



Calcul de diagrammes de Laguerre pour un problème de transport optimal en cosmologie.

Travail encadré par LÉVY Bruno, directeur de recherche
à l'INRIA de Nancy et présenté à l'École Normale Supérieure de Lyon
dans le cadre du stage d'initiation à la recherche de L3

HUMEAU Samuel

26/08/2018

Résumé

Le sujet de ce stage est la compréhension et l'expérimentation d'algorithmes destinés à résoudre un problème de transport optimal semi-discret, en lien avec des notions de géométrie algorithmique et avec comme toile de fond une application en cosmologie.

Ces algorithmes utilisent une méthode de Newton comme schéma de base. Ils nécessitent de plus le calcul d'un diagramme de Laguerre. Ce rapport introduit une méthode pour ce calcul passant par une structure d'arbre AABB (Axis-Aligned Bounding Box).

Introduction

Le problème du transport optimal fut considéré pour la première fois par Gaspard MONGE en 1781 dans [13]. Ce fut alors d'un point de vue très concret et de manière plus restrictive. Ce problème, comme l'indique son nom, possède une interprétation simple : optimiser un transport. Par exemple, considérons des mines de charbon et des usines, clientes des dites mines. Le transport du minerai des mines aux usines a un coût. Selon les positions relatives des différentes mines et usines, il doit exister des manières plus ou moins efficaces d'acheminer le combustible. Optimiser cela est une instance du transport optimal.

D'abord d'intérêt assez réduit, ce sujet fut ravivé par KANTOROVITCH dans divers articles dans les années 1940 ([9], [10]). Il généralisa la formulation de MONGE à partir de la théorie de la mesure et il en développa les applications en économie, ce qui lui valut le prix nobel d'économie en 1975. En parallèle et avec le développement des sciences informatiques, les algorithmes résolvant des problèmes de programmation linéaire fournirent des outils pour résoudre certains problèmes de transport optimal (la première fois dans [7]).

Le transport optimal connaît régulièrement des gains d'intérêts en lien avec d'autres disciplines, notamment en mécanique des fluides, en probabilité, ou encore en théorie des équations aux dérivées partielles (voir par exemple [14], [2]). Dans [6], BRENIER, FRISCH & *al* introduirent une application en cosmologie. Il semble qu'à partir d'une certaine échelle, le problème de la reconstruction de l'univers primordial puisse s'exprimer comme un problème de transport optimal que l'on pourrait résoudre numériquement.

Lors de la publication de [6], les performances des algorithmes disponibles limitaient fortement la taille des modèles numériques (de l'ordre de 100 000 points dans ce dernier article). Les algorithmes développés par Bruno LÉVY dans [11] en 2015 puis dans sa suite par KITAGAWA, MÉRIGOT et THIBERT dans [3] en 2016, proposent une solution à cette limite. Cela

fonctionne bien dans certains cas, comme en mécanique des fluides ([16]). Cependant, de par la structure particulière des données à traiter en astrophysique il est nécessaire d'adapter ces algorithmes.

C'est là qu'interviennent mes travaux. Une étape de l'algorithme présenté par KITAGAWA, MÉRIGOT & THIBERT perd de son efficacité. Le but final des recherches auxquelles j'ai pris part, est d'obtenir le gain de vitesse nécessaire pour réaliser des reconstructions de l'univers primordial aussi grandes que les simulations à n corps actuellement réalisées (de l'ordre de 500 milliards de points (voir le projet DEUSS¹)). Ce rapport de stage présente une possible avancée.

Dans une première partie, largement inspirée de [12], le transport optimal sera défini de manière plus rigoureuse. Diverses propriétés essentielles permettant la mise en place d'algorithmes seront étudiées. En particulier, le problème dual du transport optimal sera évoqué, ainsi que les notions de c -transformée et de c -superdifférentielle qui permettent respectivement de se placer dans un ensemble compact et d'établir un lien plus précis entre la solution du problème dual et du problème initial.

La partie suivante proposera une vue d'ensemble des algorithmes existant et des problèmes qu'ils posent dans le cas de l'application à l'astrophysique. La première méthode passe par un problème d'affectation et utilise des méthodes de programmation linéaire. C'est cela qui était utilisé dans l'article pionnier [6]. Son temps d'exécution ($O(n^3)$) le rend obsolète. Les méthodes suivantes ([11], [3]) utilisent le transport semi-discret et passe par un même schéma algorithmique : une méthode de Newton.

Les pistes que j'ai suivies seront exposées dans la dernière partie. Le schéma algorithmique utilisé passe par le calcul d'une structure géométrique appelée diagramme de Laguerre ([4]). C'est son calcul qui pose problème dans le cadre astrophysique. La méthode utilisée ici passe par une structure de recherche particulière : un AABB (pour Axis-Aligned Bounding Box) tree.

