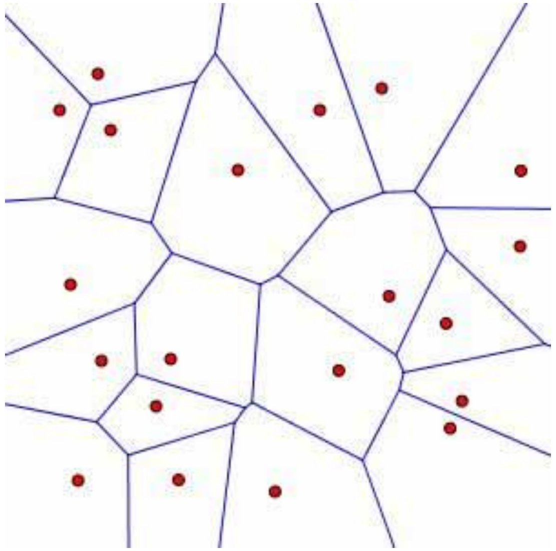
Semi-discrete optimal transport

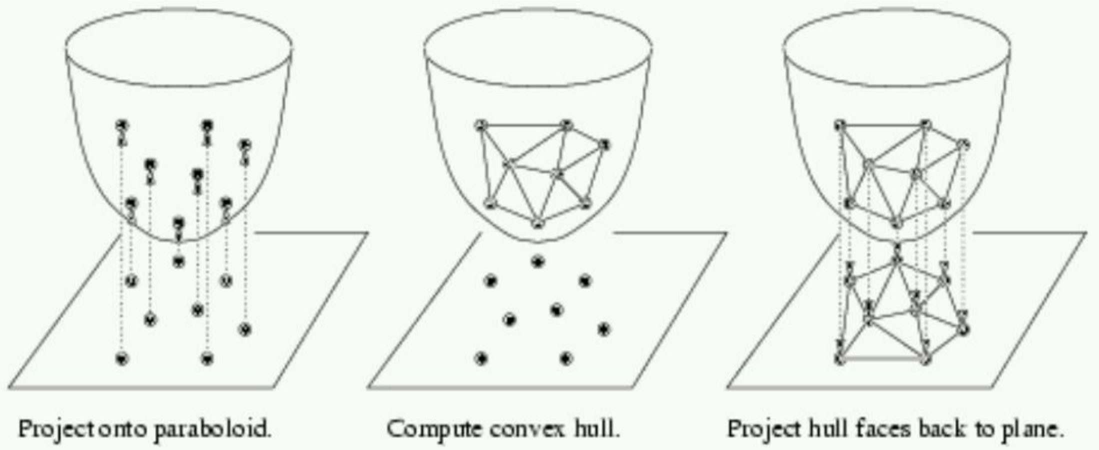
Voronoi diagram

- A partition such that each point x is assigned to its closest site x_i

$$\|x - x_i\|^2 \leq \|x - x_j\|^2 \quad \forall j$$

- The dual of a Delaunay triangulation: a triangulation of the sites such that no other site is encompassed by the circumcircle of a triangle
  - Also: convex hull of a parabolic lifting





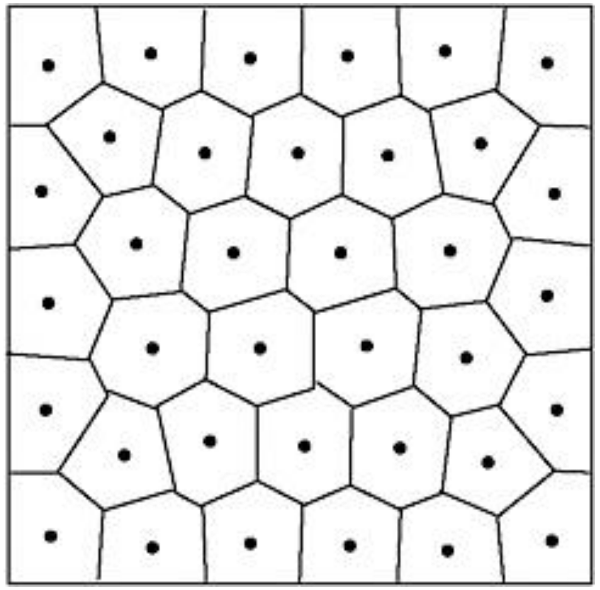
Centroidal Voronoi Diagram

- Can be defined as the solution to a least-square problem

$$\min \int_{Vor_i} \sum_i \|x - x_i\|^2 dx$$

  Also says that the centroid of Vor_i is the site x_i

- Can be computed by:
  - A Lloyd clustering algorithm
  - A descent approach on the above energy

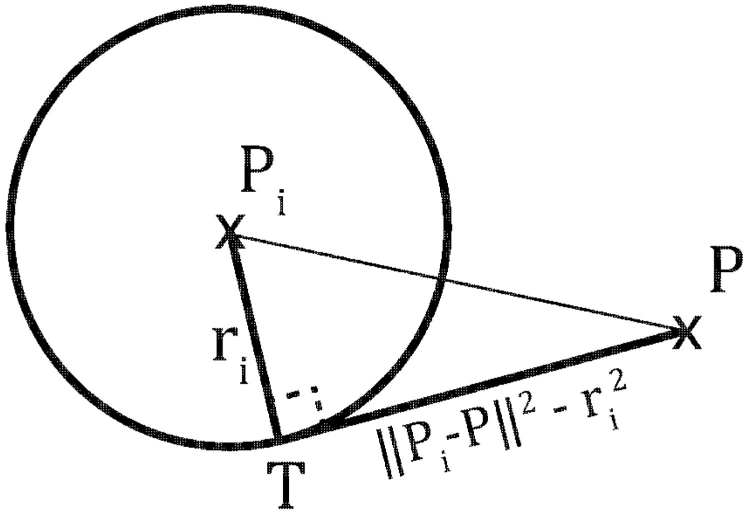


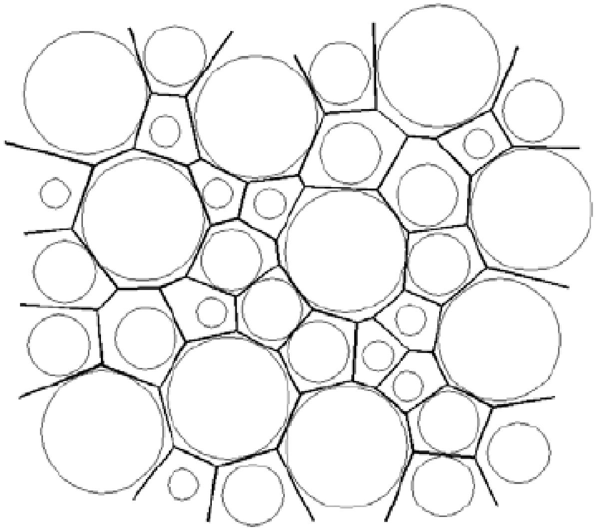
Power Diagram (Laguerre diagram)

