m)}(t) = n^m \sum_{k=0}^m (-1)^k \binom{m}{k} \mathbf{p}_{i-k} = n^m (\Delta^{*m} \star \mathbf{p})_i$$

which allows to conclude. \square

7.4 Proof of the theorems

This section deals with the proofs of the main theorems. It is subdivided into 4 sections. In Section 7.4.1, we introduce some useful results and operators used throughout the rest of the proof. In Section 7.4.2, we construct the spline approximation when the continuous curve is given and show that the distance between the spline and the continuous curve is bounded with the correct rate with respect to n . In Section 7.4.3, we construct a continuous curve when the spline approximation is given. In Section 7.4.4 the distance between constructed continuous curve and the given spline approximation is proven with the correct rate but the continuous curve does not belong to the correct multi-ball. Finally in Section 7.4.5, we gather the results of the different sections and prove the main theorems.

7.4.1 Notations and technical lemmas

In the following, we make extensive use of the shift operator σ_m defined as

$$\begin{cases} (\sigma_m \star \mathbf{p})_i = \mathbf{p}_{i+\frac{m+1}{2}} & \text{if } m \text{ is odd} \\ (\sigma_m \star \mathbf{p})_i = \frac{1}{2} (\mathbf{p}_{i+m/2} + \mathbf{p}_{i+m/2+1}) & \text{if } m \text{ is even.} \end{cases} \quad (7.15)$$

Moreover, we need a notion of support of the convolution kernel, this notion is well suited to the non-periodic case and is only useful in this context.

Lemma 7.4.1. *Let $\alpha, \beta \in \mathbb{N}$ with $\alpha + \beta < n$, we say that a kernel K has support in $[-\alpha, \beta]$ if $K_i = 0$ for each $\beta < i < n - \alpha$. For such a kernel K ,*

we have, for all $A \in \mathbb{R}^n$, for all $a \geq \beta$ and $b < n - \alpha$,

$$\left(\sum_{i=a}^b \|(K \star A)_i\|^q \right)^{\frac{1}{q}} \leq n \|K\|_{\ell^1} \left(\sum_{i=a-\beta}^{b+\alpha} \|A_i\| \right)^{\frac{1}{q}}.$$

Finally we introduce the operator Δ^{-1} , the inverse of the operator Δ .

Lemma 7.4.2. *Let $\alpha, \beta \in \mathbb{N}$. If A has support in $[-\alpha, \beta]$ and verifies $\sum_{i=0}^{n-1} A_i = 0$, define $\Delta^{-1}(A)$ as*

$$\Delta^{-1}(A)_i = \sum_{j=0}^i A_j - \sum_{j=0}^{\beta} A_j.$$

Then $\Delta \star \Delta^{-1}(A) = A$. Moreover $\Delta^{-1}(A)$ has support in $[-\alpha, \beta - 1]$ and $\|\Delta^{-1}(A)\|_{\ell^1} \leq (\beta + \alpha) \|A\|_{\ell^1}$.

Next we gather some results about the Eulerian numbers in the following lemma.

Lemma 7.4.3. *For $m \geq 2$ the kernel C^m sums up to 1 and have support in $[0, m]$ and the kernel $C^m \star \sigma_m$ is symmetric. Moreover if*

$$A = C^m \star \sigma_m - \mathbb{1},$$

then $\Delta^{-2}(A)$ exists. For $m = 0, 1$, then $A = 0$.

Lemma 7.4.3. First we recall the following well-known properties of Eulerian numbers, valid for $m \geq 1$, see [31]

$$E_k^m = E_{m-k+1}^m, \quad \text{and} \quad E_k^m = (m - k + 1)E_{k-1}^{m-1} + kE_k^{m-1}$$

We prove $\sum_{k=1}^{m+1} C_k^m = \sum_{k=0}^m \frac{E_k^m}{m!} = 1$ by an induction on m .

$$\begin{aligned} \sum_{1 \leq k \leq m} E_k^m &= \sum_{1 \leq k \leq m} (m - k + 1)E_{k-1}^{m-1} + kE_k^{m-1} \\ &= \sum_{k=1}^{m-1} (m - k)E_k^{m-1} + kE_k^{m-1} = m \sum_{k=1}^{m-1} E_k^{m-1} = m! \quad \square \end{aligned}$$

We now study the symmetry of $C^m \star \sigma_m$. If m is odd this property is a direct consequence $E_i^m = E_{m-i+1}^m$. If m is even, simply notice that

$$(C^m \star \sigma_m)_i = \frac{1}{2} \left(C_{i+\frac{m}{2}}^m + C_{i+\frac{m}{2}+1}^m \right)$$

$$= \frac{1}{2} \left(C_{-i+\frac{m}{2}+1}^m + C_{-i+\frac{m}{2}}^m \right) = (C^m \star \sigma_m)_{-i}.$$

The coefficients of A sum up to 0 and A has support in $[-\lfloor \frac{m}{2} \rfloor, \lfloor \frac{m}{2} \rfloor]$ so that $\Delta^{-1}(A)$ exists. For n large enough, we now prove that the symmetry of A ensures that the coefficients of $\Delta^{-1}(A)$ sum up to 0. For that purpose, for each j , denote $u = n - j$ so that $A_u = A_j$.

$$\begin{aligned} \sum_{i=0}^{n-1} \Delta^{-1}(A)_i &= \sum_{i=0}^{n-1} \left(\sum_{j=0}^i A_j \right) - n \sum_{j=0}^{\beta} A_j = \sum_{j=0}^{n-1} (n-j) A_j - n \sum_{j=0}^{\beta} A_j \\ &= \frac{1}{2} \left(\sum_{u=1}^n u A_u + \sum_{j=0}^{n-1} (n-j) A_j \right) - \frac{n}{2} \left(\sum_{j=0}^{\beta} A_j + \sum_{u=n-\beta}^n A_u \right) \\ &= \frac{1}{2} \left(\sum_{u=1}^n u A_u - \sum_{j=0}^{n-1} j A_j \right) - \frac{n}{2} \left(\sum_{j=0}^{n-1} A_j + A_n \right) \\ &= \frac{n}{2} A_n - \frac{n}{2} A_n = 0. \end{aligned}$$

Hence $\Delta^{-2}(A)$ exists and has support in $[-\lfloor \frac{m}{2} \rfloor, \lfloor \frac{m}{2} \rfloor - 1]$. Moreover, we have

$$\|\Delta^{-2}(A)\|_{\ell^1} \leq \frac{\kappa_m}{n} \quad \text{for } n \geq m.$$

□

7.4.2 Approximation of function by splines

We now describe the approximating spline for a continuous curve in the periodic and non-periodic case. We prove that the approximations belong to the correct Sobolev multi-balls. We also prove that the \mathcal{W}^1 -distance between this approximation and the continuous curve is bounded with the correct rates.

Proposition 7.4.1. *Let $f \in W_{\#}^{m,q}(\boldsymbol{\alpha})$, (resp. $f \in W^{m,q}(\boldsymbol{\alpha})$), define $\mathbf{p} \in \mathbb{R}^{n \times d}$ as $\mathbf{p}_i = f\left(\frac{i}{n}\right)$ for all $i = 0 \dots n-1$.*

- Then $\mathbf{p} \in \mathcal{P}_{\#,n}^{m,q}(\boldsymbol{\alpha})$ (resp. $\mathcal{P}_n^{m,q}(\boldsymbol{\alpha})$)
- If $m \geq 1$, then $d(f, s^0(\mathbf{p})) \leq \frac{\boldsymbol{\alpha}_1}{n}$.
- If $m \geq 2$, then $d(f, s^1(\mathbf{p})) \leq \frac{\boldsymbol{\alpha}_2}{n^2}$.

This proposition states that the distance from the multi-balls of Sobolev functions to the set of splines behaves exactly as stated in Theorems 7.3.1, 7.3.2 and 7.3.3.

Let $f \in W_{\#}^{m,q}(\boldsymbol{\alpha})$ (resp. $f \in W^{m,q}(\boldsymbol{\alpha})$) and set $\mathbf{p}_i = f(\frac{i}{n})$, we first prove that $\mathbf{p} \in \mathcal{P}_{\#,n}^{m,q}(\boldsymbol{\alpha})$ (resp. $\mathbf{p} \in \mathcal{P}_n^{m,q}(\boldsymbol{\alpha})$).

Démonstration. For any $k \leq m$, let $i \geq k$ in the case $f \in W^{m,q}(\boldsymbol{\alpha})$ and let i be arbitrary in the case $f \in W_{\#}^{m,q}(\boldsymbol{\alpha})$, we have

$$(\Delta^{*k} \star \mathbf{p})_i = \frac{1}{n^k} \int_{s_1=i-1}^i \int_{s_2=s_1-1}^{s_1} \cdots \int_{s_k=s_{k-1}-1}^{s_{k-1}} f^{(k)}\left(\frac{s_k}{n}\right) ds_k \cdots ds_2 ds_1$$

Notice that s_k is integrated on the interval $[i-k, i]$. We use a Fubini theorem, and in the periodic case, we use a change of variable $s_k + k \rightarrow s_k$ to obtain :

$$\begin{aligned} \|n^k (\Delta^{*k} \star \mathbf{p})_i\| &\leq \int_{s_k=0}^n \int_{s_{k-1}=s_k}^{s_k+1} \cdots \int_{s_1=s_2}^{s_2+1} \|f^{(k)}\left(\frac{s_k}{n}\right)\| \chi_{s_1 \in [i-1, i]} ds_1 \cdots ds_{k-1} ds_k \\ &\leq \int_{s_k=0}^n \left\| f^{(k)}\left(\frac{s_k}{n}\right) \right\| \theta_i(s_k) ds_k. \end{aligned}$$

In the periodic case, the functions θ_i are functions that verify

$$\forall s, 0 \leq \theta_i(s) \leq 1, \quad \sum_{i=K}^n \theta_i(s) = 1 \quad \text{and} \quad \int_0^n \theta_i(s) ds \leq 1 \quad \forall i \geq K,$$

where $K = 0$ in the periodic case and $K = k$ in the non-periodic case. By Jensen's inequality, we have :

$$\begin{aligned} \|n^k (\Delta^{*k} \star \mathbf{p})_i\|^q &\leq \left(\int_{s_k=0}^n \left\| f^{(k)}\left(\frac{s_k}{n}\right) \right\| \frac{\theta_i(s_k)}{\|\theta_i\|_{L^1}} ds_k \right)^q \|\theta_i\|_{L^1}^q \\ &\leq \left(\int_{s_k=0}^n \left\| f^{(k)}\left(\frac{s_k}{n}\right) \right\|^q \theta_i(s_k) ds_k \right) \|\theta_i\|_{L^1}^{q-1} \\ &\leq \int_{s_k=0}^n \left\| f^{(k)}\left(\frac{s_k}{n}\right) \right\|^q \theta_i(s_k) ds_k \end{aligned}$$

The k -th semi-norm of \mathbf{p} is then bounded by :

$$n^k \|\Delta^{*k} \star \mathbf{p}\|_{\ell^q} \leq \left(\sum_{i=K}^{n-1} \frac{1}{n} \int_{s_k=0}^n \left\| f^{(k)}\left(\frac{s_k}{n}\right) \right\|^q \theta_i(s_k) ds_k \right)^{1/q}$$

$$\leq \left(\int_{s=0}^n \left\| f^{(k)} \left(\frac{s}{n} \right) \right\|^q \frac{ds}{n} \right)^{1/q} = \alpha_k \quad (7.16)$$

This proves that $\mathbf{p} \in \mathcal{P}_{\sharp, n}^{m, q}(\alpha)$ in the periodic case, and $\mathbf{p} \in \mathcal{P}_n^{m, q}(\alpha)$ in the non-periodic case. \square

The second item of Proposition 7.4.1 involves bounding the distance between f and $s^0(\mathbf{p})$. In the periodic and non-periodic case, we have

$$\begin{aligned} d(f, s^0(\mathbf{p})) &\leq \int_0^1 \left\| f(t) - f\left(\frac{\lfloor tn \rfloor}{n}\right) \right\| dt \\ &\leq \sum_{i=0}^{n-1} \int_{\frac{i}{n}}^{\frac{i+1}{n}} \left\| \int_{\frac{\lfloor tn \rfloor}{n}}^t f'(s) ds \right\| dt \leq \frac{1}{n} \sum_{i=0}^{n-1} \int_{\frac{i}{n}}^{\frac{i+1}{n}} \|f'(s)\| ds \\ &= \frac{1}{n} \|f'\|_{L^1([0,1])} \leq \frac{1}{n} \|f'\|_{L^q([0,1])} \leq \frac{\alpha_1}{n} \end{aligned}$$

The last statement of Proposition 7.4.1 amounts to bounding the distance between f and $s^1(\mathbf{p})$, assuming that $m \geq 2$.

Introducing for each i the point $m_i = \frac{i+1/2}{n}$, and performing a Taylor expansion around this point, we have, for every $t \in [\frac{-1}{2n}, \frac{1}{2n}]$:

$$\begin{aligned} f(m_i + t) &= f(m_i) + t f'(m_i) + \int_{m_i}^{t+m_i} f''(s)(m_i + t - s) ds \\ s^1(\mathbf{p})(m_i + t) &= \left(\frac{1}{2} - nt\right) f\left(\frac{i}{n}\right) + \left(nt + \frac{1}{2}\right) f\left(\frac{i+1}{n}\right) \\ &= f(m_i) + t f'(m_i) + \left(\frac{1}{2} - nt\right) \int_{m_i}^{\frac{i}{n}} f''(s)(m_i - \frac{1}{2n} - s) ds \\ &\quad + \left(nt + \frac{1}{2}\right) \int_{m_i}^{\frac{i+1}{n}} f''(s)(m_i + \frac{1}{2n} - s) ds \\ (f - s^1(\mathbf{p}))(t + m_i) &= \int_{m_i}^{t+m_i} f''(s) \underbrace{(m_i - s + t)}_{\beta(t,s)} ds \\ &\quad + \int_{\frac{i}{n}}^{\frac{i+1}{n}} f''(s) \left(\underbrace{nt(m_i - s) + \frac{1}{4n} + \left(\frac{m_i - s}{2} + \frac{t}{2}\right)(\chi_{s \geq m_i} - \chi_{s \leq m_i})}_{\gamma(t,s)} \right) ds \end{aligned}$$

For $t \in [\frac{-1}{2n}, \frac{1}{2n}]$ and the s under consideration, we have $|\beta(t, s)| \leq |t|$ and

$|\gamma(t, s)| \leq |t| + \frac{1}{2n}$, so that

$$\|(f - s^1(\mathbf{p}))(t + m_i)\| \leq \left(\frac{1}{2n} + 2|t|\right) \int_{\frac{i}{n}}^{\frac{i+1}{n}} \|f''(s)\| ds$$

Finally we have

$$\begin{aligned} d(f, s^1(\mathbf{p})) &\leq \int_0^1 \|f(t) - s^1(\mathbf{p})(t)\| dt \\ &\leq \sum_{i=0}^{n-1} \int_{t=-\frac{1}{2n}}^{\frac{1}{2n}} \left(\frac{1}{2n} + 2|t|\right) dt \int_{\frac{i}{n}}^{\frac{i+1}{n}} \|f''(s)\| ds = \frac{1}{n^2} \int_0^1 \|f''(s)\| ds \\ &\leq \frac{1}{n^2} \|f''(s)\|_{L^q([0,1])} = \frac{\alpha_2}{n^2}. \end{aligned}$$

Thus the proof of Proposition 7.4.1 is complete. This calculus holds both for periodic and non periodic functions.

7.4.3 Approximation of splines by functions

Now that the spline are known, this section is devoted to the construction of the continuous curve with the correct \mathcal{W}^1 -distance and the correct Sobolev constants α .

7.4.3.1 Construction of the approximant

As announced, we have the following proposition

Proposition 7.4.2. *Let $\mathbf{p} \in \mathcal{P}_{\sharp, n}^{m, q}(\alpha)$, let $f_{\sigma_m \star \mathbf{p}}$ be as defined in Proposition 7.3.1, then there exists κ_α a constant that depends only on α such that the spline $f_{\sigma_m \star \mathbf{p}}$ belongs to $W_{\sharp}^{m, q}\left(\left(1 + \frac{\kappa_\alpha}{n^2}\right)\alpha\right)$.*

The shift kernel σ_m defined in Equation (7.15) either drifts the indexes of \mathbf{p} or of its mid points $\frac{1}{2}(\mathbf{p} + T \star \mathbf{p})$ depending on the parity of the desired spline. Notice that $\|\sigma_m\|_{\ell^1} = n^{-1}$ so that for any $\mathbf{p} \in \mathcal{P}_{\sharp, n}^{m, q}(\alpha)$, we have $\sigma_m \star \mathbf{p} \in \mathcal{P}_{\sharp, n}^{m, q}(\alpha)$ by virtue of Young's convolution inequality (7.1).