La partie physique du problème présente peu d'intérêt ici et ne sera que vaguement évoquée quand cela sera nécessaire. Le lecteur se référera à [6] pour plus de détails.

1. en particulier : www.deus-consortium.org/a-propos/

1 Un problème de transport optimal

1.1 Définition

Définissons un peu plus formellement ce qu'est le transport optimal. On s'intéresse à la formalisation dû à KANTOROVITCH, plus générale que celle de MONGE. Soit (X, T_X, μ) et (Y, T_Y, ν) deux espaces mesurables et $c : X \times Y \rightarrow \mathbb{R}_+$ une fonction mesurable sur $(X \times Y, T_X \times T_Y)$ appelée fonction de coût. On définit par Γ l'ensemble des mesures produit sur $(X \times Y, T_X \times T_Y)$ ayant μ et ν comme marginales, i.e., si $\gamma \in \Gamma$:

$$\forall B \subseteq Y, \gamma(X, B) = \nu(B) \quad (1)$$

$$\forall A \subseteq X, \gamma(A, Y) = \mu(A) \quad (2)$$

On remarque qu'il faut que $\nu(Y) = \gamma(X, Y) = \mu(X)$.

Définition

Résoudre le problème de transport optimal associé à ces objets correspond à déterminer le nombre suivant :

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

Dans les cas couverts par la formulation de MONGE, il existe une fonction $t : X \rightarrow Y$, telle que :

$$\inf_{\pi \in \Gamma} \int_{X \times Y} c(x, y) d\pi(x, y) = \inf_{T: X \rightarrow Y} \int_X c(x, T(x)) d\mu(x) = \int_X c(c, t(x)) d\mu(x)$$

t est appelée plan de transport. Dans notre cas, une réciproque à t existera toujours et c'est sa connaissance qui permettra la reconstruction de l'univers. On l'appellera plan de transport inverse.

Cette forme du problème se traduit très mal en algorithme. Bien qu'elle se comprenne facilement, il faut en changer. Souvent, les problèmes de borne inférieure admettent une formulation, dite duale, sous forme d'un problème de borne supérieure. C'est exactement ce que l'on va présenter maintenant et c'est cette forme qui donnera lieu à des algorithmes. D'un point de vue mathématique, cela vient du fait que la nouvelle forme nous place dans un espace plus "lisse" contenant un sous-ensemble compact au sein duquel on pourra converger vers la solution.

1.2 Le problème dual

Commençons par donner une idée intuitive de ce qu'est la forme duale. Dans notre exemple ci-dessus, on cherche à diminuer le plus possible le coût de transport de minerais de charbon des mines jusqu'aux usines. On peut considérer que le coût de l'extraction du minerai est pris en compte par c . Éventuellement, ce coût peut dépendre de la mine. Considérons le point de vue des usines. Elles engagent une société pour l'extraction du minerai et pour son transport jusqu'aux usines. On peut définir une fonction $\phi : X \rightarrow \mathbb{R}_+$ représentant le coût d'extraction en $x \in X$ et une autre fonction $\psi : Y \rightarrow \mathbb{R}_+$ représentant le coût d'acheminement en $y \in Y$. Ainsi, le coût d'extraction d'un minerai en $x \in X$ puis de son acheminement en $y \in Y$ est $\phi(x) + \psi(y)$. Puisque les usines ne veulent pas perdre d'argent dans le processus, elles imposent à la société que :

$$\phi(x) + \psi(y) \leq c(x, y)$$

Le coût total pour la société est alors :

$$\int_X \phi(x) d\mu(x) + \int_Y \psi(y) d\nu(y)$$

Mais l'entreprise cherche à maximiser son profit, elle veut donc chercher la valeur :

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

Trouver cette valeur est le problème dual du transport optimal (il faut supposer $(\phi, \psi) \in L_1$).

Un théorème dû à KANTOROVITCH donne alors :

Théorème 1.1 :

$$\sup_{\phi, \psi} \int_X \phi(x) d\mu(x) + \int_Y \psi(y) d\nu(y) = \inf_{\pi \in \Gamma} \int_{X \times Y} c(x, y) d\pi(x, y)$$

1.3 c -transformée et c -superdifférentielle

On veut maintenant se placer dans un sous-ensemble compact de fonctions contenant la solution. Pour cela, on a besoin de certaines notions :

Définition

- Soit $f : X \rightarrow \mathbb{R}$. On note $f^c(y) = \inf_{x \in X} [c(x, y) - f(x)]$, pour $y \in Y$. On parle de la c -transformée de f .
- Si ϕ une fonction est la c -transformée d'une fonction f , on dit que ϕ est c -concave.
- On note $\Psi_c(Y)$ l'ensemble des fonctions c -concaves sur Y .