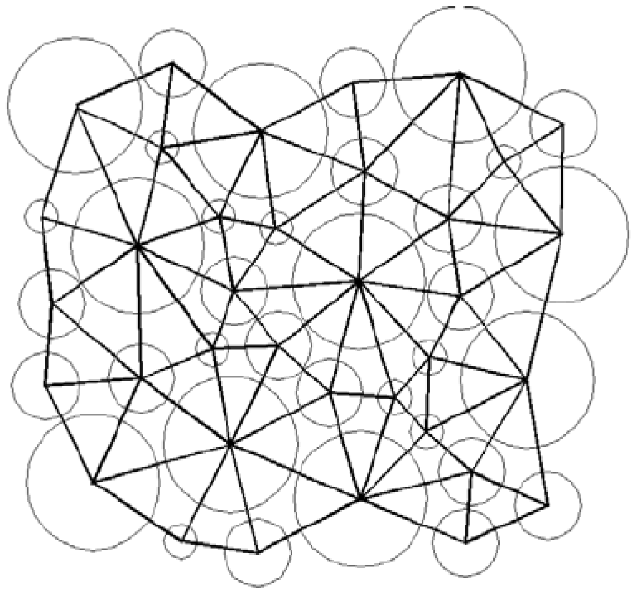
- A partition s.t. each point x is assigned to its closest site x_i with weight r_i

$$\|x - x_i\|^2 - w_i \leq \|x - x_j\|^2 - w_j \quad \forall j$$

Replaces distance to closest site with distance to closest tangential point to a circle of radius $r_i = \sqrt{w_i}$