The l -th derivative of the spline $f_{\sigma_m \star \mathbf{p}}$ is given by :

$$f_{\sigma_m \star \mathbf{p}}^{(l)}(t) = n^l g_i^{(l)}\left(nt - \frac{i}{n}\right) \chi_{t \in [\frac{i}{n}, \frac{i+1}{n}]}$$

For every i in $\llbracket 0, n-1 \rrbracket$, the l -th derivative of g_i reads as :

$$\forall t \in [0, 1], g_i^{(l)}(t) = \sum_{k=l}^m \left(C^{m-k} \star \Delta^{\star k} \star \sigma_m \star \mathbf{p} \right)_i \frac{t^{k-l}}{(k-l)!}$$

We first deal with the case $l = m$. In this case

$$\|f^{(m)}\|_{L^q} = n^m \left(\sum_{i=0}^{n-1} \frac{1}{n} \int_0^1 \left\| \left(C^0 \star \Delta^{\star m} \star \sigma_m \star \mathbf{p} \right)_i \right\|^q dt \right)^{1/q} = \alpha_m.$$

Now suppose that $l \leq m-1$. Notice that $T = (0, 1, 0, \dots) = Id - \Delta$, using Lemma 7.4.2 the coefficients of C^{m-l-1} and of C^{m-l} sum up to one. Hence the operator $A = \Delta^{-1}(C^{m-l-1} - C^{m-l} \star T^{-1})$ exists and there exists a constant $\kappa_{m,l}$ that depends only on m and l such that $\|A\|_{\ell^1} \leq \frac{\kappa_{m,l}}{n}$ and

$$C^{m-l-1} = C^{m-l} \star T^{-1} + A \star \Delta, \quad .$$

Note also that in the case $l = m-1$, one has $A = 0$. Set $\mathbf{q} = \sigma_m \star \mathbf{p}$, by the triangle inequality, we have :

$$\begin{aligned} \|f^{(l)}\|_{L^q} &= n^l \left(\sum_{i=0}^{n-1} \frac{1}{n} \int_0^1 \left\| \sum_{k=l}^m \left(C^{m-k} \star \Delta^{\star k} \star \mathbf{q} \right)_i \frac{t^{k-l}}{(k-l)!} \right\|^q dt \right)^{1/q} \\ &\leq n^l \left(\sum_{i=0}^{n-1} \frac{1}{n} \int_0^1 \left\| \sum_{k=l}^{l+1} \left(C^{m-k} \star \Delta^{\star k} \star \mathbf{q} \right)_i \frac{t^{k-l}}{(k-l)!} - t \left(A \star \Delta^{\star(l+2)} \star \mathbf{q} \right)_i \right\|^q dt \right)^{1/q} : \beta \\ &+ n^l \left(\sum_{i=0}^{n-1} \frac{1}{n} \int_0^1 \left\| \sum_{k=l+2}^m \left(C^{m-k} \star \Delta^{\star k} \star \mathbf{q} \right)_i \frac{t^{k-l}}{(k-l)!} + t \left(A \star \Delta^{\star(l+2)} \star \mathbf{q} \right)_i \right\|^q dt \right)^{1/q} : \gamma \end{aligned}$$

We claim that the first term, β , is bounded by α_l and that the second term, γ , scales as $\mathcal{O}(n^{-2})$. Indeed for the term β , we have, since $\Delta = Id - T$:

$$C^{m-l-1} \star \Delta = -C^{m-l} + C^{m-l} \star T^{-1} + A \star \Delta^{\star 2}$$

so that

$$\begin{aligned} \beta &= n^l \left(\sum_{i=0}^{n-1} \frac{1}{n} \int_0^1 \left\| (1-t) \left(C^{m-l} \star \Delta^{\star l} \star \mathbf{q} \right)_i + t \left(C^{m-l} \star \Delta^{\star l} \star \mathbf{q} \right)_{i+1} \right\|^q dt \right)^{1/q} \\ &\leq n^l \left(\sum_{i=0}^{n-1} \frac{1}{n} \int_0^1 (1-t) \left\| \left(C^{m-l} \star \Delta^{\star l} \star \mathbf{q} \right)_i \right\|^q + t \left\| \left(C^{m-l} \star \Delta^{\star l} \star \mathbf{q} \right)_{i+1} \right\|^q dt \right)^{1/q}. \end{aligned}$$

A change of index in i allows us to conclude

$$\beta \leq n^l \|C^{m-l} \star \Delta^{\star l} \star \sigma_m \star \mathbf{p}\|_{\ell^q}. \quad (7.17)$$

By virtue of Young's inequality (7.1) and $\|C^{m-l} \star \sigma_m\|_{\ell^1} = n^{-1}$, we have $\beta \leq \alpha_l$.

In order to deal with the second term γ , first assume that $l \leq m - 2$. Indeed in the case $l = m - 1$, we have $A = 0$ and then $\gamma = 0$ and nothing is to be proven. In the case $l \leq m - 2$, bound t by 1, introduce the operator

$$Q = \sum_{k=l+2}^m \frac{1}{(k-l)!} \left| C^{m-k} \star \Delta^{\star(k-(l+2))} \right| + |A|, \quad (7.18)$$

and note that there exists a constant $\kappa_{m,l}$ that depends only on m and l such that $\|Q\|_{\ell^1} \leq \frac{\kappa_{m,l}}{n}$. Then Young's inequality yields :

$$|\gamma| \leq n^l \|Q \star \Delta^{\star(l+2)} \star \mathbf{q}\|_{\ell^q} \leq \kappa_{m,l} n^l \|\Delta^{\star(l+2)} \star \mathbf{p}\|_{\ell^q} \leq \kappa_{m,l} \frac{\alpha_{l+2}}{n^2}.$$

Hence for any $l \leq m - 1$, we have

$$\|f^{(l)}\|_{L^q} \leq \alpha_l + \kappa_{m,l} \frac{\alpha_{l+2}}{n^2},$$

and for $l = m$ or $l = m - 1$, we have

$$\|f^{(l)}\|_{L^q} \leq \alpha_l.$$

Lemma 7.4.4. *Let $m \geq 1$ and $p \in \mathcal{P}_n^{m,q}(\alpha)$.*

For $\theta = 1 - \frac{20m}{n}$ and $\tau = \frac{10m}{n}$ define :

$$\tilde{f}_{\mathbf{p}}(t) = f_{\sigma_m \star \mathbf{p}}(\theta t + \tau),$$

where $f_{\sigma_m \star \mathbf{p}}$ is defined in Proposition 7.3.1, then $\tilde{f}_{\mathbf{p}} \in W^{m,q}(\alpha + \frac{\kappa_\alpha}{n^2})$

Lemma 7.4.4. The differentials of $\tilde{f}_{\mathbf{p}}$ are given by :

$$\tilde{f}_{\mathbf{p}}^{(l)}(t) = (\theta n)^l g_i^{(l)}(n\theta t + n\tau - i) \chi_{t \in [i\theta, (i+1)\theta]},$$

the Sobolev semi-norm of \tilde{f} can be written as :

$$\|\tilde{f}_{\mathbf{p}}^{(l)}\|_{L^q} = (n\theta)^l \left(\sum_{i=10m}^{n-10m} \frac{1}{n\theta} \int_0^1 \left\| \sum_{k=l}^m (C^{m-k} \star \Delta^{\star k} \star \sigma_m \star \mathbf{p})_i \frac{t^{k-l}}{(k-l)!} \right\|^q dt \right)^{1/q}$$

$$\leq \theta^{-\frac{1}{q}} \tilde{\beta} + \theta^{-\frac{1}{q}} \tilde{\gamma},$$

where we used $|\theta| < 1$ to obtain the last bound and where the variables $\tilde{\beta}$ and $\tilde{\gamma}$ are similar to β and γ defined in proof of Proposition 7.4.2, other than their sums in i range from $10m$ to $n - 10m$.

We can then follow the same outline of the proof of Proposition 7.4.2; using Lemma 7.4.1 we can verify that the σ_m shift does not interfere with non-periodicity of \mathbf{p} , owing to the sufficiently large buffer τ . Then, one has similar bounds :

$$\|\tilde{f}_{\mathbf{q}}^{(l)}\|_{L^q} \leq \theta^{-\frac{1}{q}} \left(\alpha_l + \kappa_{m,l} \frac{\alpha_{l+2}}{n^2} \right). \quad (7.19)$$

Now using that $\theta^{-1}q \leq 1 + \kappa/n$, one can conclude that

$$\|\tilde{f}_{\mathbf{q}}^{(l)}\|_{L^q} \leq \alpha_l + \frac{\kappa_{m,l}\alpha}{n}.$$

□

7.4.4 Wasserstein distance

It remains to bound the distance d between the piecewise constant or linear discretization and f the continuous approximant built with the vector \mathbf{p} .

Lemma 7.4.5. *Let $m \geq 1$ and $p \in \mathcal{P}_{\sharp,n}^{m,q}(\alpha)$ and let $f_{\sigma_m \star \mathbf{p}}$ be defined as in Proposition 7.3.1, then*

$$d(f_{\sigma_m \star \mathbf{p}}, s^0(\mathbf{p})) \leq \frac{\kappa \alpha}{n}.$$

The distance d between $f_{\sigma_m \star \mathbf{p}}$ and $s^0(\mathbf{p})$ is bounded by :

$$d(f_{\sigma_m \star \mathbf{p}}, s^0(\mathbf{p})) = \sum_{i=0}^{n-1} \int_{\frac{i}{n}}^{\frac{i+1}{n}} \left\| g_i \left(nt - \frac{i}{n} \right) - \mathbf{p}_i \right\| dt \quad (7.20)$$

$$= \sum_{i=0}^{n-1} \frac{1}{n} \int_0^1 \|g_i(u) - \mathbf{p}_i\| du. \quad (7.21)$$

Now using the triangle inequality, one has

$$\|g_i(t) - \mathbf{p}_i\| \leq \|(C^m \star \sigma_m \star \mathbf{p})_i - \mathbf{p}_i\|$$

$$+ \sum_{k=1}^m \left\| \left(C^{m-k} \star \Delta^{\star k} \star \sigma_m \star \mathbf{p} \right)_i \frac{t^k}{k!} \right\| \quad (7.22)$$

Integrating in t at the first line and summing in i at the second line, allows us to use Young's inequality for the third line given the fact that $\|C^{m-k} \star \sigma_m\|_{\ell^1} \leq 1/n$. Now for the last line, using that the ℓ^1 -norm is lower than ℓ^p -norm (by Jensen's inequality), one can conclude that second term of (7.22) is bounded by :

$$\begin{aligned} & \sum_{i=0}^{n-1} \frac{1}{n} \int_0^1 \left\| \sum_{k=1}^m \left(C^{m-k} \star \Delta^{\star k} \star \sigma_m \star \mathbf{p} \right)_i \frac{t^k}{k!} \right\| dt \\ & \leq \sum_{i=0}^{n-1} \frac{1}{n} \sum_{k=1}^m \left\| \left(C^{m-k} \star \Delta^{\star k} \star \sigma_m \star \mathbf{p} \right)_i \frac{1}{(k+1)!} \right\| \\ & \leq \sum_{k=1}^m \left\| \left(\Delta^{\star k} \star \mathbf{p} \right) \right\|_{\ell^p} \frac{1}{(k+1)!} \\ & \leq \frac{\alpha_1}{n} + \frac{\kappa_\alpha}{n^2}. \end{aligned} \quad (7.23)$$

It remains to deal with the first term appearing in inequality (7.22) which can be rewritten as

$$\| (C^m \star \sigma_m \star \mathbf{p})_i - \mathbf{p}_i \| = \| K \star \mathbf{p}_i \| \quad \text{with } K = C^m \star \sigma_m - \mathbf{1}.$$

Notice that K sums up to zero, so that there exists $A = \Delta^{-1}(K)$ with $\|A\|_{\ell^1} \leq \frac{\kappa_m}{n}$ for some constant κ_m . As a result,

$$\begin{aligned} \sum_{i=0}^{n-1} \frac{1}{n} \int_0^1 \| (C^m \star \sigma_m \star \mathbf{p})_i - \mathbf{p}_i \| dt & \leq \| K \star \mathbf{p} \|_{\ell^1} \leq \| K \star \mathbf{p} \|_{\ell^q} \\ & \leq \kappa_m \| \Delta \mathbf{p} \|_{\ell^q} \leq \frac{\alpha_1}{n} \kappa_m \end{aligned} \quad (7.24)$$

Hence, up to another constant κ_m ,

$$d(f_{\sigma_m \star \mathbf{p}}, s^0(\mathbf{p})) \leq \frac{\alpha_1}{n} \kappa_m + \frac{\kappa_\alpha}{n^2}$$

Lemma 7.4.6. *Let $m \geq 2$ and $p \in \mathcal{P}_{\sharp, n}^{m, q}(\alpha)$ and let $f_{\sigma_m \star \mathbf{p}}$ be defined as in Proposition 7.3.1, then*

$$d(f_{\sigma_m \star \mathbf{p}}, s^1(\mathbf{p})) \leq \frac{\kappa_\alpha}{n^2}$$

The distance $d(f_{\sigma_m \star \mathbf{p}}, s^1(\mathbf{p}))$ is given by :

$$\begin{aligned} d(f_{\sigma_m \star \mathbf{p}}, s^1(\mathbf{p})) &= \int_0^1 \|f(t) - s^1(\mathbf{p})(t)\|_1 dt \\ &= \sum_{i=0}^{n-1} \int_{\frac{i}{n}}^{\frac{i+1}{n}} \left\| g_i \left(nt - \frac{i}{n} \right) - s^1(\mathbf{p}) \right\|_1 dt, \end{aligned}$$

hereafter we divide the right hand side into three parts, β, γ, δ :

$$\begin{aligned} \|g_i(t) - s^1(\mathbf{p})(t)\| &\leq \|((C^m \star \sigma_m - \mathbf{1}) \star \mathbf{p})_i\| : \beta \\ &+ \left\| \left(C^{m-1} \star \sigma_m \star \Delta \star \mathbf{p} \right)_i t - (\mathbf{p}_{i+1} - \mathbf{p}_i)t \right\| : \gamma \\ &+ \left\| \sum_{k=2}^m \left(C^{m-k} \star \Delta^{\star k} \star \sigma_m \star \mathbf{p} \right)_i \frac{t^k}{k!} \right\| : \delta \end{aligned}$$

The β term is treated in a similar fashion to the previous section. Using Lemma 7.4.3, there exists a constant κ_m that depends only on m and a kernel A with $\|A\|_{\ell^1} \leq \frac{\kappa_m}{n}$ such that $A = \Delta^{-2}(C^m \star \sigma_m - \mathbf{1})$. Hence $\beta \leq \|(A \star \Delta^2 \star \mathbf{p})_i\|$. In order to deal with the δ term, bound t by 1, introduce the operator

$$Q = \sum_{k=2}^m C^{m-k} \star \Delta^{\star(k-2)} \star \sigma_m,$$

then $\delta \leq \|(Q \star \Delta^2 \star \mathbf{p})_i\|$. It is easy to check that there exists a constant κ_m that depends only on m such that $\|Q\|_{\ell^1} \leq \frac{\kappa_m}{n}$. It remains to deal with the γ term. For that purpose notice that

$$\begin{aligned} \left(C^{m-1} \star \sigma_m \star \Delta \star \mathbf{p} \right)_i - (\mathbf{p}_{i+1} - \mathbf{p}_i) &= \left(C^{m-1} \star \sigma_m \star \Delta \star \mathbf{p} - \sigma_{-3} \star \Delta \star \mathbf{p} \right)_i \\ &= \left((C^{m-1} \star \sigma_m - \sigma_{-3}) \star \Delta \star \mathbf{p} \right)_i. \end{aligned}$$

The operator $C^{m-1} \star \sigma_m - \sigma_{-3}$ sums up to zero and has support in $[-m, m]$ so that it is a first order derivative kernel in the sense of Lemma 7.4.2 and there exists a constant κ_m that depends only on m and a kernel $R = \Delta^{-1}(C^{m-1} \star \sigma_m - \sigma_{-3})$ with $\|R\|_{\ell^1} \leq \frac{\kappa_m}{n}$, so that $\gamma \leq \|(R \star \Delta^2 \star \mathbf{p})_i\|$.