Les propriétés suivantes nous assurent que s'il existe une solution, alors il en existe une de la forme (ϕ, ϕ^c) . On peut alors se placer dans l'ensemble des couples de cette forme.

Proposition:

- Si (ϕ, ψ) est un plan acceptable, alors (ϕ, ϕ^c) aussi.
- Dans ce cas (ϕ, ϕ^c) est une meilleur solution que (ϕ, ψ) .

On aimerait avoir accès à la solution initiale du problème du transport à partir de sa forme duale. Les notions introduites maintenant servent à cela.

Définition

Si ϕ est une fonction de Y dans X , on appelle c -superdifférentielle de ϕ et on note $\partial^c \phi$ l'ensemble :

$$\partial^c \phi = \{(x, y) \mid \forall z \in Y, \phi(z) \leq \phi(y) + (c(x, z) - c(x, y))\}$$

Le théorème suivant permet, dans le cas où un plan de transport existe, de le déterminer en fonction de la solution du problème dual.

Théorème 1.2: Si ϕ est une fonction c -concave, alors pour tout $(x, y) \in \partial^c \phi$,

$$\nabla \phi(y) - \nabla_y c(x, y) = 0$$

On s'intéresse au cas où le coût c est une distance quadratique, $c(x, y) = \frac{1}{2} \|x - y\|^2$. Alors :

$$\nabla_y c(x, y) = y - x$$

Donc, d'après le théorème précédent,

$$\nabla \phi(y) - (y - x) = 0$$

puis :

Corollaire 1: Dans le cas où c est une distance quadratique, il existe un plan de transport inverse (une fonction de Y dans X) solution donné par :

$$t(y) = y - \nabla\phi(y) = \nabla\left(\frac{\|y\|^2}{2} - \phi(y)\right)$$

Tout ce qui est nécessaire aux algorithmes a été présenté.

2 Différents algorithmes

2.1 Un premier algorithme pas très efficace

Dans le cas général, le problème du transport optimal fait intervenir deux ensembles continus. Cela pose problème pour des algorithmes et l'on est obligé d'effectuer une discretisation. On se retrouve avec un problème de transport avec deux ensembles discrets. L'intégrale définissant le problème devient une somme et on doit maintenant trouver la permutation qui permet d'optimiser cette somme. Il s'agit d'un problème d'assignement qui se résoud avec des techniques de programmation linéaire. La complexité est en $O(n^3)$, ce qui ne peut convenir dans notre cas.

2.2 Du transport optimal semi-discret

On peut remarquer que la reconstruction de l'univers part d'une disposition actuelle discrète : chaque point représente un amas de galaxies. La structure de l'univers à grande échelle est discrète. L'univers primordial était homogène et continu. On a donc un transport semi-discret et on peut poser $\mu = \sum_{i=0}^n \mu_i$. Cela donne lieu à un nouvel algorithme. Montrons pourquoi.

On va chercher, conformément aux prémisses mathématiques, une solution de la forme (ψ, ψ^c) , où $\psi = \sum_{i=0}^n \psi_i$ est une somme de coefficients de Dirac.

Le problème dual revient alors à déterminer :

$$\inf_{\psi \in \Psi^c(Y)} \left[\sum_{i=0}^n \psi_i \mu_i + \int_Y \psi^c(y) d\nu(y) \right]$$

Le but est de montrer que cela ne dépend que des n chiffres définissant ψ :

$$F(\psi) \stackrel{def}{=} \sum_{i=0}^n \psi_i \mu_i + \int_Y \psi^c(y) d\nu(y) \quad (3)$$

$$= \sum_{i=0}^n \psi_i \mu_i + \int_Y \inf_{x_i \in X} (c(x_i, y) - \psi(y)) d\nu(y) \quad (4)$$

$$= \sum_{i=0}^n \psi_i \mu_i + \sum_{i=0}^n \int_{Lag_{\psi}^c(x_i)} (c(x_i, y) - \psi(y)) d\nu(y) \quad (5)$$

où :

$$Lag_{\psi}^c(x_i) = \{y \in Y, \forall k \neq i, c(x_i, y) - \psi_i \leq c(x_k, y) - \psi_k\}$$

est le diagramme de Laguerre associé au i -ième point. Il s'agit d'une structure géométrique que l'on détaillera dans la section suivante.

On a alors le théorème suivant :

Théorème 2.1: La fonctionnelle F est concave.

Il suffit de maximiser F pour obtenir une valeur correcte pour ψ . On voudrait utiliser une méthode de Newton pour cela. Cela demande d'accéder à ∇F et à $\nabla^2 F$. Regardons leur valeur.