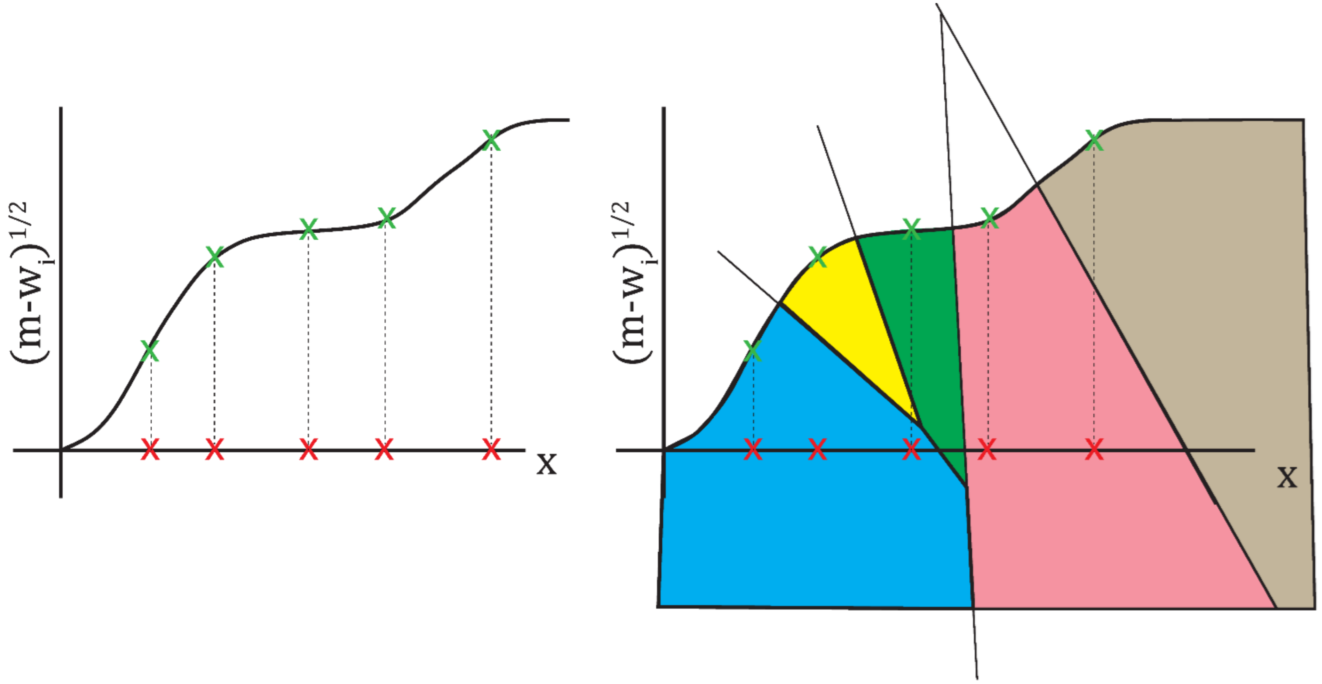




Any partition into convex polyhedral cells is a power diagram of some sites

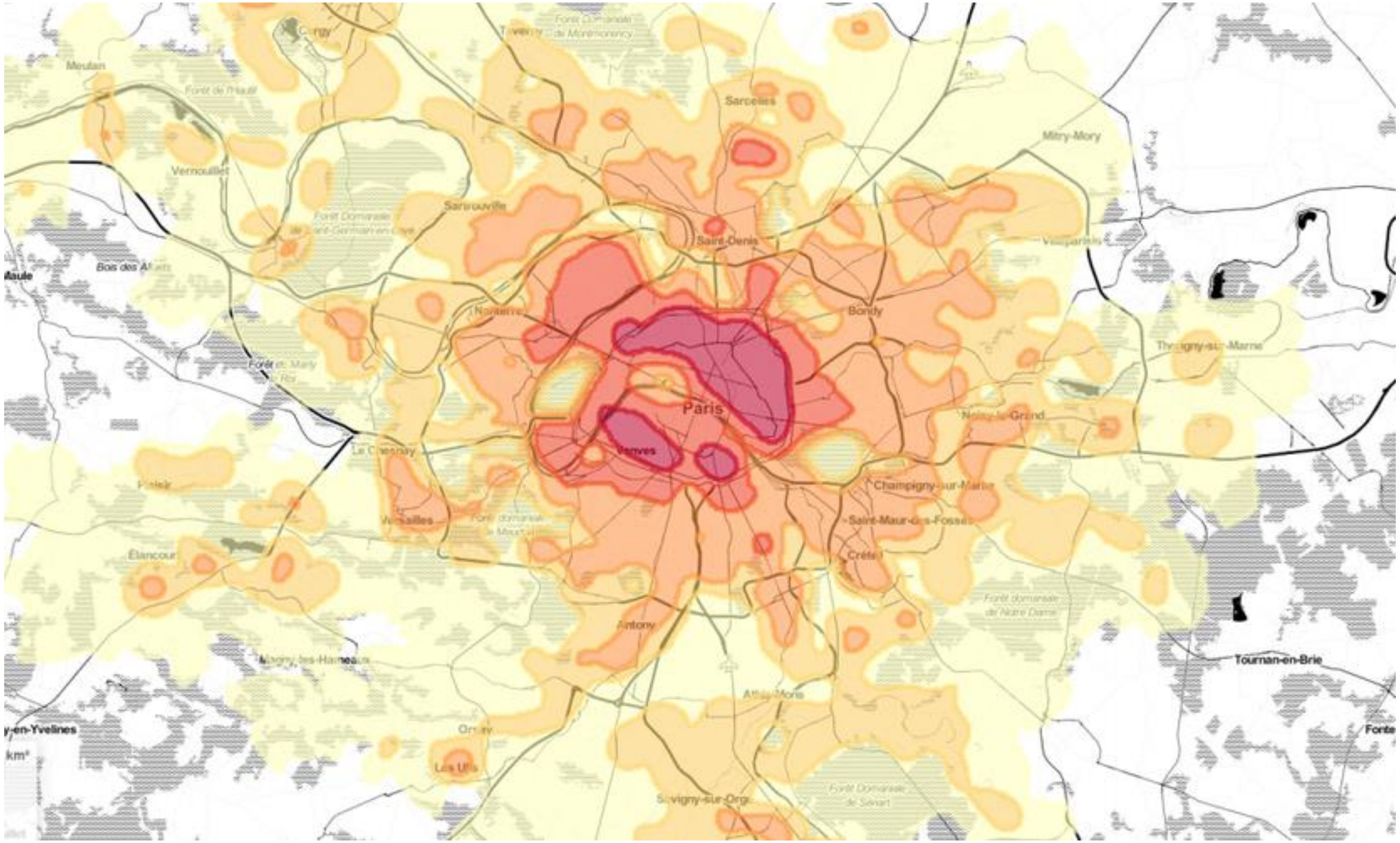
Power Diagram (Laguerre diagram)

- Can be computed by lifting a Voronoi diagram
  - Consider site coordinates $x_i' = (\, x_i \,;\, \sqrt{m - w_i})$ for large constant m ; $x' = (x\,;0)$
  - Then $\|x' - x_i'\|^2 \leq \left\|x' - x_j'\right\|^2 \;\; \forall j$



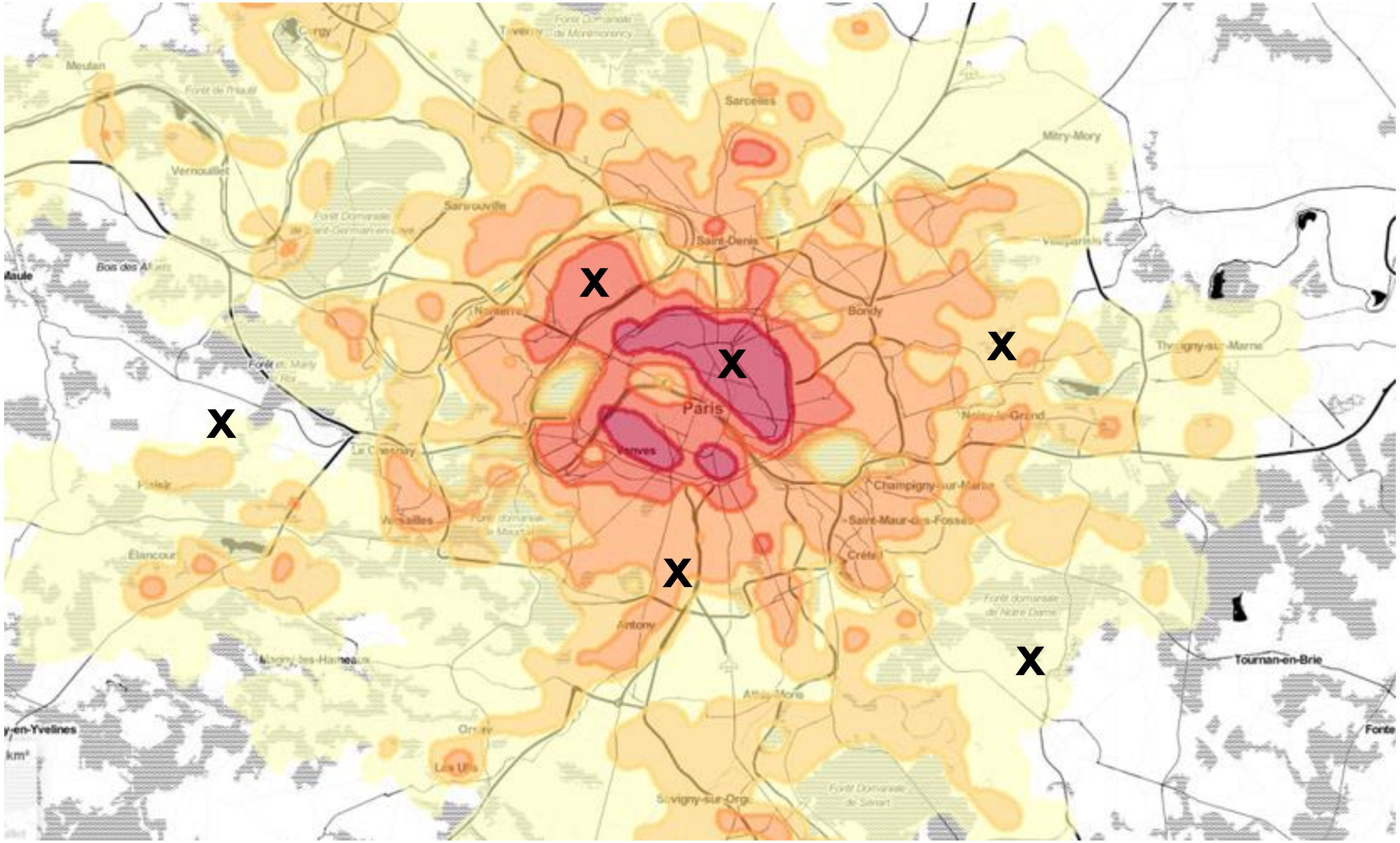
Some cells can be empty (e.g., yellow and green) and some sites can be outside of their cell (e