Collecting all the terms we have,

$$\|g_i(t) - s^1(\mathbf{p})(t)\| \leq \sum_{j=1}^3 \|(A^j \star \Delta^2 \mathbf{p})_i\| \quad \text{with } \|A^j\|_{\ell^1} \leq \frac{\kappa_m}{n},$$

and finally

$$d(f_{\sigma_m \star \mathbf{p}}, s^1(\mathbf{p})) \leq \frac{1}{n} \sum_{i=0}^{n-1} \int_0^1 \|g_i(t) - s^1(\mathbf{p})(t)\| dt \leq \kappa_m \frac{\alpha_2}{n^2}.$$

Lemma 7.4.7. *Let $m \geq 1$ and $p \in \mathcal{P}_n^{m,q}(\alpha)$, let $\tilde{f}_{\sigma_m \star \mathbf{p}}$ be defined as in Lemma 7.4.4, then*

$$d(\tilde{f}_{\sigma_m \star \mathbf{p}}, s^0(\mathbf{p})) \leq \frac{\kappa \alpha}{n}$$

Lemma 7.4.8. *Let $m \geq 2$ and $p \in \mathcal{P}_n^{m,q}(\alpha)$ and let $\tilde{f}_{\sigma_m \star \mathbf{p}}$ be defined as in Lemma 7.4.4, then*

$$d(\tilde{f}_{\sigma_m \star \mathbf{p}}, s^1(\mathbf{p})) \leq \frac{\kappa \alpha}{n}$$

Lemmas 7.4.7 and 7.4.8. Since for any \mathbf{p} , $d(s_{\mathbf{p}}^0, s_{\mathbf{p}}^1) \leq \frac{\alpha_1}{n}$, it suffices to prove Lemma 7.4.7. We have

$$\begin{aligned} d(\tilde{f}_{\sigma_m \star \mathbf{p}}, s_{\mathbf{p}}^0) &= \int_0^1 \|\tilde{f}_{\sigma_m \star \mathbf{p}}(t) - s_{\mathbf{p}}^0(t)\| dt \\ &\leq \underbrace{\int_0^1 \|\tilde{f}_{\sigma_m \star \mathbf{p}}(t) - s_{\mathbf{p}}^0(\theta t + \tau)\| dt}_{\alpha} + \underbrace{\int_0^1 \|s_{\mathbf{p}}^0(\theta t + \tau) - s_{\mathbf{p}}^0(t)\| dt}_{\beta} \end{aligned}$$

The α term is a subpart of the equation (7.20) and can be bounded in a similar fashion to (7.22) :

$$\begin{aligned} \alpha &= \int_0^1 \|f_{\sigma_m \star \mathbf{p}}(\theta t + \tau) - s_{\mathbf{p}}^0(\theta t + \tau)\| dt = \theta^{-1} \sum_{i=10m}^{n-10m} \int_{\frac{i}{n}}^{\frac{i+1}{n}} \|f_{\sigma_m \star \mathbf{p}}(t) - s_{\mathbf{p}}^0(t)\| dt \\ &\leq \theta^{-1} \left(\underbrace{\frac{1}{n} \sum_{i=10m}^{n-10m} \|(C^m \star \sigma_m \star \mathbf{p})_i - \mathbf{p}_i\|}_{\zeta} \right. \\ &\quad \left. + \underbrace{\frac{1}{n} \sum_{i=10m}^{n-10m} \sum_{k=1}^m \|(C^{m-k} \star \Delta^{\star k} \star \sigma_m \star \mathbf{p})_i \frac{1}{(k+1)!}\|}_{\eta} \right) \end{aligned}$$

Given that the support of $C^{m-k} \star \sigma_m$ is included in $[-m, m]$ and using Lemma 7.4.1 the η term can be bounded in the same manner as in Equation (7.23). For the ζ part define $A = \Delta^{-1}(C^m \star \sigma_m - \mathbf{1})$; the support of A is included in $[-m, m]$ by virtue of Lemma 7.4.1 and bounding θ^{-1} by $1 + \frac{\kappa}{n}$, we have that :

$$\alpha \leq \frac{\kappa \alpha}{n}$$

For the β part notice that $\theta + 2\tau = 1$, so that :

$$|\theta t + \tau - t| \leq \tau$$

$$\begin{aligned} \beta &= \sum_{i=0}^{n-1} \int_{\frac{i}{n}}^{\frac{i+1}{n}} \|s_{\mathbf{p}}^0(\theta t + \tau) - \mathbf{p}_i\| dt \leq \sum_{i=0}^{n-1} \int_{\frac{i}{n}}^{\frac{i+1}{n}} \sum_{k=-n\tau+1}^{n\tau} \|\mathbf{p}_{i+k} - \mathbf{p}_{i+k-1}\| \chi_{i+k \in [1, n-1]} dt \\ &\leq \frac{1}{n} \sum_{i=1}^{n-1} 2n\tau \|(\Delta \star \mathbf{p})_i\| = 2n\tau \|\Delta \star \mathbf{p}\|_{\ell^1} \leq \kappa \frac{\alpha_1}{n} \end{aligned}$$

Since $\|\Delta \star \mathbf{p}\|_{\ell^1} \leq \frac{\alpha_1}{n}$ and $\tau \leq \frac{\kappa}{n}$. This allows us to conclude that :

$$d\left(\tilde{f}_{\sigma_m \star \mathbf{p}}, s_{\mathbf{p}}^0\right) \leq \frac{\alpha_1}{n} \kappa_m + \frac{\kappa \alpha}{n^2},$$

and

$$d\left(\tilde{f}_{\sigma_m \star \mathbf{p}}, s_{\mathbf{p}}^1\right) \leq \frac{\alpha_1}{n} \kappa_m + \frac{\kappa \alpha}{n^2}.$$

□

7.4.5 Proof of theorems

The end of the proof proceeds as follows. For any $\mathbf{p} \in \mathcal{P}_n^{m,q}(\alpha)$, build $\mathbf{q} = \sigma_m \star \mathbf{p}$, notice that $\|\sigma_m\|_{\ell^1} = n^{-1}$ so that $\mathbf{q} \in \mathcal{P}_{\sharp, n}^{m,q}(\alpha)$. We have that $f_{\mathbf{q}} \in W_{\sharp}^{m,q}(\alpha + \frac{\kappa \alpha}{n^2})$. Let $\delta \in \mathbb{R}$ be a scaling factor such that $\delta f_{\mathbf{q}} \in W_{\sharp}^{m,q}(\alpha)$. Notice that there exists yet another constant depending on α only and still denoted κ_{α} such that $\delta = 1 + \frac{\kappa \alpha}{n^2}$, we then have

$$d(\delta f_{\mathbf{q}}, f_{\mathbf{q}}) \leq \frac{\kappa \alpha}{n^2}$$

By the triangle inequality for the distance d , we conclude that

$$d(s^0(\mathbf{p}), \delta f_{\mathbf{q}}) \leq \frac{\kappa}{n}, \quad \text{and } d(s^1(\mathbf{p}), \delta f_{\mathbf{q}}) \leq \frac{\kappa}{n^2} \text{ if } m \geq 2. \quad (7.25)$$

Thence $\delta f_{\mathbf{q}} \in W_{\#}^{m,q}(\boldsymbol{\alpha})$ is sufficiently close to $s^0(\mathbf{p})$ (resp. $s^1(\mathbf{p})$). This ends the proof.

Theorems 7.3.1 to 7.3.3. For any function f in $W_{\#}^{m,q}(\boldsymbol{\alpha})$ (resp. f in $W^{m,q}(\boldsymbol{\alpha})$) take $\mathbf{p} \in \mathbb{R}^{d \times n}$ such that $\mathbf{p}_i = f\left(\frac{i}{n}\right)$, then $\mathbf{p} \in \mathcal{P}_{\#,n}^{m,q}(\boldsymbol{\alpha})$ (resp. $\mathbf{p} \in \mathcal{P}_n^{m,q}(\boldsymbol{\alpha})$) by virtue of Proposition 7.4.1. Still using the result of Proposition 7.4.1, the distance between f and its approximant, whether it is a piecewise constant or a piecewise linear spline, is bounded with the correct rate.

Now for any piecewise constant or linear function $s^0(\mathbf{p}) \in \mathcal{S}_{\#,n}^{m,q}(\boldsymbol{\alpha})$ or $s^1(\mathbf{p}) \in \mathcal{L}_{\#,n}^{m,q}(\boldsymbol{\alpha})$ (resp. $s^0(\mathbf{p}) \in \mathcal{S}_n^{m,q}(\boldsymbol{\alpha})$ or $s^1(\mathbf{p}) \in \mathcal{L}_n^{m,q}(\boldsymbol{\alpha})$) build $\mathbf{q} = \sigma_m \star \mathbf{p}$ and the smoothing spline $f_{\mathbf{q}}$ defined as in Proposition 7.3.1 (resp. $\tilde{f}_{\mathbf{q}}$ defined as in Lemma 7.4.4). This spline belongs to $W_{\#}^{m,q}\left(\left(1 + \frac{\kappa_{\boldsymbol{\alpha}}}{n^2}\right)\boldsymbol{\alpha}\right)$ by using Proposition 7.4.2 (resp. $W^{m,q}\left(\left(1 + \frac{\kappa_{\boldsymbol{\alpha}}}{n^2}\right)\boldsymbol{\alpha}\right)$ by using Lemma 7.4.4). The distance d between $f_{\mathbf{q}}$ (resp. $\tilde{f}_{\mathbf{q}}$) and the piecewise constant or linear spline is bounded and the result of Lemmas 7.4.5 or 7.4.6 (resp. Lemmas 7.4.7 or 7.4.8) with the correct rates. Introduce the scaled function $\delta f_{\mathbf{q}}$ (resp. $\delta \tilde{f}_{\mathbf{q}}$) as described in (7.25) to obtain a function in $W_{\#}^{m,q}(\boldsymbol{\alpha})$ (resp. $W^{m,q}(\boldsymbol{\alpha})$) whose distance with respect to the spline is bounded with the correct rate. \square

Conclusion

In this article, we bound the Hausdorff distance between set of continuous curve with a prescribed Sobolev semi-norm on their derivative and their discrete piecewise constant and piecewise linear counterparts. Bounding the Hausdorff requires a twofold control that is :

- given a continuous curve, discretize the curve with a piecewise constant or linear spline sufficiently close in the sense of the 1-Wasserstein distance and which belongs to the suitable spline set.
- given a piecewise constant or linear spline, construct a continuous function sufficiently close in the sense of the 1-Wasserstein distance and which belong to the correct Sobolev multiballs.

The discretization step is trivial and given by the uniform sampling of the continuous curve. On the over hand finding a m times continuous function that approximates the 0-th or 1-st order spline is trickier. The construction of this continuous approximant involves using B -splines of order m but it appears that its expression is elegant (see Proposition 7.3.3). The derivatives

continuousness of this approximant yields recurrence relationships involving Eulerian numbers and that are, to the best of our knowledge new.

7.5 Numerical implementation

Authors released an open source implementation of the presented smoothing Eulerian B-splines¹. The code implements for $d = 2$ the previously presented method with a graphical user interface. Note that this code is easily scalable to higher dimensions since its time complexity depends only on the number of points \mathbf{p} . Approximation results using this toolbox are depicted in Figure 7.1 and 7.5.

7.6 Outlook

In this chapter, the approximation spline defined in Equation 7.6 is tailored for periodic curves. When the curve is non-periodic, the trick involves shifting and scaling the time parametrization (see Lemma 7.4.4) to cope with the inherent periodicity of the discrete convolutional operator.

This definition allows the re-use of previously demonstrated inequalities in the periodic case. We believe that the approximation built with this technique yields non-optimal constants due to slack in the non-periodic inequalities. Note that other approximation tricks could be used, like phantom points, to improve the constants of Lemma 7.4.7 and 7.4.8.

1. <https://github.com/lebrat/eulerianApproximation>

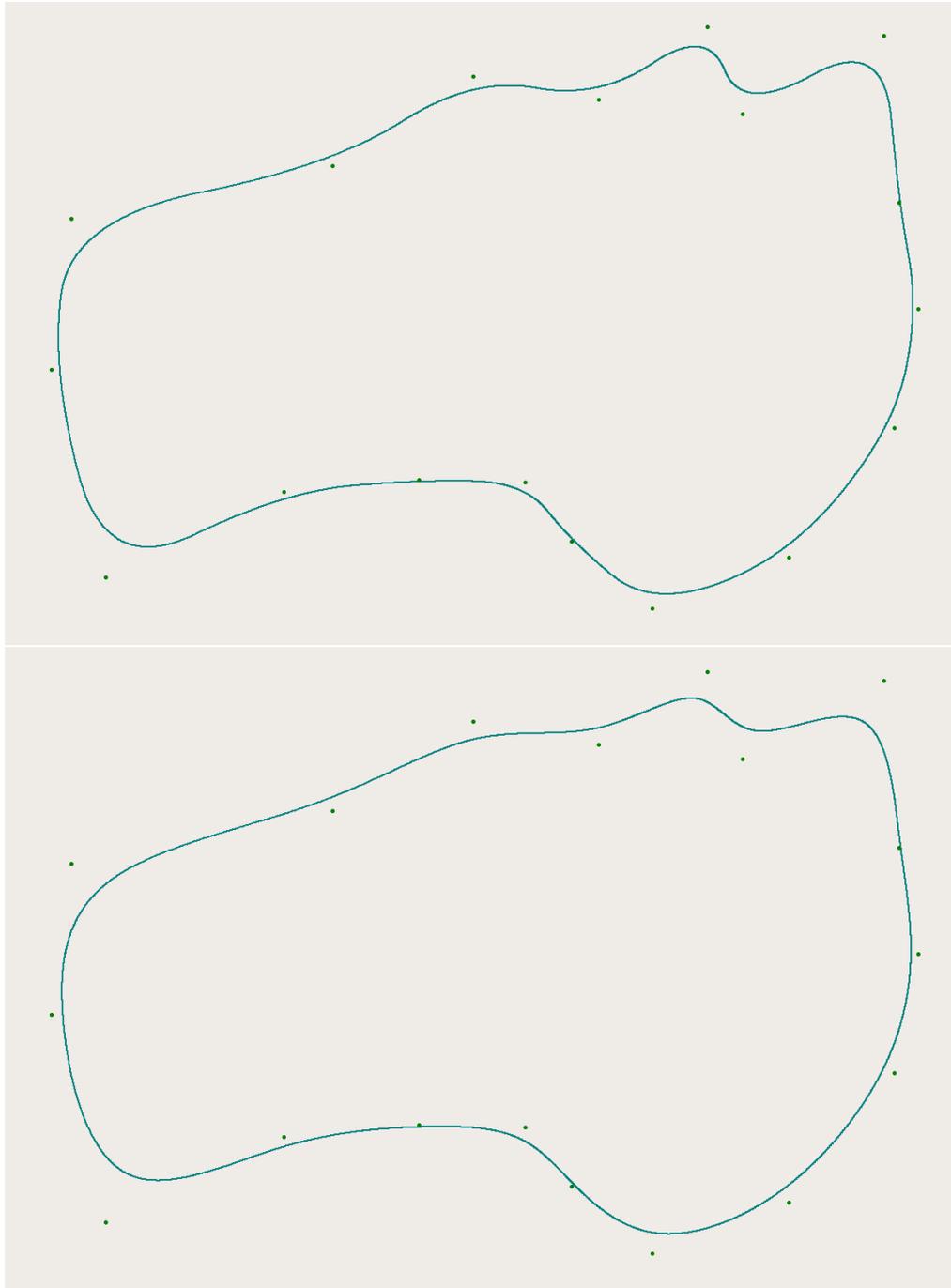


FIGURE 7.1 – Periodic approximation spline f (7.6) in blue for the controls points \mathbf{p} in green. The degree of the approximation spline is $m = 3$ (top) and $m = 4$ (bottom).



FIGURE 7.2 – Periodic approximation spline f (7.6) in blue for the controls points \mathbf{p} in green. The degree of the approximation spline is $m = 5$ (top) and $m = 8$ (bottom).

Chapter 8

Numerical issues

Contents

8.1	Performing the integration over Laguerre cells .	170
8.1.1	Green’s method of integration	171
8.1.2	Detail of the computations	172
8.1.3	Complexity analysis	175
8.2	Standard tools of optimization and computational linear algebra	177
8.2.1	Basin of attraction of the Newton method	177
8.2.2	Damped Newton method	179
8.2.3	Condition number and its effect in solving a linear system	180
8.3	Optimal transport optimization for curvling . .	182
8.3.1	Geometric properties of Laguerre tessalations in the curvling setting	182
8.3.2	Condition number of the Hessian in OT	183
8.3.3	Size of the basin of attraction in the semi-discrete OT	187
8.3.4	Non definite Hessian in OT	189
8.3.5	Regularized Newton method for Curvling	192
8.3.6	Multiscale algorithm for Curvling	195

8.1 Performing the integration over Laguerre cells

In this section we discuss the integration procedure required to compute the Monge Kantorovitch functional (1.11) and its derivatives in the semi-discrete case, see Chapter 5 for details. First one can notice that the integration can be split into two subproblems : integration over the whole Laguerre cell and integration over the boundaries of the Laguerre cell. Generally speaking, integrating over the boundary is less complex. Indeed, the computation of the Laguerre Tessellation in dimension two by the `CGAL` library [128] yields directly the description of the Laguerre cells in term of their vertices and edges (the later can be either segments/rays/lines). Integrating over the interior of the Laguerre cell remains trickier and is computationally more expensive.

First, let us describe the usual methods used to compute these integrals. Authors of [34] in the `ibnot` toolbox compute the intersection between the raw pixels describing the μ measure and each underlying Laguerre cell. Authors of [94] use a similar technique but enhance the computation of second order moments by triangulating the background measure and summing for each triangle the value of the integral obtained in closed-form. Bruno Levy [85], in his toolbox `geogram` (which includes a library for the computation of the Laguerre tessellation) supposes that the image density is given on a triangular mesh and efficiently computes its intersection with the Laguerre tessellation.