Proposition:

$$\frac{\partial F}{\partial \psi_j} = \mu_j - \int_{Lag_{\psi}^c(x_j)} \nu(x) dx$$

Proposition:

$$\frac{\partial^2 F}{\partial \psi_j \partial \psi_i}(\psi) = \int_{Lag_{\psi}^c(x_j) \cap Lag_{\psi}^c(x_i)} \frac{\mu(y)}{\|x_i - x_j\|} dS(y) \quad (i \neq j)$$

$$\frac{\partial^2 F}{\partial^2 \psi_j}(\psi) = - \sum_{i \neq j} \frac{\partial^2 F}{\partial \psi_i \partial \psi_j}(\psi)$$

Elles dépendent du diagramme de LAGUERRE. Si on accède à ce diagramme, on pourra réaliser cet algorithme.

2.3 L'algorithme final

Data: Les points initiaux $(x_i)_{0 \leq i \leq n}$
 La densité initiale $\mu = \sum_{i=0}^n \mu_i$
 La densité finale ν

Result: L'unique diagramme de Laguerre tel que :

$$\forall i, \mu_i - \int_{Lag_{\psi}^c(x_i)} \nu(x) dx = \frac{\partial F}{\partial \psi_i} = 0$$

$\psi \leftarrow (0, \dots, 0)$

while ψ n'a pas convergé **do**

 Calculer ∇F et $\nabla^2 F$

 Trouver le vecteur $p \in \mathbb{R}^n$ tel que $\nabla^2 F(\psi)p = -\nabla F(\psi)$

 Trouver un paramètre α convenant.

 Faire $\psi \leftarrow \psi + \alpha p$

end

Algorithm 1: L'algorithme de Newton

Pour quelle raison renvoie-t-on le diagramme de Laguerre et non la fonction ψ ? Il a été remarqué que dans le cas d'une distance euclidienne, il existe un plan de transport inverse donné par :

$$t(y) = y - \nabla \psi^c(y) = y - \nabla \left(\inf_{0 \leq i \leq n} (c(x_i, y) - \psi(y)) \right)$$

En supposant que la borne inférieure soit atteinte pour $j \in \{0, \dots, n\}$,

$$t(y) = y - (y - x_j) = x_j$$

Ainsi,

$$t^{-1}(x_i) = \{y | \forall j \neq i, c(x_i, y) - \psi(x) \leq c(x_j, y) - \psi(y)\} = Lag_{\psi}^c(x_i)$$

Le diagramme de Laguerre permet bien de reconstituer le transport.

Cet algorithme suppose plusieurs choses.

Il faut savoir quand arrêter la boucle *while*, i.e., savoir quand la convergence est suffisante. On converge quand toutes les composantes du gradient de F sont nulles. Le critère utilisé passe par la définition de ∇F comme différence de deux quantités : la convergence est considérée atteinte quand

$$\forall j \in \{0, \dots, n\}, |\nabla F_j| < \epsilon \min_j \mu_j,$$

où ϵ est un pourcentage pris petit (de l'ordre de 1%).

Il faut aussi pouvoir trouver α . KITAGAWA et MÉRIGOT fournissent un critère pour cela dans [3]. Il faut que la plus petite des cellules de Laguerre soit “suffisamment” grande.

Pour finir, un petit mot sur les calculs des gradients, hessiennes et autres vecteurs p . Les deux premiers utilisent le diagramme de Laguerre et des algorithmes de géométrie. La détermination de p passe par la résolution d’un système linéaire. La méthode utilisée est celle du gradient conjugué ([8] et [1] pour une bonne introduction du sujet).

3 Un nouvel algorithme

L’essentiel des outils utilisés pour implémenter la méthode de Newton fonctionne. Ce qui pose problème est le calcul des diagrammes de LAGUERRE. Commençons par détailler un peu ce qu’ils sont.

3.1 Les diagrammes de Laguerre

3.1.1 Définition

Un diagrammes de Laguerre est une structure qui, étant donné n points $(x_i)_{1 \leq i \leq n}$ et des poids $(\omega_i)_{1 \leq i \leq n}$ associés à ces points, définit une partition de l’espace en polyèdres. À chaque polyèdre, appelé cellule de Laguerre, correspond exactement un des n points, appelé racine de la cellule. Certaines cellules peuvent être vides et les racines ne sont pas nécessairement dans leur cellule.

Définition

Étant donné la définition précédente, la cellule de Laguerre associée à la i -ième cellule est donnée par l’ensemble

$$\{x | \forall j \in \{0, \dots, n\}, d(x, x_i) - \omega_i \leq d(x, x_j) - \omega_j\}$$

Il s’agit donc de l’ensemble des points de l’espace plus proche de x_i que de tous les autre points x_j en tenant compte des poids (ω_i) .