Semi-discrete Optimal Transport



Population density f

Semi-discrete Optimal Transport



Set of bakeries, factories, …?

Semi-discrete Optimal Transport

No constraint on production: population go to their nearest bakery/factory/… regardless of populat

Semi-discrete Optimal Transport

Limited production: population go to the nearest bakery/factory **with sufficient production**!

Semi-discrete Optimal Transport

(needs for) population here

=

quantity produced here

Limited production: population go to the nearest bakery/factory **with sufficient production**!

Back to optimal transport

- Optimal transport (Monge version) :

$$\min \int \|x - T(x)\|^2 \, d\mu(x)$$

Considering μ is continuous with density ρ

$$\min \int \|x - T(x)\|^2 \, \rho(x)dx$$

Considering ν (the target measure) discrete: $\nu = \sum \lambda_p \delta_p$

The mass preservation constraint is:

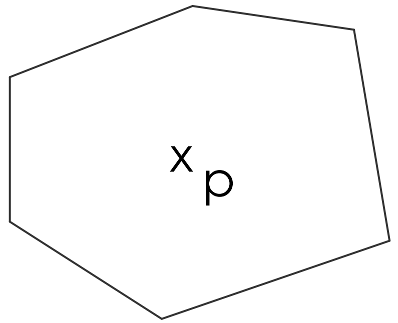
$$\lambda_p = \int_{T^{-1}(\{p\})} \rho(x)dx$$

A Multiscale Approach to Optimal Transport [Mérigot 2011]
Minkowski-Type Theorems and Least-Squares Clustering [Aurenhammer et al. 98]

Back to optimal transport

- In this case : $T^{-1}(\{p\}) = Vor^W(p)$

a power cell for some weight w_p



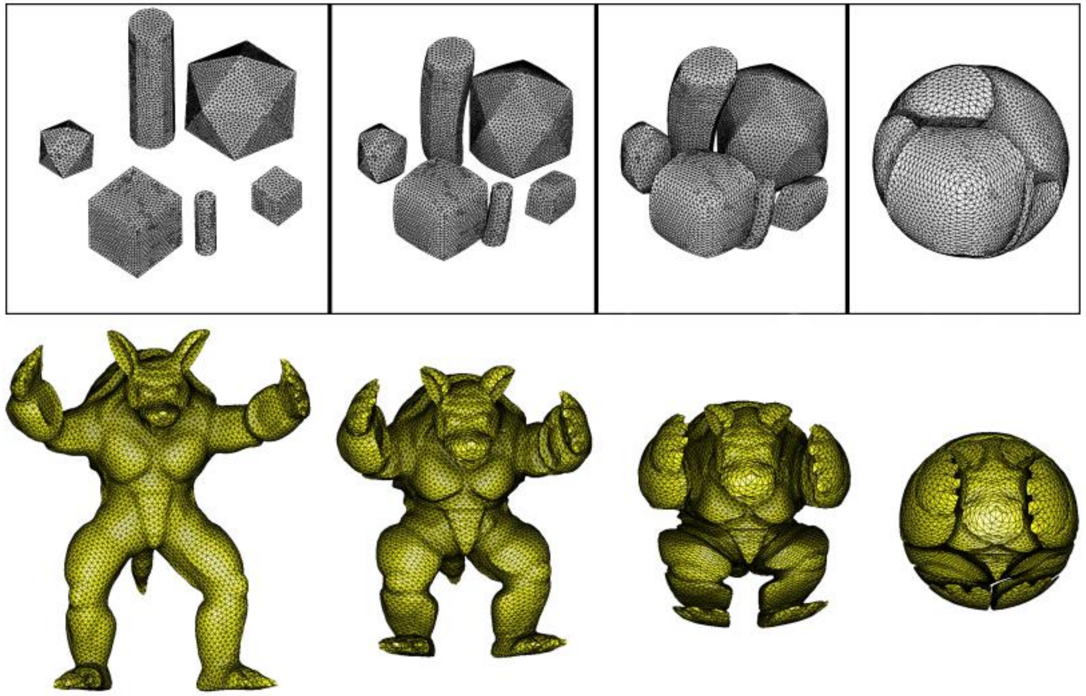
- This determines a partition, so Monge problem is:

$$\min \sum_p \int_{Vor^W(p)} ||x-p||^2 \, \rho(x) \, dx$$

- Idea: optimize weights w for each site to grow/shrink power cells until $\lambda_p = \int_{T^{-1}(\{p\})} \rho(x) dx$
- Gradient of appropriate functional given by $\frac{\partial \phi}{\partial w(p)}(w) = \lambda_p - \int_{Vor^W(p)} \rho(x) \, dx$