As noted in [89, 138] the volume integration required for the computation of the cost function and its gradient can be recast to contour integration using a standard divergence theorem. To recap, for each Laguerre cell one needs to compute the following integrals :

$$I_1 = \int_{\mathcal{L}_i(\phi, \mathbf{x})} (\|\mathbf{x}_i - x\|^2 - \phi_i) d\mu(x), \quad \text{and} \quad I_2 = \int_{\mathcal{L}_i(\phi, \mathbf{x})} d\mu(x),$$

for respectively the cost function (5.14) and its gradient (5.15). To solve this problem, it is sufficient to compute the moment of the measure μ on each \mathcal{L}_i up to order 2. Indeed one can trivially expand the dot product to get :

$$I_1 = \int_{\mathcal{L}_i(\phi, \mathbf{x})} \|x\|^2 d\mu(x) - 2\mathbf{x}_i \cdot \int_{\mathcal{L}_i(\phi, \mathbf{x})} x d\mu(x) + (\|\mathbf{x}_i\|^2 - \phi_i) \int_{\mathcal{L}_i(\phi, \mathbf{x})} d\mu(x)$$

8.1.1 Green's method of integration

Let M be a $H^1(\mathcal{L}_i)$ vector field then Green's formula states [23] :

$$\int_{\mathcal{L}_i} \nabla \cdot M dx = \int_{\partial\mathcal{L}_i} M \cdot \mathbf{n}_i dx,$$

where \mathbf{n}_i is the outward pointing normal vector of the boundary $\partial\mathcal{L}_i$. This theorem is key since it helps us to curtail the integration of the Laguerre cell down to line integrals along the sequence of oriented edges. Note that computing these line integrals does not introduce extra computations. Indeed the integration procedure already goes over these edges for Hessian computation (1.12).

The final ingredient to describe this new integration algorithm is to rewrite the k -th order moment of the μ -measure as the divergence of a H^1 vector field. In the following paragraph we describe the method for the special case of dimension 2.

The measure μ has a density defined as a bilinear interpolation ($Q1$ Lagrangian finite elements) of the input image \mathbf{u} of size $(n_1 \times n_2)$. The support of μ is the box $[0, 1] \times [0, 1]$. Define the discretization step-size in the i -th direction as $\Delta_{x_i} = \frac{1}{n_i - 1}$. For any (x_1, x_2) denote $\mathbf{i} = \lfloor \frac{x_1}{\Delta_{x_1}} \rfloor$ and $\mathbf{j} = \lfloor \frac{x_2}{\Delta_{x_2}} \rfloor$. The expression of the considered density μ is :

$$\begin{aligned} \mu(x_1, x_2) &= \mathbf{u}[\mathbf{i}, \mathbf{j}] + \theta \Delta \mathbf{u}_{x_1}[\mathbf{i}, \mathbf{j}] + \beta \Delta \mathbf{u}_{x_2}[\mathbf{i}, \mathbf{j}] + \theta\beta \Delta \mathbf{u}_{x_1 x_2}[\mathbf{i}, \mathbf{j}], \quad (8.1) \\ \text{with } \theta &= \left\{ \frac{x_1}{\Delta_{x_1}} \right\} \quad \text{and} \quad \beta = \left\{ \frac{x_2}{\Delta_{x_2}} \right\}, \end{aligned}$$

where $\{a\} = a - \lfloor a \rfloor$ denotes the decimal part of a , with

$$\left| \begin{array}{l} \Delta \mathbf{u}_{x_1}[\mathbf{i}, \mathbf{j}] = \mathbf{u}[\mathbf{i} + 1, \mathbf{j}] - \mathbf{u}[\mathbf{i}, \mathbf{j}] \\ \Delta \mathbf{u}_{x_2}[\mathbf{i}, \mathbf{j}] = \mathbf{u}[\mathbf{i}, \mathbf{j} + 1] - \mathbf{u}[\mathbf{i}, \mathbf{j}] \\ \Delta \mathbf{u}_{x_1 x_2}[\mathbf{i}, \mathbf{j}] = \mathbf{u}[\mathbf{i}, \mathbf{j} + 1] + \mathbf{u}[\mathbf{i} + 1, \mathbf{j} + 1] \\ \quad \quad \quad - \mathbf{u}[\mathbf{i} + 1, \mathbf{j}] - \mathbf{u}[\mathbf{i}, \mathbf{j} + 1] \end{array} \right. \quad (8.2)$$

For each $\ell \in \llbracket 0, 2 \rrbracket$ and for each $k \in \llbracket 0, d \rrbracket$ we denote M_k^ℓ a well chosen vector field such that

$$\nabla \cdot M_k^\ell = (x \cdot e_k)^\ell \mu.$$

where e_k is k -th vector of the canonical basis.

In order to compute exact integration, several choices for the vector fields M_k^ℓ are possible. The simplest one is to suppose that M_k^ℓ derives from a gradient and to solve a Laplace equation. Note however that the right hand side can be a $Q3$ finite element so that M has to be a $Q5$ finite element. In order to limit space complexity of the integration procedure, we rather use integration on a well chosen direction. Hence our choice of vector fields M_k^ℓ are always directed along a principal axis of the image, the whole procedure is detailed below.

8.1.2 Detail of the computations

This section details meticulously calculations that yield the vector fields required to convert integrals over Laguerre cells into integrals over the boundaries of the Laguerre cells. These details allow a more digestible reading of our code.

For the 0-th order raw moment The vector field M_k^0 does not depend on k and is chosen colinear to e_1 ,

$$M_k^0 = \tilde{M}^0 e_1,$$

with,

$$\tilde{M}^0 = \Delta_{x_1} \left(\theta u + \frac{\theta^2}{2} \Delta u_{x_1} + \theta \beta \Delta u_{x_2} + \frac{\theta^2}{2} \beta \Delta u_{x_1 x_2} + r_0 + \beta r_1 \right)$$

where r_0 and r_1 are functions that are constant over a pixel and ensure the continuity of the vector field M^0 , their closed form is given by :

$$\begin{aligned} r_0 [i, j] &= \sum_{k=0}^{i-1} u[k, j] + \frac{1}{2} (u[i, j] - u[0, j]) \\ r_1 [i, j] &= \sum_{k=0}^{i-1} (u[k, j+1] - u[k, j]) + \frac{1}{2} (u[i+1, j+1] - u[i+1, j] \\ &\quad - u[0, j+1] + u[0, j]) . \end{aligned}$$

8.1. PERFORMING THE INTEGRATION OVER LAGUERRE CELLS 173

For the 1-st order raw moment In the following paragraph we wish to integrate $x_1 d\mu$ and $x_2 d\mu$. With respect to the parameter x_1 , we define the vector field M_1^1 as :

$$M_1^1 = \tilde{M}_1^1 e_1$$

with,

$$\begin{aligned} \tilde{M}_1^1 = \Delta_{x_1}^2 \left[i \left(\theta u + \frac{\theta^2}{2} \Delta u_{x_1} + \beta \theta \Delta u_{x_2} + \beta \frac{\theta^2}{2} \Delta u_{x_1 x_2} \right) \right. \\ \left. + \frac{\theta^2}{2} u + \frac{\theta^3}{3} \Delta u_{x_1} + \beta \frac{\theta^2}{2} \Delta u_{x_2} + \beta \frac{\theta^3}{3} \Delta u_{x_1 x_2} + r_0 + \beta r_1 \right], \end{aligned}$$

where r_0 and r_1 are given by :

$$\begin{aligned} r_0 [i, j] &= \sum_{k=0}^{i-1} k \left(u[k, j] + \frac{1}{2} \Delta u_{x_1}[k, j] \right) + \frac{1}{2} u[k, j] + \frac{1}{3} \Delta u_{x_1}[k, j] \\ r_1 [i, j] &= \sum_{k=0}^{i-1} k \left(\Delta u_{x_2}[k, j] + \frac{1}{2} \Delta u_{x_1 x_2}[k, j] \right) + \frac{1}{2} \Delta u_{x_2}[k, j] + \frac{1}{3} \Delta u_{x_1 x_2}[k, j]. \end{aligned}$$

Similarly the vector field $M_2^1 = \tilde{M}_2^1 e_2$ is prescribed to be align with the second direction with :

$$\begin{aligned} \tilde{M}_2^1 = \Delta_{x_2}^2 \left[j \left(\beta u + \theta \beta \Delta u_{x_1} + \frac{\beta^2}{2} \Delta u_{x_2} + \theta \frac{\beta^2}{2} \Delta u_{x_1 x_2} \right) \right. \\ \left. + \frac{\beta^2}{2} u + \theta \frac{\beta^2}{2} \Delta u_{x_1} + \frac{\beta^3}{3} \Delta u_{x_2} + \theta \frac{\beta^3}{3} \Delta u_{x_1 x_2} + r_0 + \theta r_1 \right]. \end{aligned}$$

Once again r_0 and r_1 have to be chosen to ensure continuity the of \tilde{M}_2^1 :

$$\begin{aligned} r_0 [i, j] &= \sum_{k=0}^{j-1} k \left(u[i, k] + \frac{1}{2} \Delta u_{x_2}[i, k] \right) + \frac{1}{2} u[i, k] + \frac{1}{3} \Delta u_{x_2}[i, k] \\ r_1 [i, j] &= \sum_{k=0}^{j-1} k \left(\Delta u_{x_1}[i, k] + \frac{1}{2} \Delta u_{x_1 x_2}[i, k] \right) + \frac{1}{2} \Delta u_{x_1}[i, k] + \frac{1}{3} \Delta u_{x_1 x_2}[i, k] \end{aligned}$$

For the 2-nd order raw moment Finally we describe how to integrate the second order term. First in the x_1 direction, we define the vector field M_1^2 as :

$$M_1^2 = \tilde{M}_1^2 e_1$$

with,

$$\begin{aligned} \tilde{M}_1^2 = \Delta_{x_1}^3 & \left[i^2 \left(\theta \mathbf{u} + \frac{\theta^2}{2} \Delta \mathbf{u}_{x_1} + \theta \beta \Delta \mathbf{u}_{x_2} + \beta \frac{\theta^2}{2} \Delta \mathbf{u}_{x_1 x_2} \right) \right. \\ & + 2i \left(\frac{\theta^2}{2} \mathbf{u} + \frac{\theta^3}{3} \Delta \mathbf{u}_{x_1} + \frac{\theta^2}{2} \beta \Delta \mathbf{u}_{x_2} + \frac{\theta^3}{3} \beta \Delta \mathbf{u}_{x_1 x_2} \right) \\ & \left. + \frac{\theta^3}{3} \mathbf{u} + \frac{\theta^4}{4} \Delta \mathbf{u}_{x_1} + \frac{\theta^3}{3} \beta \Delta \mathbf{u}_{x_2} + \frac{\theta^4}{4} \beta \Delta \mathbf{u}_{x_1 x_2} + \mathbf{r}_0 + \beta \mathbf{r}_1 \right], \end{aligned}$$

where \mathbf{r}_0 and \mathbf{r}_1 are given by :

$$\begin{aligned} \mathbf{r}_0 [i, j] &= \sum_{k=0}^{i-1} k^2 \left(\mathbf{u}[k, j] + \frac{1}{2} \Delta \mathbf{u}_{x_1}[k, j] \right) + 2k \left(\frac{1}{2} \mathbf{u}[k, j] + \frac{1}{3} \Delta \mathbf{u}_{x_1}[k, j] \right) \\ &+ \frac{1}{3} \mathbf{u}[k, j] + \frac{1}{4} \Delta \mathbf{u}_{x_1}[k, j] \\ \mathbf{r}_1 [i, j] &= \sum_{k=0}^{i-1} k^2 \left(\Delta \mathbf{u}_{x_2}[k, j] + \frac{1}{2} \Delta \mathbf{u}_{x_1 x_2}[k, j] \right) + 2k \left(\frac{1}{2} \Delta \mathbf{u}_{x_2}[k, j] + \frac{1}{3} \Delta \mathbf{u}_{x_1 x_2}[k, j] \right) \\ &+ \frac{1}{3} \Delta \mathbf{u}_{x_2}[k, j] + \frac{1}{4} \Delta \mathbf{u}_{x_1 x_2}[k, j]. \end{aligned}$$

In a similar fashion we define $M_2^2 = \tilde{M}_2^2 e_2$

$$\begin{aligned} \tilde{M}_2^2 = \Delta_{x_2}^3 & \left[j^2 \left(\beta \mathbf{u} + \theta \beta \Delta \mathbf{u}_{x_1} + \frac{\beta^2}{2} \Delta \mathbf{u}_{x_2} + \theta \frac{\beta^2}{2} \Delta \mathbf{u}_{x_1 x_2} \right) \right. \\ & + 2j^2 \left(\frac{\beta^2}{2} \mathbf{u} + \theta \frac{\beta^2}{2} \Delta \mathbf{u}_{x_1} + \frac{\beta^3}{3} \Delta \mathbf{u}_{x_2} + \theta \frac{\beta^3}{3} \Delta \mathbf{u}_{x_1 x_2} \right) \\ & \left. + \frac{\beta^3}{3} \mathbf{u} + \theta \frac{\beta^3}{3} \Delta \mathbf{u}_{x_1} + \frac{\beta^4}{4} \Delta \mathbf{u}_{x_2} + \theta \frac{\beta^4}{4} \Delta \mathbf{u}_{x_1 x_2} + \mathbf{r}_0 + \theta \mathbf{r}_1 \right], \end{aligned}$$

where \mathbf{r}_0 and \mathbf{r}_1 are given by :

$$\begin{aligned} \mathbf{r}_0 [i, j] &= \sum_{k=0}^{j-1} k^2 \left(\mathbf{u}[i, k] + \frac{1}{2} \Delta \mathbf{u}_{x_2}[i, k] \right) + 2k \left(\frac{1}{2} \mathbf{u}[i, k] + \frac{1}{3} \Delta \mathbf{u}_{x_2}[i, k] \right) \\ &+ \frac{1}{3} \mathbf{u}[i, k] + \frac{1}{4} \Delta \mathbf{u}_{x_2}[i, k] \end{aligned}$$

$$r_1[i, j] = \sum_{k=0}^{j-1} k^2 \left(\Delta u_{x_1}[i, k] + \frac{1}{2} \Delta u_{x_1 x_2}[i, k] \right) + 2k \left(\frac{1}{2} \Delta u_{x_1}[i, k] + \frac{1}{3} \Delta u_{x_1 x_2}[i, k] \right) + \frac{1}{3} \Delta u_{x_1}[i, k] + \frac{1}{4} \Delta u_{x_1 x_2}[i, k]$$

With these vector fields the contour integrals can be computed following the two steps :

- Intersect the pixel grid with the edges defining the boundary of the Laguerre cell. Note that these intersections have a closed form and are efficiently computed.
- Perform exact computation of the integral using Gaussian quadrature rule.

8.1.3 Complexity analysis

We can now compare the complexity of volume integration with the complexity of contour integration. The principal crux of volume integration stems from assigning the pixels to each Laguerre cell. Thence both the time and space complexity are linear with respect to the number of pixels. According to the previous notation the complexity is of order $\mathcal{O}(n_1 n_2)$.

In contrast contour integration requires extra space complexity. Indeed, one has to store 14 auxiliary matrices of the size of the input image. Note that this computation is done beforehand since these matrices are tessellation invariant, and this computation is done once and for all when a image is given (throughout the computation of optimal transport **and** the optimization of Dirac masses and Dirac position).

Nevertheless time complexity is reduced, since contour integrations already appear in the Hessian computation. One can bound time complexity of contour integration by

$$\mathcal{O} \left(2 \max(n_1, n_2) \# \mathbf{x} \max_i \deg(\mathcal{L}_i) \right),$$

with $\# \mathbf{x}$ the number of points composing the discrete measure and $\deg(\mathcal{L}_i)$ the number of neighbors of the i th Laguerre cell.

In Figure 8.1 we compare the time required to perform integrations for these methods. For the pixel integration we use the `iBnot` toolbox¹. The results are consistent with the complexity analysis.

1. <http://www.geometry.caltech.edu/BlueNoise/data/bnot-src.zip>

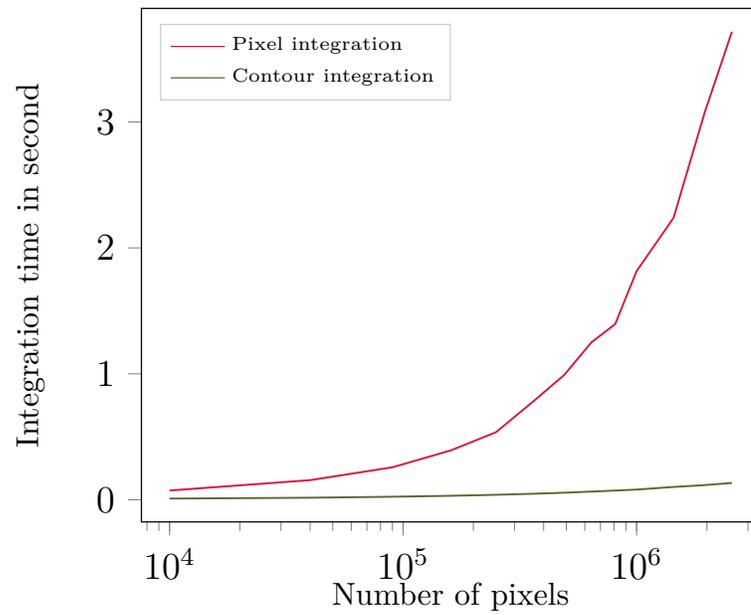


FIGURE 8.1 – Time to compute the gradient of the Monge Kantorovitch functional for an increasing number of pixels and a fixed number of Dirac masses in the discrete measure.