L’espace dans lequel on se place peut être de dimension quelconque. Dans le cas de notre application en astrophysique, il est de dimension 3. Il s’agit de l’espace usuel.

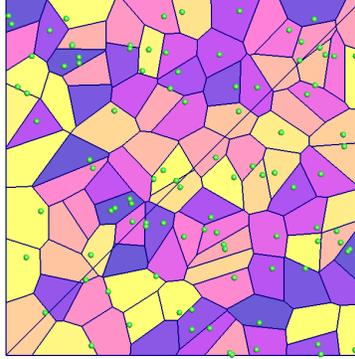


FIGURE 1 – Un exemple de diagramme de laguerre en dimension 2.

La structure n'est pas simple à manipuler d'un point de vue algorithmique. On se ramène à une autre structure géométrique plus simple : un diagramme de Voronoï.

3.1.2 Diagrammes de Voronoï

Les diagrammes de Voronoï sont un cas particulier des diagrammes de Laguerre pour lequel tous les poids ω_i sont nuls. Les cellules sont alors exactement les points de l'espace qui sont les plus proches de la racine.

Un diagramme de Laguerre en dimension m est en fait un diagramme de Voronoï en dimension $m + 1$ que l'on intersecte avec un sous-espace de dimension m . En effet, plaçons-nous dans un espace de dimensions $m + 1$. On considère les points x'_i donnés par les points x_i auquel on rajouterait la coordonnée $\sqrt{\omega_{max} - \omega_i}$. Le diagramme de Voronoï associé à cet ensemble de points est donné par l'ensemble des cellules

$$\{x | \forall j \neq i, d(x, x'_i) \leq d(x, x'_j)\}$$

La distance que l'on utilise n'est pas la racine carrée de la somme des carrés des coefficients des points, mais plutôt la somme des carrés des coefficients. Cela assure une meilleure précision et c'est ce qui permet de transformer un diagramme de Laguerre en un diagramme de Voronoï. Alors, si x a son m -ième coefficient nul et en confondant x dans \mathbb{R}^m et dans \mathbb{R}^{m+1} :

$$d(x, x'_i) = d(x, x_i) + \sqrt{\omega_{max} - \omega_i}^2 = d(x, x_i) + \omega_{max} - \omega_i$$

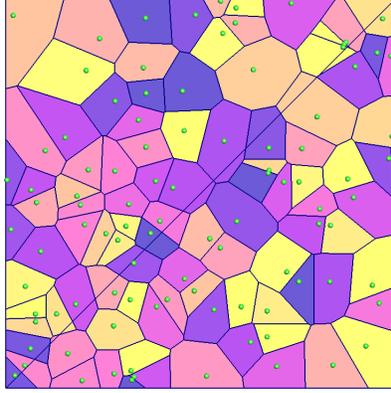


FIGURE 2 – Un exemple de diagramme de Voronoï en dimension 2. Remarquez que les arrêtes sont les bissectrices des racines.

D'où,

$$x \in Cell_{Lag}(x_i) \Leftrightarrow \forall j \neq i, d(x, x_i) - \omega_i \leq d(x, x_j) - \omega_j \quad (6)$$

$$\Leftrightarrow \forall j \neq i, d(x, x_i) + (\omega_{max} - \omega_i) \leq d(x, x_j) + (\omega_{max} - \omega_j) \quad (7)$$

$$\Leftrightarrow \forall j \neq i, d(x, x'_i) \leq d(x, x'_j) \quad (8)$$

$$\Leftrightarrow x \in Cell_{Voro}(x'_i) \quad (9)$$

On se simplifie encore la tâche en utilisant la structure duale du diagramme de Voronoï : la triangulation de Delaunay. Chaque sommet du diagramme est représenté par un m -simplex (en dimension 2 des triangle, en dimension 3 des tétraèdres, etc), les racines (et donc en quelque sorte les polyèdre) devenant les sommets des simplex et les arrêtes restant des arrêtes. Considérons un triangle de la triangulation. Le centre du cercle circonscrit au triangle est équidistant de chaque sommet du triangle. Dans le diagramme de Voronoï il s'agit donc du sommet d'un polyèdre. Aucun autre point ne peut se trouver à l'intérieur du cercle circonscrit. Sinon, son centre serait plus proche de ce point que des sommets du triangle, contredisant le fait que le centre soit le sommet d'un polyèdre dans le diagramme de Voronoï. Réciproquement, une triangulation telle que tout point ne soit dans le cercle circonscrit d'aucun triangle est une triangulation de Deulaunay.

Le fait de n'avoir plus que des m -simplices rend le stockage en mémoire beaucoup plus simple.