Back to optimal transport

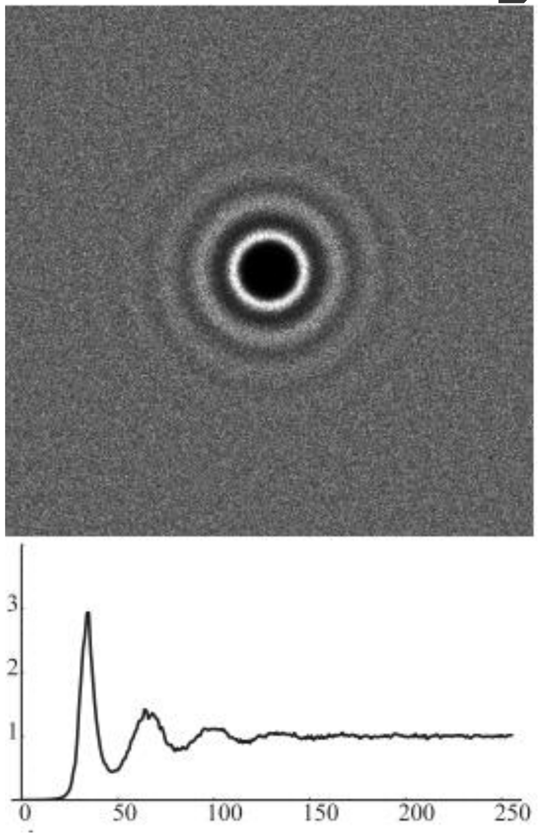
A Multiscale Approach to Optimal Transport [Mérigot 2011]



A Numerical Algorithm for L2 Semi-discrete Optimal Transport in 3D [Lévy 2015]

Application

- Also optimizes for the locations p



Blue Noise through Optimal Transport [de Goes et al. 2012]

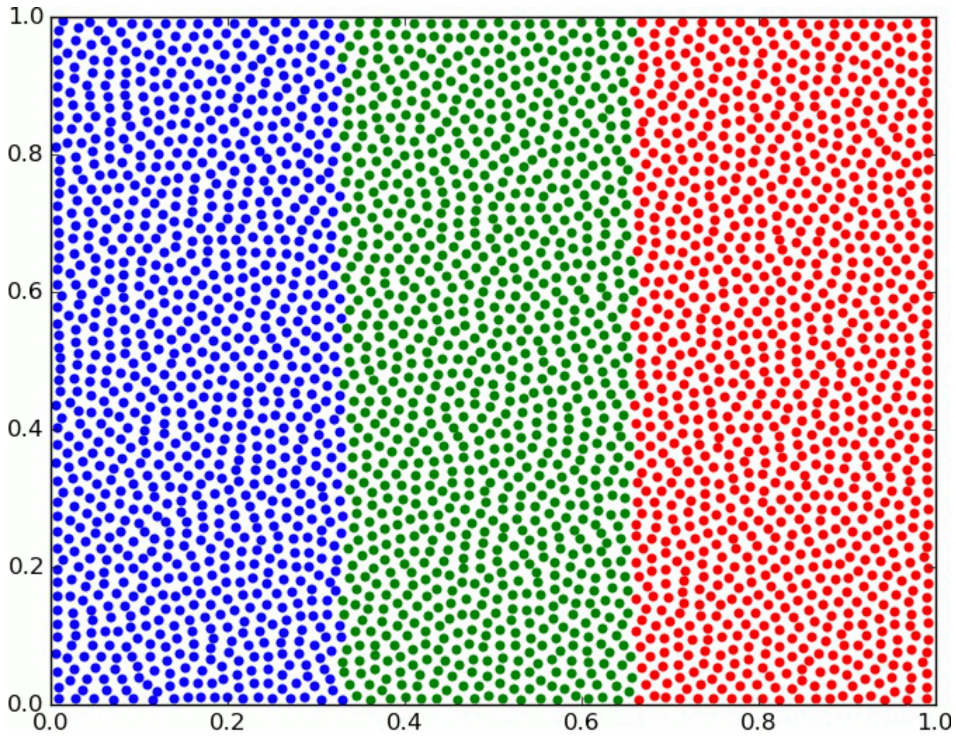
Application to fluid simulation

Fluids with Optimal Transport

- Lagrangian scheme
  - Add forces as usual (gravity, viscosity, surface tension…)
- Recover incompressibility throught OT [Gallouët & Mérigot 2016]
  - Computes OT from particles to uniform density
  - Add force from particle towards power cell centroid
  - Enforces particles to spread uniformly => incompressibility

Fluids with Optimal Transport

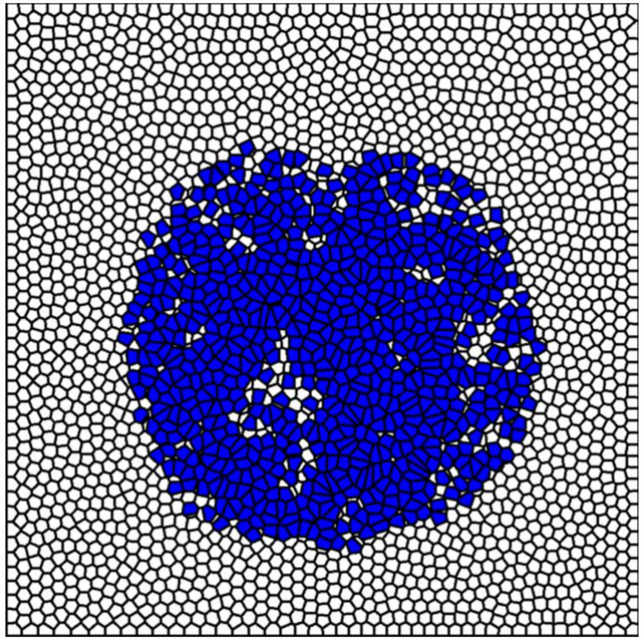
- Algorithm for 1 time step, at time step t : Explicit Euler
  - W = OT(Particles, Uniform Density)
  - For each particle:
    - $F^i_{spring} = \frac{1}{\epsilon^2}(Centroid(Laguerre_i) - X^t_i)$
    - $F^i = F^i_{spring} + m_i\vec{g}$
    - $v_i^{t+1} = v_i^n + \frac{dt}{m_i}F^t$
    - $X_i^{t+1} = X_i + dt\ v_i^{t+1}$
    - Bounce particles back into the domain



[Gallouët & Mérigot 2016]

My implem: $\epsilon = 0.004$, dt = 0.002 and $m_i = 200$.