8.2 Standard tools of optimization and computational linear algebra

The aim of the following subsections 8.2.1, 8.2.2, 8.2.3 is to recall some properties in optimization and computational linear algebra. These in-depth knowledges are key to understanding the subtleties of the damped Newton algorithm used to optimized the Kantorovitch functional.

First, we describe iterative methods used to solve unconstrained convex minimization problem. Standard algorithm are performed by repeating the two following steps at iteration k :

1. Find a descent direction $d^{(k)}$.
2. Select a timestep $\tau^{(k)}$ and update the variable x : $x^{(k+1)} = x^{(k)} + \tau^{(k)}d^{(k)}$.

The algorithm is stopped when some convergence criterion are met. When the derivatives of the functional are known up to the second order one can distinguish two major types of methods :

- First order methods, that make only use of the gradient information to determine a descent direction $d^{(k)}$.
- Second order methods or Newton type methods, where the second order information (or its approximation) is used to find a descent direction. Typically this direction reads as $d^{(k)} = -(B^{(k)}(x^{(k)}))^{-1}\nabla f(x^{(k)})$. When $B^{(k)}$ is the Hessian of f the method is called **Newton method**. Conversely, when $B^{(k)}$ is an approximation of the Hessian of f the resulting algorithm is termed **quasi-Newton method**.

In the following section we will describe more precisely second order methods, which are more appropriate to solve the semi-discrete L^2 optimal transport problem [34, 94, 138, 85, 80, 96].

8.2.1 Basin of attraction of the Newton method

The principal asset of the Newton type method is its quadratic convergence in a vicinity of minimizer. This neighborhood, where the convergence rate of the algorithm is boosted is coined as the basin of attraction. First let us recap several properties of second order methods, before describing in more details the basin of attraction of Newton type methods.

Define \hat{f} the approximation of f at point $x^{(k)}$ by :

$$\hat{f}(x^{(k)} + h) = f(x^{(k)}) + \nabla f(x^{(k)})^T h + \frac{1}{2} h^T B^{(k)}(x^{(k)}) h, \quad (8.3)$$

when $B^{(k)} = H[f](x^{(k)})$, then the approximation \hat{f} is the Taylor approximation of f . If $B^{(k)}$ is a semi-definite positive matrix, then the quadratic form \hat{f} is convex in h . The optimality condition for the approximate \hat{f} yields :

$$h_\star = -(B^{(k)}(x^{(k)}))^{-1} \nabla f(x^{(k)}) := d^{(k)}. \quad (8.4)$$

The direction $d^{(k)}$ is called the (quasi)-Newton direction. If $B^{(k)}$ is s.d.p., one can easily check that $d^{(k)}$ is a descent direction, that is :

$$\nabla f(x)^T d^{(k)} = -\nabla f(x)^T (B^{(k)})^{-1} \nabla f(x) < 0 \quad \text{if} \quad \nabla f(x) \neq 0.$$

The Newton-type iterate minimizes the local second order approximation \hat{f} of f , it is given by :

$$x^{(k+1)} = x^{(k)} - \tau^{(k)} (B^{(k)}(x^{(k)}))^{-1} \nabla f(x^{(k)}), \quad (8.5)$$

with $B^{(k)}$ the Hessian or its approximation, and $\tau^{(k)}$ the stepsize. When $\tau^{(k)}$ is set to 1, and $B^{(k)}$ is the Hessian, then (8.5) is called pure Newton's step. If the function f is quadratic, and strictly convex the pure Newton's step gives the exact minimizer in one iteration. If the function is close to quadratic, Newton's iteration with the natural stepsize $\tau^{(k)} = 1$ should be a good estimate.

However, the pure Newton method has to be used with caution, indeed this method can diverge even for strictly convex functions. We recall Nesterov's famous counter example [102], let f be strictly convex and given by :

$$f(x) = \sqrt{1+x^2} \quad \text{with} \quad f'(x) = \frac{x}{\sqrt{1+x^2}} \quad \text{and} \quad f''(x) = \frac{1}{(1+x^2)^{3/2}}.$$

In this case the pure Newton iterate (scheme (8.5) with $\tau^{(k)} = 1$) reads as :

$$x^{(k+1)} = -\left(x^{(k)}\right)^3.$$

So that the pure Newton method diverges if $x_0 > 1$ and oscillates if $x_0 = \pm 1$. The insight to take away from this counter-example is : if we are not inside

the basin of attraction of a minimizer, then the pure Newton's method can diverge.

The Newton algorithm can be interpreted as a steepest descent method for the metric $H[f](x)$, so that, if one stay sufficiently close to a minimum x_* then the metric chosen $H[f](x) \approx H[f](x_*)$ is neat. Intuition suggests that the rate of change of this metric is also crucial in the performance of the Newton method. Assume that the Hessian of f is Lipschitz continuous with constant M , that is :

$$\|H[f](x) - H[f](y)\|_2 \leq M \|x - y\|_2,$$

then M is a bound of the third order derivative of f . The quantity M depicts the rate of variation of the metric for the steepest descent method, or to put it in another way the speed of variation of the quadratic model (8.3). We recall the following theorem from Nesterov [102].

Theorem 8.2.1. *Suppose that :*

1. *The function f is $\mathcal{C}^2(\mathbb{R}^2)$ with M -Lipschitz continuous Hessian.*
2. *There exists a local minimum x_* with a positive definite Hessian :*

$$H[f](x_*) \succeq l_* I_n, \quad l_* > 0.$$

3. *The initial starting point x_0 is close enough to x_* :*

$$\|x_0 - x_*\| < \frac{2l_*}{3M} \tag{8.6}$$

Then $\|x^{(k)} - x_\| < \frac{2l_*}{3M}$ for all k and the pure Newton method converges quadratically :*

$$\|x^{(k+1)} - x_*\| \leq \frac{M\|x^{(k)} - x_*\|^2}{2(l - M\|x^{(k)} - x_*\|)} \tag{8.7}$$

This theorem tells us that the Newton method will converge if the initial guess is close enough to the minimizer. The vicinity of size $\frac{2l_*}{3M}$ around the minimizer x_* (8.6) is called **basin of attraction** of the Newton method.

8.2.2 Damped Newton method

Outside the basin of attraction the Newton method with stepsize 1 may diverge. In order to have an algorithm that converges globally one has to

enforce the Armijo-Wolfe rule on the Newton iterate (8.5) by finding $\tau^{(k)} < 1$ that ensures a sufficient decrease of the cost function. Such an algorithm is called **damped Newton method**. The convergence analysis (see [102] for details) is then split in two phases, first the *damped Newton phase* and then when we are sufficiently close to a basin of attraction the *Newton phase* ($\tau^{(k)} = 1$) with quadratic convergence.

Let $L_{x_0}^-(f)$ denote the sublevel set of f for the value $f(x^{(0)})$:

$$L_{x^{(0)}}^-(f) = \{x \in \mathbb{R}^n, f(x) \leq f(x^{(0)})\}$$

Suppose that f is strongly convex and that its Hessian is bounded on $L_{x^{(0)}}^-(f)$:

$$\forall x \in L_{x^{(0)}}^-(f), \quad H[f](x) \succeq lI_n \quad \text{and} \quad H[f](x) \preceq LI_n \quad \text{with} \quad 0 < l \leq L$$

There exist upper bounds [20] on the number of *damped Newton* iterations n_{dn} in order to attain the basin of quadratic convergence given by :

$$n_{\text{dn}} \leq \frac{f(x_0) - f(x_*)}{\gamma} \quad \text{with} \quad \gamma = \frac{l^5}{L^2 M^2}. \quad (8.8)$$

Note that in Equation (8.8) the upper bound on the number of iterations depends on the ratio $\frac{l}{L}$. This quantity differs from the inverse of the condition number κ^{-1} since we are looking within $L_{x_0}^-(f)$ to the smallest and the largest eigenvalues of the Hessian.

This bound is very pessimistic and unusable in practice to have a descent estimate of the time spent in the *damped Newton phase*. Nevertheless, two quantities are of interest in this formula the Lipschitz constant M of the Hessian matrix and the ratio of the smallest and largest eigenvalues.

8.2.3 Condition number and its effect in solving a linear system

Let us recap some properties of computational linear algebra associated with the condition number of a matrix. The condition number measures how sensitive the linear function is to errors or similarly how a small change of the input affects its output. The spectral condition number of a symmetric positive-definite matrix A is given by :

$$\kappa(A) := \frac{\lambda_{\max}}{\lambda_{\min}}, \quad (8.9)$$

where λ_{\max} and λ_{\min} are the largest (respectively the smallest) eigenvalues of the matrix A . When $\kappa(A)$ is large the system is said to be ill-conditioned and one should expect numerical problems.

If we want to solve the system $Ax = \mathbf{rhs}$ with a computer, we can first remark the impact of the condition number $\kappa(A)$ on the precision of the solution $x = A^{-1}\mathbf{rhs}$ found. Indeed, since the computations are made in floating point arithmetic the result x given by a computer is subject to rounding, cancellation and absorption errors.

Remark 8.2.1. *The precision for solving a linear system depends on the machine epsilon (ε_{ma}), and on the condition number of the matrix in the following way :*

$$\text{err}(A^{-1}) \propto \varepsilon_{ma} \kappa(A) \quad (8.10)$$

For instance, for the double precision, the epsilon machine is $2^{-53} \approx 1.1\text{e} - 16$ so that if the condition number is $1\text{e}10$ the solution will have roughly six digits of accuracy.

There exist two different numerical methods solve the linear system $Ax = \mathbf{rhs}$ with A a sparse matrix :

- An iterative method based on the conjugate gradient algorithm potentially with preconditioning.
- A direct sparse method that performs Gaussian elimination.

If we use iterative method to solve $x = A^{-1}\mathbf{rhs}$ an approximation error is added up to the intrinsic floating point arithmetic error described in Remark 8.2.1. It is important to quantify how the additional approximation error is decreasing with respect to the number of iterations made by the iterative method.

Lemma 8.2.2. *The vanilla conjugate gradient algorithm solves $Ax_{\star} = b$. The convergence rate of the iterates x_m toward the solution x_{\star} is given by [115] :*

$$\|x_{\star} - x_m\|_A \leq 2 \left[\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right]^m \|x_{\star} - x_0\|_A,$$

where x_0 is the initial guess, and $\|x\|_A := \sqrt{(Ax, x)}$.

The rate of convergence given in Lemma 8.2.2 can be improved by preconditioning the matrix A . However the speed of convergence towards the solution x_{\star} depends intrinsically on the condition number of the matrix A .

8.3 Optimal transport optimization for curvling

In this section we apply some of the theoretical results described in the previous section to the special case of curvling (when the Dirac masses are taken along a curve), in order to detail how it differs from the regular semi-discrete setting (where the Dirac masses are constraints-free). With these differences new numerical challenges appear. Indeed, solving an optimal transport problem is more difficult when the Dirac masses are taken along a line. In the following section, we summarize geometrical characteristic of the Laguerre tessellations in the curvling setting. These features are premonitory signs of the future hardships in optimizing the Monge Kantorovitch functional.

8.3.1 Geometric properties of Laguerre tessalations in the curvling setting

We detail in Section 1.2.3 how the use of the Laguerre tessellation allows us to cast an infinite dimensional problem (continuous valued coupling) into a finite dimensional concave maximization problem. The transport plan is then given by the Laguerre cell associated with each Dirac mass; and the optimization involves finding the optimal Laguerre tessellation which satisfies the following optimality condition :

$$\forall i, \mu(\mathcal{L}_i) = \nu_i.$$

In practice, this capacity matching problem is solved by shrinking or extending the size of the Laguerre cells accordingly to the μ -mass they lie on. This optimization is driven by the Lagrange multiplier ϕ , indeed every point x belonging to the Laguerre cell i satisfies :

$$\|\mathbf{x}_i - x\|_2^2 - \phi_i \leq \|\mathbf{x}_j - x\|_2^2 - \phi_j, \quad \forall j \neq i. \quad (8.11)$$

From inequality (8.11), one can easily deduce that when the difference between two multipliers $\phi_i - \phi_j$ of two adjacent cells \mathcal{L}_i and \mathcal{L}_j varies, the boundary $\mathcal{L}_i \cap \mathcal{L}_j$ moves linearly in the outward normal direction at a rate (see Chapter 6 Equation (6.12) for details of the calculation) :

$$\frac{1}{2\|\mathbf{x}_i - \mathbf{x}_j\|}. \quad (8.12)$$

The optimization of the Monge Kantorovitch functional using the Laguerre tessellation has an elegant geometric interpretation. Thence inspecting the overall tessellation aspect provides an insightful grasp on the potential difficulties.

When the Dirac masses of the semi-discrete optimal transport are taken along a curve, the optimal Laguerre tessellation is very narrow in the tangential direction of curve as shown in Figure 8.2.

In contrast, in the bluenoise setting (when no constraints are imposed on the location of Dirac masses), the tessellation tends to be hexagonal. In two dimensions, the centroidal Voronoi tessellation (CVT) is an hexagonal mesh and some boundary pieces [103].

As a consequence of these tessellations specificities, the curvling case is harder to optimize. Indeed :

- The minimal distance between two points is small and diminishes as we discretize the curve in a thinner manner, so that the magnitude of the boundaries' displacement (8.12) is larger. For a fixed length curve and in two dimensions, when the number of points is multiplied by a factor n the typical distance between points is divided by a factor n . In contrast for the stippling case this distance divided by \sqrt{n} .
- The "*thickness*" of the Laguerre cell can be tracked using the radius of the largest inscribing circle in the Laguerre cell. Numerically, we observe that for the same number of points, this quantity is always bigger for stippling. In the curvling case, due to the quasi-alignment of points along the curve, the tessellation is made up of elongated Laguerre cells. As a consequence, the minimal perturbation to make a cell disappear for curvling is smaller in comparison with stippling.

The main conclusion to be drawn from these observations is that computing the optimal transport distance in the curvling case is trickier. Indeed, the optimization of the Kantorovitch functional with respect to the dual variable ϕ has to be performed cautiously selecting small stepsizes in order to make no Laguerre cells disappear.

8.3.2 Condition number of the Hessian in OT

Hessian describes the variations of the gradient of g which is proportional to the mass seen by each Laguerre cell. In this respect, Hessian analysis is closely linked to the geometric discussion of Section 8.3.1.

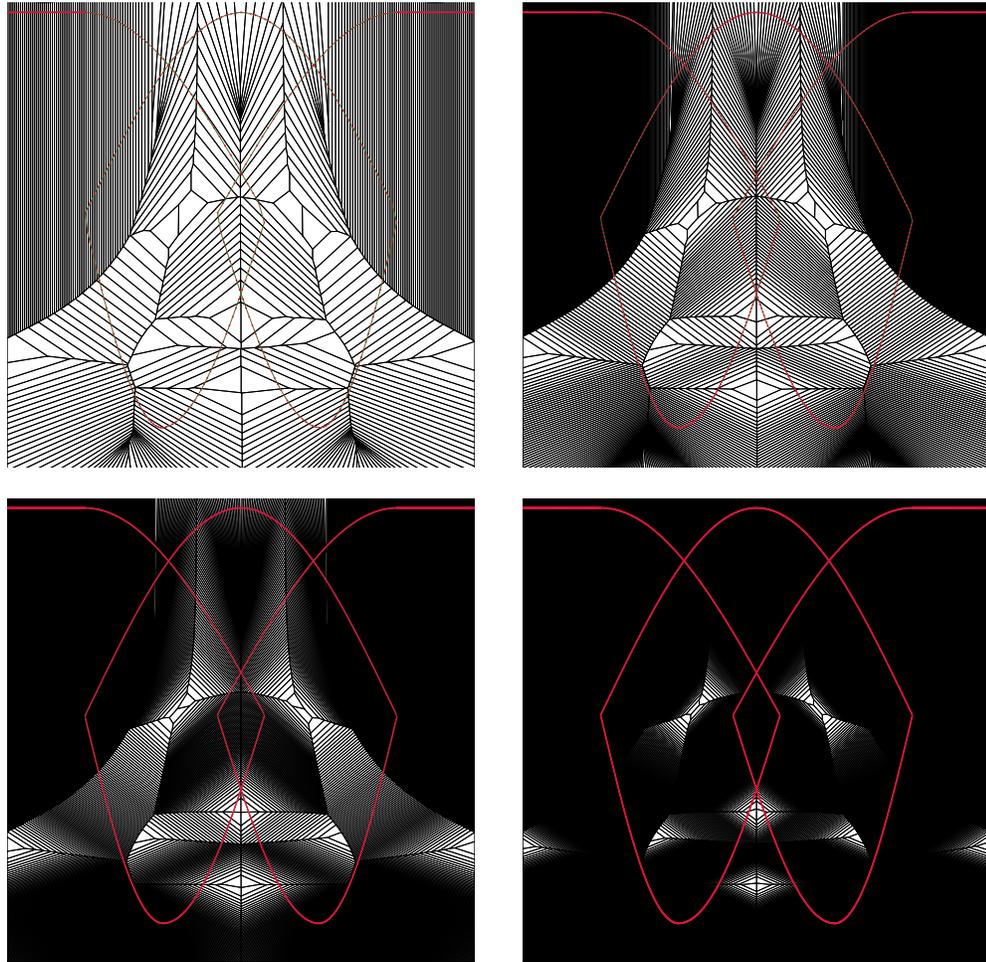


FIGURE 8.2 – Evolution of the geometry of optimal Laguerre tessellation when the number of discretization points of the curve increases. Here the green curve is discretized by red Dirac masses (from top to bottom 400, 1000, 2000, 4000).