3.2 Deux algorithmes existants

3.2.1 Bowyer Watson

Il existe pour cela plusieurs méthodes. La plus utilisée est celle utilisant l'algorithme de BOWYER-WATSON ([5], [15]). Essentiellement, il s'agit d'ajouter un à un les points considérés et de reconstruire seulement la partie de la triangulation de Deulaunay qui est détruite par l'insertion.

On commence par définir un triangle englobant tous les points afin d'initialiser la triangulation. Puis, on insère chaque point, un à un. Lorsqu'on insère un point x , on parcourt les triangles existants et on supprime ceux contenant x . Les arêtes de ces triangles forment un polyèdre englobant. Chaque sommet de ce polyèdre est utilisé pour construire de nouveaux triangles avec x .

La triangulation est valide à chaque étape, ce qui garantit bien ce que l'on veut.

Cet algorithme a l'avantage d'être assez efficace. En dehors de quelques cas dégénérés pour lesquels le temps d'exécution est en $O(n^2)$ il s'exécute en $O(n \log(n))$. En revanche, il nécessite de travailler sur la structure globale de la triangulation, limitant fortement les possibilités de parallélisation et nécessitant beaucoup de mémoire vive.

Puisque l'on voudrait l'utiliser sur 500 milliards de points, il ne peut convenir.

3.2.2 Nivoliers

Une deuxième méthode est locale : chaque cellule est calculée en parallèle. Le principe est le suivant. Une cellule de Voronoi est donnée par :

$$Cell_{Voro}(x_i) = \{y | \forall j \neq i, d(y, x_i) \leq d(y, x_j)\} \quad (10)$$

$$= \bigcap_j \{y | d(y, x_i) \leq d(y, x_j)\} \quad (11)$$

Une cellule de Voronoi est donc l'intersection de tous les demi-espaces contenant x_i et délimités par les bissecteurs entre x_i et les autres points. Un bissecteur est l'hyperplan qui est à équidistance entre deux points.

Une cellule de Voronoi peut être calculée par intersection successive avec les différents demi-espaces. Pour une cellule de Laguerre il suffit de préalablement intersecter avec \mathbb{R}^m . Bien sûr, on ne veut pas tout intersecter pour des questions de temps d'exécution. C'est pour cela que l'on introduit les techniques suivantes.

Nos points sont en nombre fini. Ils sont contenus dans une boîte englobante. On peut initialiser la cellule par cette boîte (en pratique, cette boîte est normalisée).

On peut se demander quels sont les points définissant les demi-espaces les plus intéressants. Considérons une racine x ainsi que sa cellule C en cours de calcul. Un demi-espace intéressant doit ronger la zone de C . Il faut donc que l'un des points strictement interne à la cellule soit strictement plus proche d'une autre racine y que de x . Mais si l'on ronge C , il faut que l'un des coins de C lui soit retiré. En effet, un demi-espace doit séparer la cellule en deux. Il faut qu'il y ait des coins de chaque côté pour qu'il coupe effectivement la cellule. Les demi-plans intéressants sont donc ceux qui retirent des coins à la cellule. Il faut chercher les racines les plus proches des coins pour être efficace.

Deux méthodes sont principalement utilisées. Elles furent introduites dans [17]. La première calcule le barycentre de la cellule, puis cherche les racines qui en sont le plus proches. L'avantage est de n'avoir qu'une requête de recherche à effectuer. L'inconvénient est qu'on ne sait pas quand s'arrêter. La seconde parcourt l'ensemble des sommets de la cellule et vérifie s'ils sont présents dans la cellule finale, i.e., vérifie si la racine est bien le point le plus proche du sommet. Cette méthode permet de valider les coins au fur et à mesure du calcul de la cellule. Quand ils sont tous validés, c'est que le calcul est fini. En revanche, il est nécessaire de faire une requête de recherche de plus proche voisin par sommet.

Ces deux méthodes ne sont pas égales. J'ai pu, sur quelques exemples vérifier que la validation des coins est toujours plus efficace.

En réalité on commence par initialiser les cellules en les intersectant avec les 100 points les plus proches de la racine. En effet l'efficacité de cette méthode repose sur le fait d'avoir déjà une bonne approximation de la cellule.

Les requêtes de plus proches voisins utilisent une structure de kd-tree. Tant que les points sont assez bien répartis (c'est-à-dire quand il y a peu de mouvement dans le transport), c'est efficace. Seulement, en cosmologie, les points se regroupent en structure assez lisse, courbes, surfaces et laissent de gros trous dans l'espace. En conséquence, quand on recherche un plus proche voisin, la structure de kd-tree n'aide pas à rejeter rapidement un grand nombre de points. Dans les dernières étapes de Newton, le calcul du diagramme devient quadratique en temps. Cet algorithme ne convient pas. En revanche, il résout le problème de mémoire dû à la méthode globale.