Fluids with Optimal Transport



- For free-surface fluids, use partial optimal transport
- Keep air particles and fluid particles
  - Can perform Lloyd iterations on air particles
- Instead of enforcing that each particle has mass 1/N
  - Enforce each fluid particles to have mass $\mathrm{volume\_fluid}/\mathrm{N}_{\mathrm{fluid}}$
  - Enforce **the sum** of all air particles to have total mass volume_air

  ⇒ only 1 unknown w_{air} for all air particles

  ⇒ all air power cells have the same weight w_{air}.

“Partial Optimal Transport for a Constant-Volume Lagrangian Mesh with Free Boundaries” [Levy 2022]

Fluids with Optimal Transport

- Inside fluid $w_i > w_{air}$
  - Fluid cells erode air cells
- In total, $N_{fluid} + 1$ unknown power cell weights to determine
- Warm restart at each time step.
- Then, only move fluid particles

Fluids with Optimal Transport

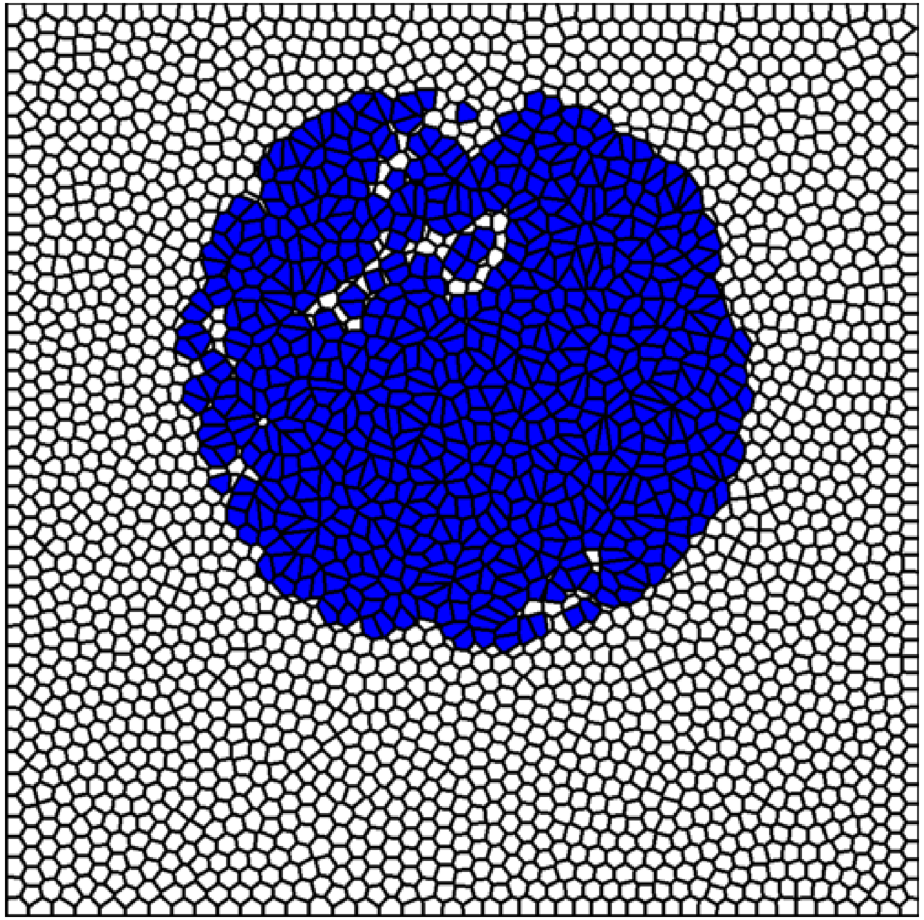
- In practice, now:

$$g(W) = \sum_{i=1}^{N_{fluid}} \int_{Pow_W(y_i)} (\|x - y_i\|^2 - w_i)\mathrm{d}x + \sum_{i=N_{fluid}+1}^{N_{fluid+N_{air}+1}} \int_{Pow_W(y_i)} (\|x - y_i\|^2 - w_{air})\mathrm{d}x + \sum_{i=1}^{N_{fluid}} \frac{vol_{fluid}}{N_{fluid}} w_i + vol_{air} w_{air}$$

$$\frac{\partial g}{\partial w(y_i)}(W) = \frac{vol_{fluid}}{N_{fluid}} - Area(Pow_i) \qquad \text{for } 1 \leq \mathrm{i} \leq \mathrm{N_{fluid}}$$

$$\frac{\partial g}{\partial w(y_{air})}(W) = vol_{air} - \sum_{j=N_{fluids}+1}^{N_{fluids}+N_{air}+1} Area(Pow_i) \qquad \text{for i} = \mathrm{N_{fluid}} + 1$$

Fluids with Optimal Transport



$N_{fluid} = 700,\ N_{air} = 2500,$ 1 sec/frame at beginning, 30 sec/frame at the end, with Nanoflann

Fluids with Optimal Transport

- Asymptotically, as $N_{air} \to \infty$
  - Still 1 unknown
  - Now, power cell of a fluid cell expressed as:

$$\|x - x_i\|^2 - w_i \leq \|x - x_j\|^2 - w_j \quad \text{for all fluid index j}$$
$$\|x - x_i\|^2 - w_i \leq \boxed{0} - w_{air}$$

There are air "particles" everywhere : min squared distance = 0

  - ⇒ Boundary of fluid = arc of a circle of radius $\sqrt{w_i - w_{air}}$

(recall $w_i \geq w_{air}$ in fluid and on its boundary)

  - So, fluid cells are intersections of Laguerre cells and disks
    - Can approximate disks with polygons and use Sutherland-Hodgman again!