The condition number (8.9) is the ratio of the largest and smallest eigenvalues of $H[g]$. This quantity is difficult to track. To the best of our knowledge, the sole approximate of the localizations of the eigenvalues can be obtained via Gershgorin circle theorem. This theorem tells us that the eigenvalues are contained within the following discs :

$$\lambda_i \in [0, 2P_\mu(\mathcal{L}_i)] \quad \text{with} \quad P_\mu(\mathcal{L}_i) = \sum_{j \neq i} \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|} \int_{\partial\mathcal{L}_i \cap \partial\mathcal{L}_j} d\mu(x).$$

In the curvling case the radius of these disks grows as the discretization step decreases faster than in the stippling case. Indeed the distance between the points \mathbf{x} decreases. Moreover the μ -perimeter of the Laguerre cells should be larger due to their elongated shape far away from the hexagonal tiling of stippling (honeycomb conjecture).

However, this reasoning is quite rough, and the increasing size of these disks is not enough to conclude on the precise localizations of the highest and smallest eigenvalues. Nevertheless, the spectral condition number can be computed numerically (see Figure 8.3) with linear algebra libraries. This numerical experiment confirms our intuition, for the same number of points the curvling case has an higher condition number. Moreover, when the number of points increases, the condition number of the Hessian scales quadratically in the curvling case, whereas this same quantity scales linearly in the stippling case.

8.3.2.1 Consequences of the high condition number of the Hessian

The precision of the Newton's descent direction (8.4) is key to have a good rate of convergence of the **damped Newton method** method. In Section 8.2.3 we recalled that the error is driven by the condition number of the Hessian matrix.

There is a tradeoff between the quality of the Newton direction and the time spent computing it. Although iterative methods with preconditioning seem appealing for their promptness of convergence, we did not manage to find a method that yields a precise Newton direction within decent amount of time. Indeed, in the curvling setting, having a small approximation error requires a lot of iterations of the iterative method. This is why, in our codes, we compute the Newton direction using a direct sparse solver.

Even with this choice of method the floating point arithmetic error of Remark 8.2.1 remains. We suspect this error to sully the quality of the Newton

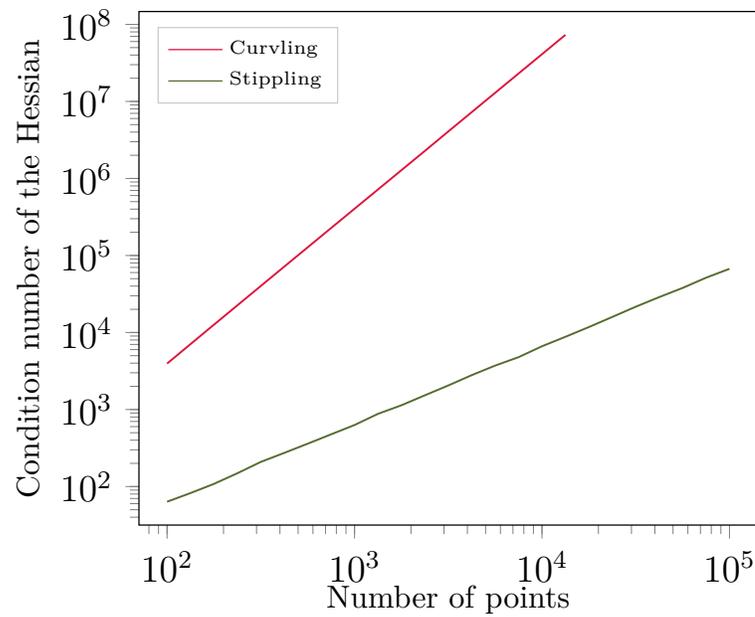


FIGURE 8.3 – Evolution of the value of the maximum eigen value of the Hessian matrix when the number of points increases.

direction found by inverting the Hessian.

8.3.3 Size of the basin of attraction in the semi-discrete OT

In Section 8.2.1 the speed of convergence of the **damped Newton method** depends on the quickness to reach the basin of attraction, where the convergence rate of the method is quadratic. This section, is devoted to analyze how rapidly these basins can be reached in both stippling and curvling when computing of the semi-discrete optimal transport.

The Lipschitz constant M of the Hessian matrix cannot be computed explicitly, since we do not have the expression of the third order derivative of f . Formally, third order derivatives measure the variation of the second order derivatives of the Kantorovitch functional which are computed with the μ -perimeter of the Laguerre cells. In the light of the geometrical Section 8.3.1, variations of the perimeter of a hexagonal tessellation seem to be smoother than a thin rectangular tiling.

Moreover empirical estimates of condition number the Hessian performed in Section 8.3.2 are higher in the curvling case. Now using the damped Newton's iteration estimate (8.8) one can reasonably think that the damped Newton will be lengthy in the curvling setting. In the following paragraph we cook up a numerical example that illustrates how attaining the *Newton phase* is more difficult in the curvling than in the stippling case.

In this experiment we take a background measure that is uniform on $[0, 1]^2$ for an increasing number of points. We distinguish three point configurations :

- Points uniformly taken on the whole support of the measure $[0, 1]^2$ *stippling 1*.
- Points uniformly taken on a sub-domain of the support $[0, 1/2]^2$ *stippling 2*.
- Points taken along a curve : *curvling* (see Figure 8.2).

The variable γ is computed using the relation :

$$\gamma \leq \leq \frac{f(x_0) - f(x_*)}{n_{\text{dn}}},$$

with n_{dn} the number of damped Newton iterations, it is proportional to the cost function's improvement made following the Newton descent direction.

For the *stippling 1* case we observe that the Voronoi tiling (the initialization) is really close to the optimal Laguerre tessellation, and a unique

# of points	<i>stippling 1</i>		<i>stippling 2</i>		<i>curvling</i>	
	γ	# dn	γ	# dn	γ	# dn
5000	$2.7 \cdot 10^{-4}$	1	$1.4 \cdot 10^{-2}$	5	$1.4 \cdot 10^{-4}$	22
10000	$8.4 \cdot 10^{-5}$	1	$9.2 \cdot 10^{-3}$	7	$5.9 \cdot 10^{-5}$	25
25000	$3.2 \cdot 10^{-5}$	1	$9.2 \cdot 10^{-3}$	8	$3.6 \cdot 10^{-5}$	32
50000	$2.1 \cdot 10^{-5}$	1	$7.5 \cdot 10^{-3}$	10	$2.1 \cdot 10^{-5}$	45

TABLE 8.1 – Evolution of the upper bound on γ and # dn number of damped newton iterations when the number of points increases.

damped Newton iteration is enough to converge. Since the first guess is close to the optimal solution, the cost function decrease is small which leads to a small upper bound of γ .

In the *stippling 2* case the initial guess is deliberately further away from the final minimum, as a consequence the number of damped Newton iterations increases. However the γ value remains fairly constant and the *Newton phase* is reached in at most ten iterations.

In the *curvling* case the initialization is far from the optimal value and we need many more iterations to reach the Newton phase (from 20 to 45). Moreover the value of γ decreases with the number of points.

Note that for a number of points higher than 50000, and in the curvling case, the classical damped Newton iteration starts to yield tessellations with empty cells. These empty cells cause an early termination of the algorithm since the Hessian ceases to be invertible.

In addition the choice of a uniform background measure promotes the regularity of the derivatives, and thus gives a very optimistic overview of the damped Newton algorithm in the curvling case. This example shows the limitations of this algorithm for our setting.

However it is hard to determine if the optimization problem is intrinsically difficult with small basins of attraction for its minimizer, or if the problem is only due to numerical errors inverting the Hessian during the Newton step. The rest of this section is devoted to finding an optimization algorithm which overcomes these difficulties.

8.3.4 Non definite Hessian in OT

Let us first point out a severe pitfall in the damped Newton optimization which is cells with zero μ -mass. Indeed at these points the Hessian is non-invertible and the convergence theory developed in [80] does not apply.

In the previous sections we analyzed the limits of the damped Newton algorithm for curving. Its principal deficiency is that for a given descent direction, and for a step-size which satisfies an Armijo-Wolfe condition empty cells can appear. The first idea that should come to mind to fix this problem is to enforce an extra condition in the line-search :

$$\forall i, \quad \mu(\mathcal{L}_i) > 0 \quad (8.13)$$

This supposedly naïve method allows us to cope with empty Laguerre cells as long as the initialization ϕ_{init} is conform ; that is, each Laguerre cell issued from the weighted diagram $(\mathbf{x}, \phi_{\text{init}})$ has a non-empty μ -mass . In practice, and for difficult problems the non-empty cell condition (8.13) is difficult to meet, and the chosen stepsizes are small, hence the line-search procedure is time consuming. Since the overall method selects small stepsizes, the Newton metric can undergo slight modifications for a long period, the convergence of the algorithm can be very slow.

In the damped Newton method the stepsize is selected in a geometric way ; a hyper-parameter κ is chosen in $]0, 1[$ and the stepsize is then set as $\tau^{(k)} = \kappa^i$ for the smallest integer i which satisfies the Armijo-Wolfe conditions. Picking an admissible $\tau^{(k)}$ can therefore be cumbersome. In the following paragraph we propose a method that explicitly computes the minimal admissible stepsize.

First suppose that at iteration k the Laguerre tessellation with weights $\phi^{(k)}$ has no hidden cells, and that the update $\phi^{(k+1)} = \phi^{(k)} + \tau^{(k)}d^{(k)}$ makes some cells disappear. The algorithm in 2D is the following : list all the points with an empty Laguerre cell. For each point \mathbf{x}_h in that list, find the face of the regular triangulation (triangle) in which the point lies, and denote $\{i_1; i_2, i_3\}$ the indexes of the corresponding vertices of that face. Note that if only one hidden point lies in the face $(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \mathbf{x}_{i_3})$ its Laguerre cell has to be in competition with $\mathcal{L}_{i_1}, \mathcal{L}_{i_2}$ and \mathcal{L}_{i_3} , due to the duality between the regular triangulation and the Laguerre tessellation. Moreover the Laguerre cell \mathcal{L}_h will spawn for a sufficiently small τ at the intersection $\mathcal{L}_{i_1} \cap \mathcal{L}_{i_2} \cap \mathcal{L}_{i_3}$. This point of competition x_τ satisfies the following equation :

$$\forall i \in \{i_1; i_2, i_3\}, \quad \|\mathbf{x}_i - x_\tau\|^2 - \phi_i - \tau d_i = \text{constant}. \quad (8.14)$$

This point can be computed explicitly solving the following linear system :

$$Ax_\tau = \mathbf{b} + \tau \mathbf{c},$$

with

$$A = \begin{pmatrix} 2(\mathbf{x}_{i_2} - \mathbf{x}_{i_1})^T \\ 2(\mathbf{x}_{i_3} - \mathbf{x}_{i_1})^T \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} \|\mathbf{x}_{i_2}\|^2 - \|\mathbf{x}_{i_1}\|^2 + \phi_{i_1} - \phi_{i_2} \\ \|\mathbf{x}_{i_3}\|^2 - \|\mathbf{x}_{i_1}\|^2 + \phi_{i_1} - \phi_{i_3} \end{pmatrix} \quad \mathbf{c} = \begin{pmatrix} d_{i_1} - d_{i_2} \\ d_{i_1} - d_{i_3} \end{pmatrix}.$$

The value of parameter τ that discloses \mathcal{L}_h can be evaluated via the following formula :

$$\tau = \frac{\|\mathbf{x}_{i_1}\|^2 - \|\mathbf{x}_h\|^2 + \phi_h - \phi_{i_1} + 2 \langle \mathbf{x}_h - \mathbf{x}_{i_1}, A^{-1} \mathbf{b} \rangle}{d_{i_1} - d_h - 2 \langle \mathbf{x}_h - \mathbf{x}_{i_1}, A^{-1} \mathbf{c} \rangle}. \quad (8.15)$$

Now two adaptations of the algorithm can be considered :

- One-pass underestimation of τ_{\min} : in this case the algorithm proceeds as explained above and returns a potentially lower bound of the maximal τ . When several hidden points are in the same face one iteration of the presented algorithm may not pick the maximal value of τ that discloses all the hidden vertices. Indeed, the computation of x_τ is based on the underlying power diagram connectivity that yields the indexes of the competing Laguerre cells. When several points are contained in same face this connectivity is incorrect.
- Exact computation of τ , in this algorithm, if a hidden vertex is alone in a power diagram face then the value of τ is computed with Equation (8.15) and compared to the minimal value found so far. If several hidden vertices share the same power diagram face, all the possible values of τ are computed and the highest value : τ_m is correct, and compared to the current minimal value of τ . Indeed, the points $(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \mathbf{x}_{i_3})$ can be in competition with at most one hidden vertex, τ_m is the value that spawns the Laguerre cell associated to this hidden vertex. Then for all the remaining vertices of the face, recompute a power diagram for a $\tau < \tau_m$. Repeat this step until there are no more points to process. This algorithm computes the minimal value

τ_{\min} for which the weight update $\phi^{(k)} + \tau_{\min}^{(k)} d^{(k)}$ yields a Laguerre tessellation where the cells are at least reduced to a point. Then we set $\phi^{(k+1)} = \phi^{(k)} + \xi \tau_{\min}^{(k)} d^{(k)}$ with ξ in $]0, 1[$.

Note that if two or more hidden vertices are contained in the same face, the maximum value of τ within the face is exact, the remaining value are lower bounds on the real τ . Indeed, the computed x_τ for these vertices is remote from the actual point of competition, hence the resulting τ computed with (8.15) is pessimistic.

These two algorithms are written in C++ using the CGAL [128] library, and they benefit from its swift implementation. This routine reduces the time spent in the linesearch when there are hidden cells, but the Amijo-Wolfe and the non-empty condition (8.13) still need to be enforced. This method yields a cut-off value of τ , and in the second version (exact computation) the largest admissible stepsize we could take to ensure that none of the cells are hidden.

Note that there is a difference between hidden and empty cells, hidden cells are Laguerre cells that are reduced to a void set, on the opposite, empty cells have zero μ -mass but are not reduced to the empty set. The explicit computation of τ only deals with hidden cells.

Indeed, for background measure μ with a non-ubiquitous support, some Laguerre cells can be non-hidden but empty, even with the natural initialization choice $\phi_{\text{init}} = 0_{\mathbb{R}^n}$ which yields a Voronoi tessellation. Nevertheless, under the assumption that the starting point ϕ_{init} has no empty Laguerre cells, the condition (8.13) can be upheld through the iterates even if it requires taking small steps. Notwithstanding this nice property when optimizing the Monge Kantorovitch functional with respect to the ϕ variable, when we optimize with respect to the Dirac masses locations \mathbf{x} , there is no guarantee of finding a valid ϕ_{init} .

We now collect all the evidences that points towards the necessity of a change of optimization method :

- The resulting damped Newton algorithm assumes the existence and continuity of the Hessian and its invert which is simply not embedded in our framework.
- The high condition number of the Hessian leads numerical error that impacts the computation of the Newton direction.
- For the curvling setting the quality of the metric given by the Newton method is poor, at least in the early stages of the optimization.
- If some Laguerre cells are empty, the algorithm stops. In section 8.3.6,

we describe a multiscale algorithm that provides us a good initialization for the Dirac masses positions and the Lagrange multipliers. During algorithm we can encounter empty Laguerre cells. For this reason, the damped newton algorithm is inadequate for multi-scaled runs.