3.3 Une éventuelle solution

La structure de recherche précédente, i.e. le kd-tree ne convenait pas. On aimerait en changer. J'ai donc essayé une autre méthode. Mon but n'était pas de l'optimiser, mais seulement de faire des statistiques pour en voir le comportement dans un premier temps.

Le principe est le suivant. On commence par initialiser les cellules avec quelques plus proches voisins. L'expérience nous dit que cette étape ne coûte pas trop cher et qu'elle fournit une assez bonne approximation des cellules. Maintenant, on ne retient des cellules qu'une structure englobante, ce qui permet de récupérer les informations les plus importantes sans pour autant utiliser trop de mémoire. Dans mon cas il s'agissait d'AABB (Axis-Aligned Bounding Box), i.e. de boîtes englobantes dont les arrêtes sont alignées sur les axes. On aurait pu utiliser quelque chose de plus précis, comme des ellipsoïdes.

Maintenant, considérons deux boîtes englobante A et B de racine a et b . Si A et B ne s'intersectent pas, alors les cellules finales de a et de b ne peuvent avoir de frontière commune. Ainsi, intersecter les cellules de a et b par leur bissecteur est inutile.

On calcule l'ensemble des boîtes qui s'intersectent². D'après ce qui précède, on obtient les cellules qu'il faut intersecter et cela permet de finir le calcul des cellules.

Le calcul des intersections se fait à l'aide d'une structure d'arbre AABB. Il s'agit de l'espace qui est divisé en différentes boîtes triées selon l'ordre de Morton. La structure d'arbre permet de rapidement mettre de côté les ensembles qui ne s'intersectent pas.

Pour que cette méthode puisse être efficace il faut au moins une condition : que le nombre d'intersections détecté soit raisonnable. C'est le principal point auquel je me suis intéressé.

3.4 Quelques résultats

Avant même d'implémenter la méthode, j'ai essayé de m'assurer de sa pertinence. Je me suis intéressé au rapport du volume des boîtes englobantes par celui des cellules. Pour la grande majorité des cas, ce rapport est situé entre 1 et 2. Seuls quelques cas pathologiques avaient des rapports très grands. Tous les graphes que j'ai obtenus étaient semblable au suivant :

2. voir https://doc.cgal.org/latest/AABB_tree/index.html pour plus de détails

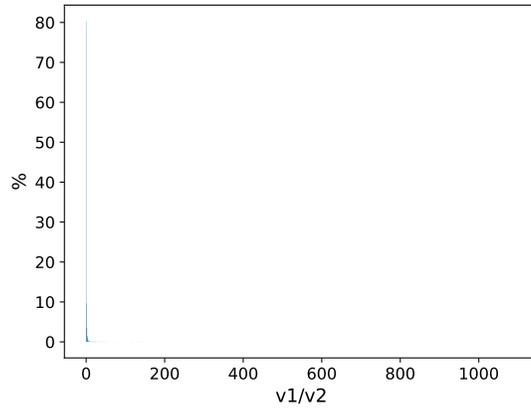


FIGURE 3 – En abscisse le rapport du volume des boites englobantes sur celui des cellules. En ordonnées le pourcentage de cellules correspondantes.

Le tableau suivant fournit quelques résultats sur le nombre de boites s’intersectant pour différents nombres de points et pour différentes structures de données.

Le nombre d’intersections est linéaire en fonction de la taille des données dans le cas des données uniformes. En revanche, pour de grand nombres de points issus de données cosmologiques, ce même nombre d’intersection est quadratique. L’hypothèse logique est que les données cosmologiques présentent des structures qui ne sont visibles qu’avec un grand nombre de points. Ainsi, l’initialisation avec les quelques points les plus proches ne fournit peut-

Nombre de points n	Données cosmologiques : nombre de boites s’intersectant divisé par n^2	données uniformes : nombre de boites s’intersectant divisé par n
1000	0,061958	25,194
10000	0,00617350	28,5904
50000	0,00283	27,72016
75000	0,00263	29,27872
100000	0,00262	27,91178
200000	0,00246	28,66027

FIGURE 4 – Quelques chiffres en rapport avec le nombre de boites s’intersectant trouver par l’algorithme

être pas une bonne approximation de la cellule.

Si ce nouvel algorithme ne semble pas résoudre nos problèmes, il pourrait néanmoins être utile dans d'autres domaines. L'étape suivante serait de l'exécuter entièrement, mais cela aurait nécessité la réécriture de tout le code manipulant les cellules en mémoire. Par manque de temps je n'ai pu le faire, et je me suis contenté de la partie sur les arbres AABB.