Fluids with Optimal Transport

- In practice, now:

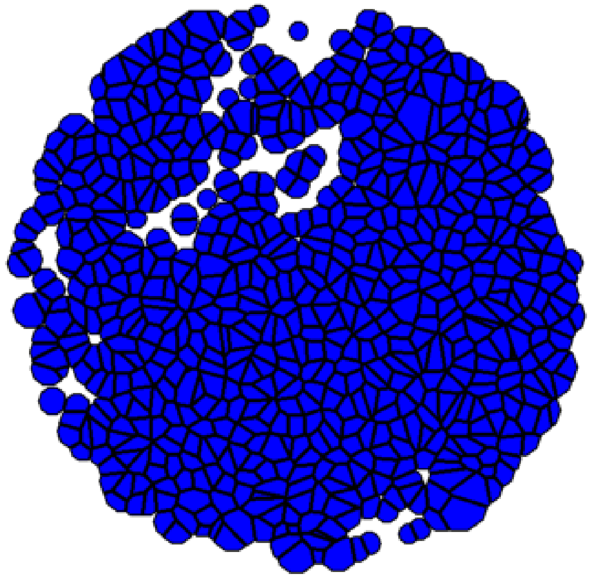
$$g(W) = \sum_{i=1}^{N_{fluid}} \int_{Pow_W(y_i)} (\|x - y_i\|^2 - w_i)\mathrm{d}x \; + \sum_{i=1}^{N_{fluid}} \frac{desired\ vol_{fluid}}{N_{fluid}} w_i + w_{air}(desired\ vol_{air} - estimated\ vol_{air})$$

$$\frac{\partial g}{\partial w_i}(W) = \frac{desired\ vol_{fluid}}{N_{fluid}} - Area(Pow_i) \qquad \text{for } 1 \leq \mathrm{i} \leq \mathrm{N}_{\mathrm{fluid}}$$

$$\frac{\partial g}{\partial w(air)}(W) = desired\ vol_{air} - estimated\ vol_{air}$$

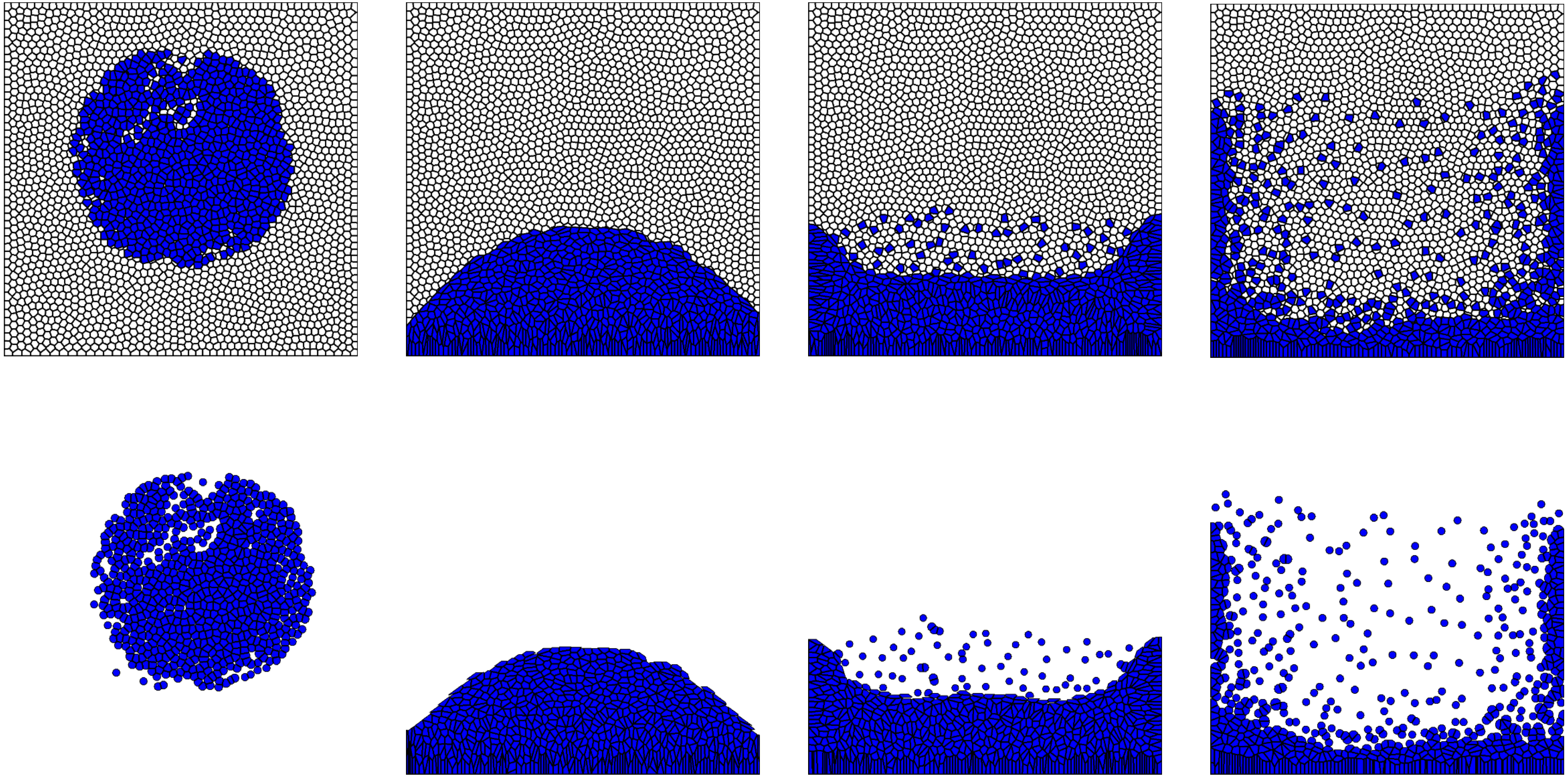
With $estimated\ vol_{air} = 1 - \sum_i Area(Pow_i)$