8.3.5 Regularized Newton method for Curvling

In this Section, we search for a more robust alternative to the damped Newton algorithm for the curvling case. Using the ideas developed in [111], we regularize the Hessian adding an identity term proportional to the norm of the gradient. This idea is quite similar to a method developed in the sixties [91], and known as the Levenberg-Marquardt regularization, which is a Newton-type method (8.5) for the following approximation of the Hessian matrix :

$$B_\lambda = - (H[f] + \lambda \mathbf{I})^{-1} \nabla f, \quad (8.16)$$

with \mathbf{I} the identity matrix, and where λ is a parameter that tunes the blending between first order and second order descent directions. Indeed, for several choices of λ we recognize well-known methods :

- For the choice $\lambda = 0$, B_0 is the Newton direction.
- For the choice $\lambda = \|\nabla f\|$, $B_{\|\nabla f\|}$ is the regularized Newton direction.
- The choice $\lambda \rightarrow \infty$, B_λ is equivalent to the steepest descent direction with step $1/\lambda$.

In Table 8.2, we reproduce the numerical experiment carried out in Table 8.1 in order compare convergence efficiency of the Newton direction against regularized Newton direction. The first conclusion to be drawn from this experiment is that the Newton direction cannot be beaten close to a minimizer (*stippling 1* case). Indeed, close to a minimizer the function is close to quadratic and the metric given by Newton is foolproof. The second conclusion is that when the problem become harder (*curvling*), in the sense that it requires more iterations to attain the *Newton phase* for the damped Newton algorithm, the regularized Newton method converges promptly.

In Section 8.2, we emphasize the importance of the condition number of the Hessian. In fact, the local “quality” of the Hessian direction can be estimated with it. A high condition number indicates huge variations of the quadratic approximation, hence a coarse direction to follow. The numerical tests tend to suggest that the regularized Newton is in this case a better direction to follow in terms of the decrease of the gradient’s. In the *curvling*

# of point	<i>stippling 1</i>		<i>stippling 2</i>		<i>curvling</i>	
	# DN	# RNM	# DN	# RNM	# DN	# RNM
5000	1	3	5	10	22	15
10000	1	4	7	8	25	17
25000	1	4	8	10	32	18
50000	1	3	10	11	45	20

TABLE 8.2 – Comparison of the damped Newton algorithm (DN) against the regularized Newton method (RNM)

case, and for 50000 points the regularized Newton method converges in half of the iterations needed for the classical damped Newton method. Note that for fairness, the two algorithms are tested with the same Armijo Wolfe condition and the same linesearch in such a way that only the direction differs.

On the other hand, in Section 8.3.2 we discuss numerical quality of the descent direction d found by solving $B_\lambda d = -\nabla f$ with respect to the spectral condition number of the matrix B_λ . The Levenberg-Marquardt regularization decreases the spectral condition number if it is greater than 1, so that the regularized descent direction is less tarnished by numerical error.

The global convergence of the Monge Kantorovitch maximization problem can be extended when its functional is less smooth, that is not \mathcal{C}^2 , which is the case when $\mu(\mathcal{L}_i(\mathbf{x}, \phi)) = 0$. Thanks to the regularized Newton method, global convergence and quadratic rate in the vicinity of the maximizer can be guaranteed, see Proposition 5.3.3 for details. However the numerical experiment of Table 8.2 illustrates that the damped Newton algorithm outperforms the regularized Newton method in the more straightforward *stippling 1* case.

Based on this analysis, the engineering idea involves in tuning the regularization parameter λ of the Levenberg-Marquardt regularization (8.16). Indeed in simple cases, we want to blend more second-order information and in the hard cases more first-order information. In Algorithm 8, the decrease of the regularization is driven by the parameter γ_1 , whereas γ_2 controls the increase of regularization.

Note that condition (8.13) is lessened though condition $\mathbf{Empty}(\phi + d) \leq \mathbf{Empty}(\phi)$, where \mathbf{Empty} is a function that returns the number of Laguerre cells such that their mass is lower than 10^{-12} . The Algorithm 8 is tailored to work with zero μ -mass cells, hence we only enforce the non-increasing number of empty cells condition. This new condition allows take larger steps

Algorithm 8 Levenberg-Marquardt regularization for semi-discrete optimal transport

Inputs :

Oracle for $(f, \nabla f, H[f])$, **Empty** number of empty Laguerre cells,
 c_1 and c_2 (line-search parameter),
 γ_1 and γ_2 (regularization parameters),
tol, iterMax termination criterion
 ϕ initial point

```

1: itt  $\leftarrow$  0
2:  $\lambda \leftarrow \frac{\|\nabla f\|}{\gamma_1}$ .
3: while  $\|\nabla f\| > \text{tol}$  and  $\text{itt} < \text{iterMax}$  do
    $\lambda \leftarrow \min(\gamma_1 \lambda, \|\nabla f\|)$ 
    $d \leftarrow -(H[f](\phi) + \lambda \mathbf{I})^{-1} \nabla f(\phi)$ 
4:   while  $(\nabla f(\phi + d))^T d < c_2 \nabla f(\phi)^T d$  and  $f(\phi + d) < f(\phi) + c_1 \nabla f(\phi)^T d$ 
5:     and  $\text{Empty}(\phi + d) > \text{Empty}(\phi)$  do
6:        $\lambda \leftarrow \gamma_2 \lambda$ 
7:        $d \leftarrow -(H[f](\phi) + \lambda \mathbf{I})^{-1} \nabla f(\phi)$ 
8:     end while
9:     itt  $\leftarrow$  itt + 1
10:     $\phi \leftarrow \phi + d$ 
11: end while

```

which enable the method to reach the *Newton phase* faster.

8.3.6 Multiscale algorithm for Curvling

In this section we present a multiscale algorithm that allows us to quickly obtain curves carried by a large number of Dirac masses. In order to have a scheme that scales consistently with the number of points we have to normalize the finite difference operators defined in Equations (5.29) and (5.30).

The optimization of the positions of the Dirac masses (denoted \mathbf{x}) can be very difficult for a large number of points and for an arbitrary initialization. The objective of this method is to produce a good initialization for n points. Following the ideas of [94] we develop a multi-scale algorithm for both the dual variable ϕ and the positions \mathbf{x} . This method successively solves the optimization of easier stippling subproblems with $n2^{-\ell}$ points. For each of these subproblems we compute the optimal transport distance before updating the positions. This optimization is repeated until obtaining a discretized curve sufficiently close to the target image. We then apply the upscaling step that involves doubling the number of Dirac masses and updating $\ell \leftarrow \ell - 1$. This procedure is repeated until $\ell = 0$. The positions of the Dirac masses for an upscale from the level $\ell + 1$ to ℓ reads as :

$$\forall i \in \llbracket 0, n2^{-\ell-1} - 2 \rrbracket, \quad \left\{ \begin{array}{l} \mathbf{x}_{2i}^{\ell} = \mathbf{x}_i^{\ell+1} \\ \mathbf{x}_{2i+1}^{\ell} = \mathbf{x}_i^{\ell+1} + \mathbf{x}_{i+1}^{\ell+1} \\ \mathbf{x}_{n2^{-\ell}-2}^{\ell} = \mathbf{x}_{n2^{-(\ell+1)}-1}^{\ell+1} \\ \mathbf{x}_{n2^{-\ell}-1}^{\ell} = \frac{1}{2} \left(\mathbf{x}_0^{\ell+1} + \mathbf{x}_{n2^{-(\ell+1)}-1}^{\ell+1} \right) \end{array} \right.$$

for a periodic curve.

In the non-periodic case, the upscaling is defined by :

$$\forall i \in \llbracket 0, n2^{-l-1} - 2 \rrbracket, \begin{cases} \mathbf{x}_{2i}^\ell &= \mathbf{x}_i^{\ell+1} \\ \mathbf{x}_{2i+1}^\ell &= \mathbf{x}_i^{\ell+1} + \mathbf{x}_{i+1}^{\ell+1} \\ \mathbf{x}_{n2^{-l}-2}^\ell &= \mathbf{x}_{n2^{-(\ell+1)}-1}^{\ell+1} \\ \mathbf{x}_{n2^{-l}-1}^\ell &= \frac{1}{2} \left(3\mathbf{x}_{n2^{-(\ell+1)}-1}^{\ell+1} - \mathbf{x}_{n2^{-(\ell+1)}-2}^{\ell+1} \right) \end{cases}$$

The idea in [94] is to assign to the Lagrange multiplier of each Dirac mass of level ℓ , the multiplier's value of the closest Dirac mass in the level $\ell + 1$. In our multiscale setting, finding the nearest point is trivial. However when the index of a point of level ℓ is odd the point is equidistant from two points of level $\ell + 1$. In that case we set the Lagrange multiplier equal to the mean of the Lagrange multipliers of Dirac masses at the same distance. The value of the Lagrange multiplier of level ℓ , ϕ_i^ℓ is then given by :

$$\forall i \in \llbracket 0, n2^{-l-1} - 2 \rrbracket, \begin{cases} \phi_{2i}^\ell &= \phi_i^{\ell+1} \\ \phi_{2i+1}^\ell &= \phi_i^{\ell+1} + \phi_{i+1}^{\ell+1} \\ \phi_{n2^{-l}-2}^\ell &= \phi_{n2^{-(\ell+1)}-1}^{\ell+1} \\ \phi_{n2^{-l}-1}^\ell &= \frac{1}{2} \left(\phi_0^{\ell+1} + \phi_{n2^{-(\ell+1)}-1}^{\ell+1} \right) \end{cases}$$

in the periodic case. In the non-periodic case it is given by :

$$\forall i \in \llbracket 0, n2^{-l-1} - 2 \rrbracket, \begin{cases} \phi_{2i}^\ell &= \phi_i^{\ell+1} \\ \phi_{2i+1}^\ell &= \phi_i^{\ell+1} + \phi_{i+1}^{\ell+1} \\ \phi_{n2^{-l}-2}^\ell &= \phi_{n2^{-(\ell+1)}-1}^{\ell+1} \\ \phi_{n2^{-l}-1}^\ell &= \frac{1}{2} \left(3\phi_{n2^{-(\ell+1)}-1}^{\ell+1} - \phi_{n2^{-(\ell+1)}-2}^{\ell+1} \right) \end{cases}$$

Empirically, we notice that initializations given by the multiscale algorithm converge up to 5 times faster than arbitrary initialization curves (for a number of points larger than 10^5). Hence the multiscale approach is very appealing for measures with hundreds thousands of points : in this case a multiscale approach leads to 2 – 3 times faster convergence (even if it requires the resolution of a few stippling subproblems).



FIGURE 8.4 – Multiscale example with from left to right and top to bottom $2^{14}, 2^{15}, 2^{16}, 2^{17}$ points to discretize the curve.



FIGURE 8.5 – Multiscale example with from left to right and top to bottom $2^{18}, 2^{19}, 2^{20}$ points to discretize the curve and the original image.

Bibliographie

- [1] B. Adcock, A. C. Hansen, C. Poon, and B. Roman. Breaking the coherence barrier : A new theory for compressed sensing. In *Forum of Mathematics, Sigma*, volume 5. Cambridge University Press, 2017.
- [2] E. Akleman, Q. Xing, P. Garigipati, G. Taubin, J. Chen, and S. Hu. Hamiltonian cycle art : Surface covering wire sculptures and duotone surfaces. *Computers & Graphics*, 37(5) :316–332, 2013.
- [3] Y. Asami. A note on the derivation of the first and second derivative of objective functions in geographical optimization problems. *Journal of the Faculty of Engineering, The University of Tokio (B)*, 41(1) :1–13, 1991.
- [4] K. Atasu, T. Parnell, C. Dünner, M. Sifalakis, H. Pozidis, V. Vasileiadis, M. Vlachos, C. Berrospi, and A. Labbi. Linear-complexity relaxed word mover’s distance with gpu acceleration. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 889–896. IEEE, 2017.
- [5] F. Aurenhammer. Power diagrams : properties, algorithms and applications. *SIAM Journal on Computing*, 16(1) :78–96, 1987.
- [6] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3) :345–405, 1991.
- [7] F. Aurenhammer, F. Hoffmann, and B. Aronov. Minkowski-type theorems and least-squares clustering. *Algorithmica*, 20(1) :61–76, 1998.
- [8] M. Balzer, T. Schlömer, and O. Deussen. *Capacity-constrained point distributions : a variant of Lloyd’s method*, volume 28. ACM, 2009.
- [9] J.-D. Benamou and Y. Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3) :375–393, 2000.

- [10] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2) :A1111–A1138, 2015.
- [11] J.-D. Benamou, G. Carlier, Q. Mérigot, and E. Oudet. Discretization of functionals involving the monge–ampère operator. *Numerische mathematik*, 134(3) :611–636, 2016.
- [12] D. P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21(1) :152–171, 1981.
- [13] D. P. Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.
- [14] D. P. Bertsekas and D. A. Castanon. The auction algorithm for the transportation problem. *Annals of Operations Research*, 20(1) :67–96, 1989.
- [15] D. P. Bertsekas and J. Eckstein. Dual coordinate step methods for linear network flow problems. *Mathematical Programming*, 42(1-3) :203–243, 1988.
- [16] B. Beygu, K. Kreckel, R. van de Weygaert, J. van der Hulst, and J. Van Gorkom. An interacting galaxy system along a filament in a void. *The Astronomical Journal*, 145(5) :120, 2013.
- [17] J.-D. Boissonnat, C. Wormser, and M. Yvinec. Curved voronoi diagrams. In *Effective Computational Geometry for Curves and Surfaces*, pages 67–116. Springer, 2006.
- [18] N. Bonneel, M. Van De Panne, S. Paris, and W. Heidrich. Displacement interpolation using lagrangian mass transport. In *ACM Transactions on Graphics (TOG)*, volume 30, page 158. ACM, 2011.
- [19] D. P. Bourne, B. Schmitzer, and B. Wirth. Semi-discrete unbalanced optimal transport and quantization. *arXiv preprint arXiv :1808.01962*, 2018.
- [20] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [21] C. Boyer, N. Chauffert, P. Ciuciu, J. Kahn, and P. Weiss. On the generation of sampling schemes for magnetic resonance imaging. *SIAM Journal on Imaging Sciences*, 9(4) :2039–2072, 2016.
- [22] C. Boyer, P. Weiss, and J. Bigot. An algorithm for variable density sampling with block-constrained acquisition. *SIAM Journal on Imaging Sciences*, 7(2) :1080–1107, 2014.

- [23] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.
- [24] H. Brezis, P. G. Ciarlet, and J. L. Lions. *Analyse fonctionnelle : théorie et applications*, volume 91. Dunod Paris, 1999.
- [25] N. Chauffert, P. Ciuciu, J. Kahn, and P. Weiss. Comment représenter une image avec un spaghetti? In *GRETSI*, 2015.
- [26] N. Chauffert, P. Ciuciu, J. Kahn, and P. Weiss. A projection method on measures sets. *Constructive Approximation*, 45(1) :83–111, 2017.
- [27] N. Chauffert, P. Weiss, J. Kahn, and P. Ciuciu. Gradient waveform design for variable density sampling in Magnetic Resonance Imaging. *arXiv preprint arXiv :1412.4621*, 2014.
- [28] Z. Chen, Z. Shen, J. Guo, J. Cao, and X. Zeng. Line drawing for 3D printing. *Computers & Graphics*, 2017.
- [29] City of Philadelphia. Open data in the philadelphia region. <https://www.opendataphilly.org/dataset/crime-incidents>, 2017.
- [30] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.
- [31] L. Comtet. *Advanced Combinatorics : The art of finite and infinite expansions*. Springer Science & Business Media, 2012.
- [32] M. Cuturi. Sinkhorn distances : Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.
- [33] F. de Goes. *Geometric discretization through primal-dual meshes*. PhD thesis, California Institute of Technology, 2014.
- [34] F. De Goes, K. Breeden, V. Ostromoukhov, and M. Desbrun. Blue noise through optimal transport. *ACM Transactions on Graphics (TOG)*, 31(6) :171, 2012.
- [35] F. de Goes, C. Wallez, J. Huang, D. Pavlov, and M. Desbrun. Power particles : an incompressible fluid solver based on power diagrams. *ACM Trans. Graph.*, 34(4) :50–1, 2015.
- [36] F. de Gournay, J. Kahn, and L. Lebrat. 3/4-discrete optimal transport. *arXiv preprint arXiv :1806.09537*, 2018.

- [37] F. de Gournay, J. Kahn, and L. Lebrat. Approximation of curves with piecewise constant or piecewise linear functions. *arXiv preprint arXiv :1909.04582*, 2019.
- [38] F. de Gournay, J. Kahn, and L. Lebrat. Differentiation and regularity of semi-discrete optimal transport with respect to the parameters of the discrete measure. *Numerische Mathematik*, 141(2) :429–453, 2019.
- [39] F. de Gournay, J. Kahn, L. Lebrat, and W. Pierre. Approches variationnelles pour le stippling : distances l2 ou transport optimal? In *GRETSI 2017 XXVI*, Sept. 2017.
- [40] F. de Gournay, J. Kahn, L. Lebrat, and W. Pierre. Optimal transport approximation of 2-dimensional measures. *SIAM Journal on Imaging Sciences*, 12(2) :762–787, 2019.
- [41] J. B. Du. Interactive Media Arts. <https://jackbdu.wordpress.com/category/ima-capstone/>, 2017.
- [42] Q. Du, M. Emelianenko, and L. Ju. Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. *SIAM journal on numerical analysis*, 44(1) :102–119, 2006.
- [43] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations : Applications and algorithms. *SIAM review*, 41(4) :637–676, 1999.
- [44] Q. Du and M. Gunzburger. Grid generation and optimization based on centroidal voronoi tessellations. *Applied Mathematics and Computation*, 133(2) :591–607, 2002.
- [45] J. G. Dunham. Optimum uniform piecewise linear approximation of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1) :67–75, 1986.
- [46] R. A. Dwyer. Higher-dimensional voronoi diagrams in linear expected time. *Discrete & Computational Geometry*, 6(3) :343–367, 1991.
- [47] I. Z. Emiris and M. I. Karavelas. The predicates of the apollonius diagram : algorithmic analysis and implementation. *Computational Geometry*, 33(1-2) :18–57, 2006.
- [48] L. C. Evans, J. Spruck, et al. Motion of level sets by mean curvature I. *J. Diff. Geom.*, 33(3) :635–681, 1991.
- [49] J. Feydy, T. Séjourné, F.-X. Vialard, S.-I. Amari, A. Trouvé, and G. Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. *arXiv preprint arXiv :1810.08278*, 2018.