Conclusion

Le problème du transport optimal est un problème mathématique intéressant dont on ne cesse de découvrir de nouvelles applications. Dans le cas du transport semi-discret, on dispose de bons outils numériques. Seulement, face à la taille des données à traiter en cosmologie, il est nécessaire de développer de nouveaux algorithmes. Des choses ont déjà été tentées, notamment un algorithme local pour le calcul des diagrammes de Laguerre, qui, s'il était prometteur, échoue en temps de calcul. Un autre algorithme basé sur un arbre AABB semblait intéressant, mais après expérimentation, il semble avoir un temps quadratique sur des données cosmologiques, ce qui n'est pas acceptable dans ce cadre. Il se pourrait néanmoins que cela vienne de l'initialisation des cellules qui est mauvaise. Pour avoir de vrais résultats comparables aux autres méthodes, il faudrait implémenter une version fonctionnelle et optimiser.

D'autres problèmes se posent par rapport à cette application en cosmologie. Il faudrait que l'algorithme utilise des conditions aux limites périodiques. De plus, le but est l'exécution sur un cluster, ce qui apporte à nouveau des contraintes.

Références

- [1] Nicolas RAY Bruno LÉVY and Dimitry SOKOLOV. Géométrie numérique. in *Informatique Mathématique Une photographie en 2018*, 2018.
- [2] Yann BRENIER Jean-David BENAMOU. The monge-kantorovitch mass transfer and its computational fluid mechanics formulation. *International journal for Numerical Methods in Fluids*, 2000.
- [3] Quentin MÉRIGOT Jun KITAGAWA and Boris THIBERT. A newton algorithm for semi-discrete optimal transport. *CoRR*, 2016. abs/1603.05579.

- [4] Franz AURENHAMMER. Power diagrams : Properties, algorithms and applications. *SIAM J. Comput.*, 16(1) :78–96, 1987.
- [5] A. BOWYER. Computing dirichlet tessellations. *The computer journal*, 1981.
- [6] Y. BRENNIER, U. FRISCH, M. HENON, G. LOEPER, S. MATARRESE, R. MOHAYAEI, AND A. SOBOLEVSKII. Reconstruction of the early universe as a convex optimization problem. *arXiv*, 2003. arXiv :astro-ph/0304214v3.
- [7] George B. DANTZIG. Programming of interdependant activities : Ii mathematical model. *Econometrica*, 17(3/4) :200–211, 1949.
- [8] Magnus R. HESTENES and Eduard STIEFEL. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 1952.
- [9] L. KANTOROVITCH. On the translocation of masses. *Management Science*, 1942.
- [10] L. KANTOROVITCH. On a problem of monge. 1948.
- [11] Bruno LÉVY. A numerical algorithm for l_2 semi-discrete optimal transport in 3d. *ESAIM M2AN (Mathematical Modeling and Analysis)*, 2015.
- [12] Bruno LÉVY et Erica SCHWINDT. Notions of optimal transport theory and how to implement them on a computer. *arXiv*, 2017.
- [13] Gaspard MONGE. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie des Sciences*, 1784 :666–604, 1781.
- [14] Cedric VILLANI. *Optimal transport : Old and new*. Springer, 2009.
- [15] D. F. WATSON. Computing the n -dimensional delaunay tessellation with application to voronoi polytopes. *The computer journal*, 1981.
- [16] Quentin MÉRIGOT Thomas GALLOUËT. A lagrangian scheme à la brenier for the incompressible euler equations. *Foundations of Computational Mathematics*, 2018.
- [17] Christophe GEUZAIN Vincent NIVOLIER. Calcul d'un daigramme de voronoï restreint en dimension quelconque. 2014.

Annexe : contexte du stage

Ce stage a été fait à l'INRIA (Institut National de Recherche en Informatique et en Automatique) de Nancy, au sein de l'équipe Alice. L'équipe, en fin de vie, était en train de se diviser en deux. J'ai pu rencontrer tous ses membres, mais j'ai surtout cotoyé les chercheurs qui restent pour le moment dans l'équipe ; en particulier Bruno Lévy, Nicolas Ray et Dimitry Sokolov. J'ai partagé un bureau avec les deux derniers et le premier est mon maître de stage. Dimitry et Nicolas m'ont beaucoup appris sur la vie des chercheurs. Leur sujet d'étude était assez proche du mien. J'ai pu avoir de nombreuses conversations avec eux afin de confronter ce que je comprenais ou non. Ils m'ont beaucoup aidé.

J'ai eu l'occasion d'aller à Paris avec mon maître de stage le temps d'une journée. Le but était d'exécuter une version de l'algorithme sur une partie restreinte des données. Cela n'a pas été fructueux, mais l'expérience a été intéressante pour moi. D'autres chercheurs, notamment Quentin Mérigot, sont venus à l'INRIA pour discuter avec Bruno Lévy des perspectives futures, en rapport avec mon sujet d'étude.