- [50] R. Flamary, M. Cuturi, N. Courty, and A. Rakotomamonjy. Wasserstein discriminant analysis. *arXiv preprint arXiv :1608.08063*, 2016.
- [51] R. Fletcher. On the barzilai-borwein method. In *Optimization and control with applications*, pages 235–256. Springer, 2005.
- [52] R. W. Floyd. An adaptive algorithm for spatial gray-scale. In *Proc. Soc. Inf. Disp.*, volume 17, pages 75–77, 1976.
- [53] M. Fornasier, J. Haškovec, and G. Steidl. Consistency of variational continuous-domain quantization via kinetic theory. *Applicable Analysis*, 92(6) :1283–1298, 2013.
- [54] A. Galichon. *Optimal transport methods in economics*. Princeton University Press, 2018.
- [55] T. O. Gallouët and Q. Mérigot. A lagrangian scheme à la brenier for the incompressible euler equations. *Foundations of Computational Mathematics*, 18(4) :835–865, 2018.
- [56] W. Gangbo and R. J. McCann. The geometry of optimal transportation. *Acta Mathematica*, 177(2) :113–161, 1996.
- [57] M. T. Gastner and M. Newman. Optimal design of spatial distribution networks. *Physical Review E*, 74(1) :016117, 2006.
- [58] A. Genevay, G. Peyré, and M. Cuturi. Learning generative models with sinkhorn divergences. *arXiv preprint arXiv :1706.00292*, 2017.
- [59] P. Giselsson and S. Boyd. Linear convergence and metric selection for douglas-rachford splitting and ADMM. *IEEE Transactions on Automatic Control*, 62(2) :532–544, 2017.
- [60] R. Glowinski. On alternating direction methods of multipliers : a historical perspective. In *Modeling, simulation and optimization for science and technology*, pages 59–82. Springer, 2014.
- [61] M. Gräf, D. Potts, and G. Steidl. Quadrature errors, discrepancies, and their relations to halftoning on the torus and the sphere. *SIAM Journal on Scientific Computing*, 34(5) :A2760–A2791, 2012.
- [62] S. Graf and H. Luschgy. *Foundations of quantization for probability distributions*. Springer, 2007.
- [63] G. N. Grapiglia and Y. Nesterov. Regularized Newton Methods for Minimizing Functions with Hölder Continuous Hessians. *SIAM Journal on Optimization*, 27(1) :478–506, 2017.

- [64] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of computational physics*, 73(2) :325–348, 1987.
- [65] P. M. Gruber. Optimum quantization and its applications. *Advances in Mathematics*, 186(2) :456–497, 2004.
- [66] P. Gwosdek, C. Schmaltz, J. Weickert, and T. Teuber. Fast electrostatic halftoning. *Journal of real-time image processing*, 9(2) :379–392, 2014.
- [67] T.-X. He. Eulerian polynomials and b-splines. *Journal of Computational and Applied Mathematics*, 236(15) :3763–3773, 2012.
- [68] A. Henrot and M. Pierre. *Variation et optimisation de formes : une analyse géométrique*, volume 48. Springer Science & Business Media, 2006.
- [69] A. Hertzmann. A survey of stroke-based rendering. *IEEE Computer Graphics and Applications*, 23(4) :70–81, July 2003.
- [70] S. Hiller, H. Hellwig, and O. Deussen. Beyond stippling—methods for distributing objects on the plane. In *Computer Graphics Forum*, volume 22, pages 515–522. Wiley Online Library, 2003.
- [71] F. L. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal of mathematics and physics*, 20(1-4) :224–230, 1941.
- [72] K. E. Hoff III, J. Keyser, M. Lin, D. Manocha, and T. Culver. Fast computation of generalized voronoi diagrams using graphics hardware. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 277–286. ACM Press/Addison-Wesley Publishing Co., 1999.
- [73] M. Iri, K. Murota, and T. Ohya. A fast voronoi-diagram algorithm with applications to geographical optimization problems. In *System modelling and optimization*, pages 273–288. Springer, 1984.
- [74] F. Jarre and P. L. Toint. Simple examples for the failure of Newton’s method with line search for strictly convex minimization. *Mathematical Programming*, 158(1-2) :23–34, 2016.
- [75] R. Jordan, D. Kinderlehrer, and F. Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1) :1–17, 1998.
- [76] L. V. Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pages 199–201, 1942.

- [77] L. V. Kantorovich. On a problem of monge. In *CR (Doklady) Acad. Sci. URSS (NS)*, volume 3, pages 225–226, 1948.
- [78] C. S. Kaplan, R. Bosch, et al. TSP art. In *Renaissance Banff : Mathematics, music, art, culture*, pages 301–308. Bridges Conference, 2005.
- [79] S. Y. Kim, R. Maciejewski, T. Isenberg, W. M. Andrews, W. Chen, M. C. Sousa, and D. S. Ebert. Stippling by example. In *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*, pages 41–50. ACM, 2009.
- [80] J. Kitagawa, Q. Mérigot, and B. Thibert. Convergence of a Newton algorithm for semi-discrete optimal transport. *Journal of the European Math Society (JEMS)*, To appear.
- [81] S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde. Optimal mass transport : Signal processing and machine-learning applications. *IEEE signal processing magazine*, 34(4) :43–59, 2017.
- [82] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2) :83–97, 1955.
- [83] T. Lacombe, M. Cuturi, and S. Oudot. Large scale computation of means and clusters for persistence diagrams using optimal transport. In *Advances in Neural Information Processing Systems*, pages 9770–9780, 2018.
- [84] C. Lazarus. *L'échantillonnage compressif en IRM : conception optimisée de trajectoires d'échantillonnage pour accélérer l'IRM*. PhD thesis, École doctorale Electrical, optical, bio-physics and engineering ,Orsay Essonne, 2018.
- [85] B. Lévy. A numerical algorithm for l2 semi-discrete optimal transport in 3d. *ESAIM : Mathematical Modelling and Numerical Analysis*, 49(6) :1693–1715, 2015.
- [86] B. Lévy and Y. Liu. L p centroidal voronoi tessellation and its applications. In *ACM Transactions on Graphics (TOG)*, volume 29, page 119. ACM, 2010.
- [87] B. Lévy and E. L. Schwindt. Notions of optimal transport theory and how to implement them on a computer. *Computers & Graphics*, 72 :135–148, 2018.
- [88] G. Li and T. K. Pong. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4) :2434–2460, 2015.

- [89] Y. Liu, W. Wang, B. Lévy, F. Sun, D.-M. Yan, L. Lu, and C. Yang. On centroidal Voronoi tessellation, energy smoothness and fast computation. *ACM Transactions on Graphics (ToG)*, 28(4) :101, 2009.
- [90] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2) :129–137, 1982.
- [91] D. W. Marquardt. An algorithm for least-squares estimation of non-linear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2) :431–441, 1963.
- [92] M. McAsey and L. Mou. Optimal locations and the mass transport problem. *Contemporary Mathematics*, 226 :131–148, 1999.
- [93] F. Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4) :417–487, 2011.
- [94] Q. Mérigot. A multiscale approach to optimal transport. In *Computer Graphics Forum*, volume 30, pages 1583–1592. Wiley Online Library, 2011.
- [95] Q. Mérigot. A comparison of two dual methods for discrete optimal transport. In *Geometric science of information*, pages 389–396. Springer, 2013.
- [96] Q. Mérigot, J. Meyron, and B. Thibert. An algorithm for optimal transport between a simplex soup and a point cloud. *SIAM Journal on Imaging Sciences*, 11(2) :1363–1389, 2018.
- [97] J. Meyron, Q. Mérigot, and B. Thibert. Light in power : a general and parameter-free algorithm for caustic design. In *SIGGRAPH Asia 2018 Technical Papers*, page 224. ACM, 2018.
- [98] L. Moisan. Affine plane curve evolution : A fully consistent scheme. *IEEE Transactions on Image Processing*, 7(3) :411–420, 1998.
- [99] G. Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l’Académie Royale des Sciences de Paris*, 1781.
- [100] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5) :577–685, 1989.
- [101] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1) :125–161, 2013.

- [102] Y. Nesterov. *Introductory lectures on convex optimization : A basic course*, volume 87. Springer Science & Business Media, 2013.
- [103] D. Newman. The hexagon theorem. *IEEE Transactions on information theory*, 28(2) :137–139, 1982.
- [104] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan. A General Analysis of the Convergence of ADMM. In *ICML*, pages 343–352, 2015.
- [105] A. M. Oberman and Y. Ruan. An efficient linear programming method for optimal transportation. *arXiv preprint arXiv :1509.03668*, 2015.
- [106] J. B. Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78(2) :109–129, 1997.
- [107] G. Pages and B. Wilbertz. Optimal delaunay and voronoi quantization schemes for pricing american style options. In *Numerical methods in Finance*, pages 171–213. Springer, 2012.
- [108] N. Papadakis, G. Peyré, and E. Oudet. Optimal transport with proximal splitting. *SIAM Journal on Imaging Sciences*, 7(1) :212–238, 2014.
- [109] R. Peyre. Comparison between w_2 distance and h^{-1} -norm, and localization of wasserstein distance. 2016.
- [110] B. T. Polyak. Introduction to optimization. translations series in mathematics and engineering. *Optimization Software*, 1987.
- [111] R. A. Polyak. Regularized newton method for unconstrained convex optimization. *Mathematical programming*, 120(1) :125–145, 2009.
- [112] D. Potts and G. Steidl. Fast summation at nonequispaced knots by NFFT. *SIAM Journal on Scientific Computing*, 24(6) :2013–2037, 2003.
- [113] M. Raydan. The barzilai and borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7(1) :26–33, 1997.
- [114] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *International journal of computer vision*, 40(2) :99–121, 2000.
- [115] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [116] F. Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55 :58–63, 2015.

- [117] S. Schlechtweg, T. Germer, and T. Strothotte. RenderBots Multi-Agent Systems for Direct Image Generation. In *Computer Graphics Forum*, volume 24, pages 137–148. Wiley Online Library, 2005.
- [118] C. Schmaltz, P. Gwosdek, A. Bruhn, and J. Weickert. Electrostatic halftoning. In *Computer Graphics Forum*, volume 29, pages 2313–2327. Wiley Online Library, 2010.
- [119] B. Schmitzer. A sparse multiscale algorithm for dense optimal transport. *Journal of Mathematical Imaging and Vision*, 56(2) :238–259, 2016.
- [120] L. Schumaker. *Spline functions : basic theory*. Cambridge University Press, 2007.
- [121] S. Smale. Mathematical problems for the next century. *The Mathematical Intelligencer*, 20(2) :7–15, 1998.
- [122] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional Wasserstein distances : Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4) :66, 2015.
- [123] J. Solomon, R. Rustamov, G. Leonidas, and A. Butscher. Wasserstein propagation for semi-supervised learning. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 306–314, 2014.
- [124] A. Tagliasacchi, I. Alhashim, M. Olson, and H. Zhang. Mean curvature skeletons. In *Computer Graphics Forum*, volume 31, pages 1735–1744. Wiley Online Library, 2012.
- [125] E. Tempel, R. Kipper, E. Saar, M. Bussov, A. Hektor, and J. Pelt. Galaxy filaments as pearl necklaces. *Astronomy & Astrophysics*, 572 :A8, 2014.
- [126] E. Tempel, R. S. Stoica, R. Kipper, and E. Saar. Bisous model—detecting filamentary patterns in point processes. *Astronomy and Computing*, 16 :17–25, 2016.
- [127] T. Teuber, G. Steidl, P. Gwosdek, C. Schmaltz, and J. Weickert. Dithering by differences of convex functions. *SIAM Journal on Imaging Sciences*, 4(1) :79–108, 2011.
- [128] The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 4.9 edition, 2016.

- [129] A. Tolstoi. Methods of finding the minimal total kilometrage in cargo transportation planning in space. *TransPress of the National Commissariat of Transportation*, 1 :23–55, 1930.
- [130] I. Tomek. Two algorithms for piecewise-linear continuous approximation of functions of one variable. *IEEE Transactions on Computers*, 100(4) :445–448, 1974.
- [131] R. Ulichney. *Digital halftoning*. MIT press, 1987.
- [132] C. Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.
- [133] C. Villani. *Optimal transport : old and new*, volume 338. Springer Science & Business Media, 2008.
- [134] R. Wang, Y. Xu, and Z. Xu. A spline interpretation of eulerian numbers. *arXiv preprint arXiv :0808.2349*, 2008.
- [135] W. Wang, D. Slepčev, S. Basu, J. A. Ozolek, and G. K. Rohde. A linear optimal transportation framework for quantifying and visualizing variations in sets of images. *International journal of computer vision*, 101(2) :254–269, 2013.
- [136] L.-Y. Wei. Multi-class blue noise sampling. *ACM Transactions on Graphics (TOG)*, 29(4) :79, 2010.
- [137] S. J. Wright and J. Nocedal. Numerical optimization. *Springer Science*, 35(67-68) :7, 1999.
- [138] S.-Q. Xin, B. Lévy, Z. Chen, L. Chu, Y. Yu, C. Tu, and W. Wang. Centroidal power diagrams with capacity constraints : computation, applications, and extension. *ACM Transactions on Graphics (TOG)*, 35(6) :244, 2016.
- [139] A. Yezzi. Modified curvature motion for image smoothing and enhancement. *IEEE Transactions on Image Processing*, 7(3) :345–352, 1998.

TITLE : Projection in the 2-Wasserstein sense on structured measure space.

Abstract

This thesis focuses on the approximation for the 2-Wasserstein metric of probability measures by structured measures . The set of structured measures under consideration is made of consistent discretizations of measures carried by a smooth curve with a bounded speed and acceleration. We compare two different types of approximations of the curve : piecewise constant and piecewise linear. For each of these methods we develop fast and scalable algorithms to compute the 2-Wasserstein distance between a given measure and the structured measure. The optimization procedure reveals new theoretical and numerical challenges, it consists in two steps : first the computation of the 2-Wasserstein distance, second the optimization of the parameters of structure. This work is initially motivated by the design of trajectories in MRI acquisition, however we provide new applications of these methods.

KEYWORDS : Wasserstein distance, optimization, measure theory, sampling theory, blue noise, curve projection, quantization, optimal transport, power diagrams.

AUTEUR : Léo LEBRAT.

TITRE : Projection au sens de Wasserstein 2 sur des espaces structurés de mesures.

DIRECTEURS DE THÈSE : Frédéric de GOURNAY et Jonas KAHN.

LIEU ET DATE DE SOUTENANCE : Jeudi 19 Décembre, Amphi Schwartz 1R3 Institut de mathématiques de Toulouse, 118 Route de Narbonne, 31400 Toulouse, France.

Résumé

Cette thèse s'intéresse à l'approximation pour la métrique de 2-Wasserstein de mesures de probabilité par une mesure structurée. Les mesures structurées étudiées sont des discrétisations consistantes de mesures portées par des courbes continues à vitesse et à accélération bornées. Nous comparons deux types d'approximations pour ces courbes continues : l'approximation constante par morceaux et linéaire par morceaux. Pour chaque méthode, des algorithmes rapides et fonctionnant pour une discrétisation fine ont été développés. Le problème d'approximation se divise en deux étapes avec leurs propres défis théoriques et algorithmiques : le calcul de la distance de Wasserstein 2 et son optimisation par rapport aux paramètres de structure. Ce travail est initialement motivé par la génération de trajectoires d'IRM en acquisition compressée, toutefois nous donnons de nouvelles applications potentielles pour ces méthodes.

MOT-CLÉS : Distance de Wasserstein, optimisation, théorie de la mesure, théorie de l'échantillonnage, blue noise, projection de courbes, quantification, transport optimal, diagramme de Laguerre.

DISCIPLINE ADMINISTRATIVE : MITT : Domaine Mathématiques : Mathématiques appliquées.

INTITULÉ ET ADRESSE DU LABORATOIRE : Université Toulouse 3 Paul Sabatier. Institut de Mathématiques de Toulouse (UMR 5219) 118 Route de Narbonne, 31400 Toulouse.