Fluids with Optimal Transport



$N_{fluid} = 700,\ N_{air} = 1,$ 1 sec/frame

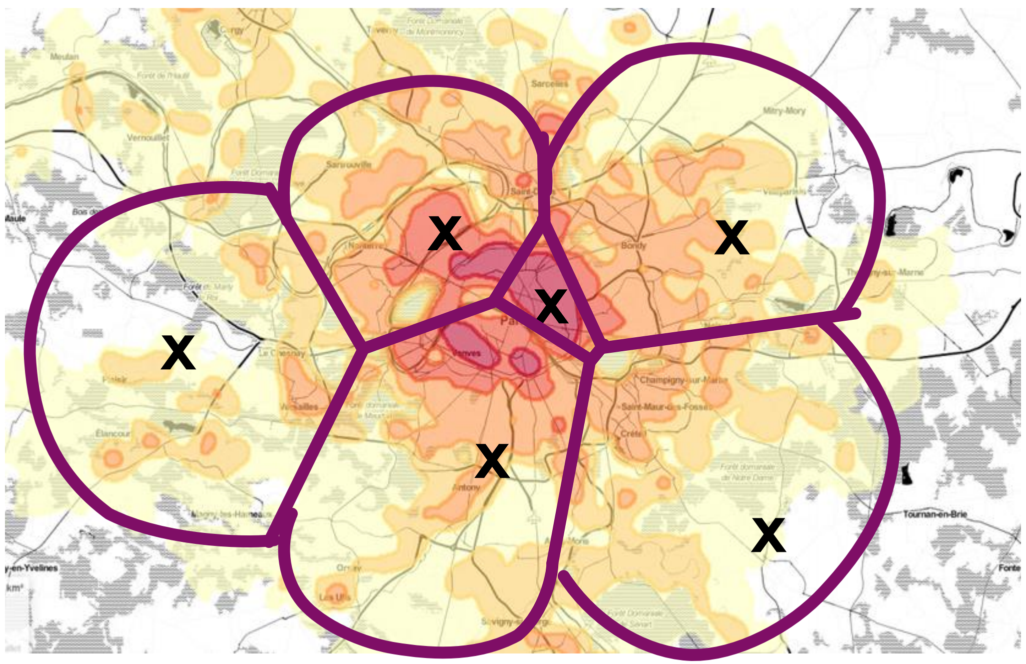
Fluids with Optimal Transport



Fluids with Optimal Transport

- Corresponds to transporting optimally volume_fluid among volume_total
  - Partial optimal transport

$units\ produced = area$



$units\ produced \leq area$

Regularized optimal transport

The Sinkhorn algorithm

- Kantorovich optimal transport: $\min_{m} \sum_i \sum_j c_{i,j}\, m_{i \to j}$ with constraints
- Rewritten as :

$$\min_{M \in \mathcal{U}(r,c)} \langle C, M \rangle$$

with $\mathcal{U}(r,c)$ matrices whose rows sum to r and columns to c

- Idea: consider instead

$$\min_{M \in \mathcal{U}(r,c)} \langle C, M \rangle - \epsilon E(M)$$

where $E(M) = -\sum M_{ij}(\log(M_{ij}) - 1)$ is the entropy, ϵ a small constant

Iterative Bregman Projections for Regularized Transportation Problems [Benamou et al. 2014]
Sinkhorn Distances: Lightspeed Computation of Optimal Transport [Cuturi 2013]

The Sinkhorn algorithm

$$\min_{M \in \mathcal{U}(r,c)} \langle C, M \rangle - \epsilon E(M)$$

- Can be rewritten as a projection:

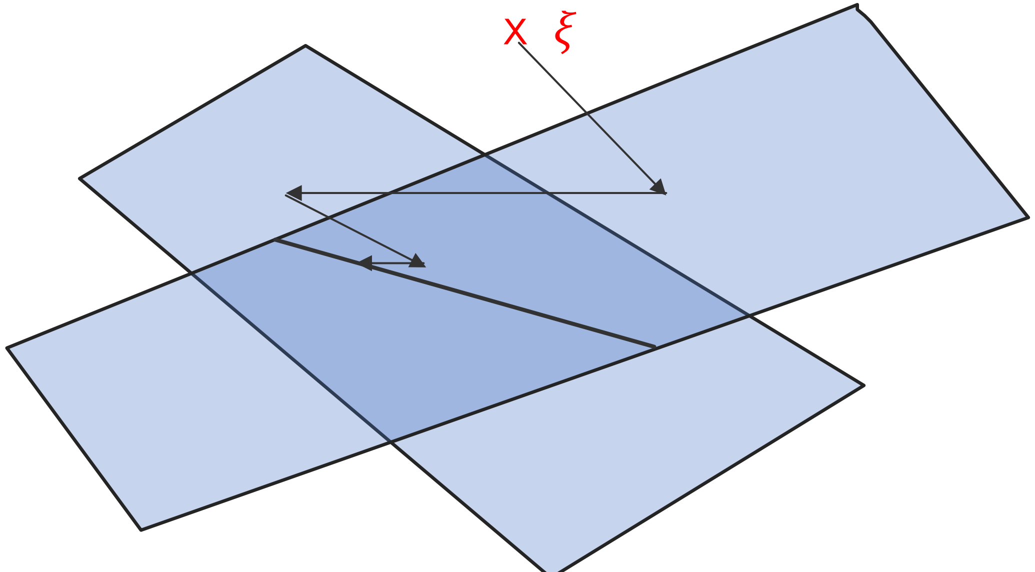
$$\min_{M \in \mathcal{U}(r,c)} KL(M, \xi)$$

where $\xi = \exp\left(-\frac{C}{\epsilon}\right)$ and $KL(M, \xi) = \sum M_{ij}\left(\log\left(\frac{M_{ij}}{\xi_{ij}}\right) - 1\right)$ the Kullback-Leibler divergence

The Sinkhorn algorithm

$$\min_{M \in \mathcal{U}(r,c)} KL(M, \xi)$$

- This is a projection on the intersection of two affine constraints, due to $\mathcal{U}(r,c)$
- We can thus apply Bregman projections: we iteratively project on each constraint



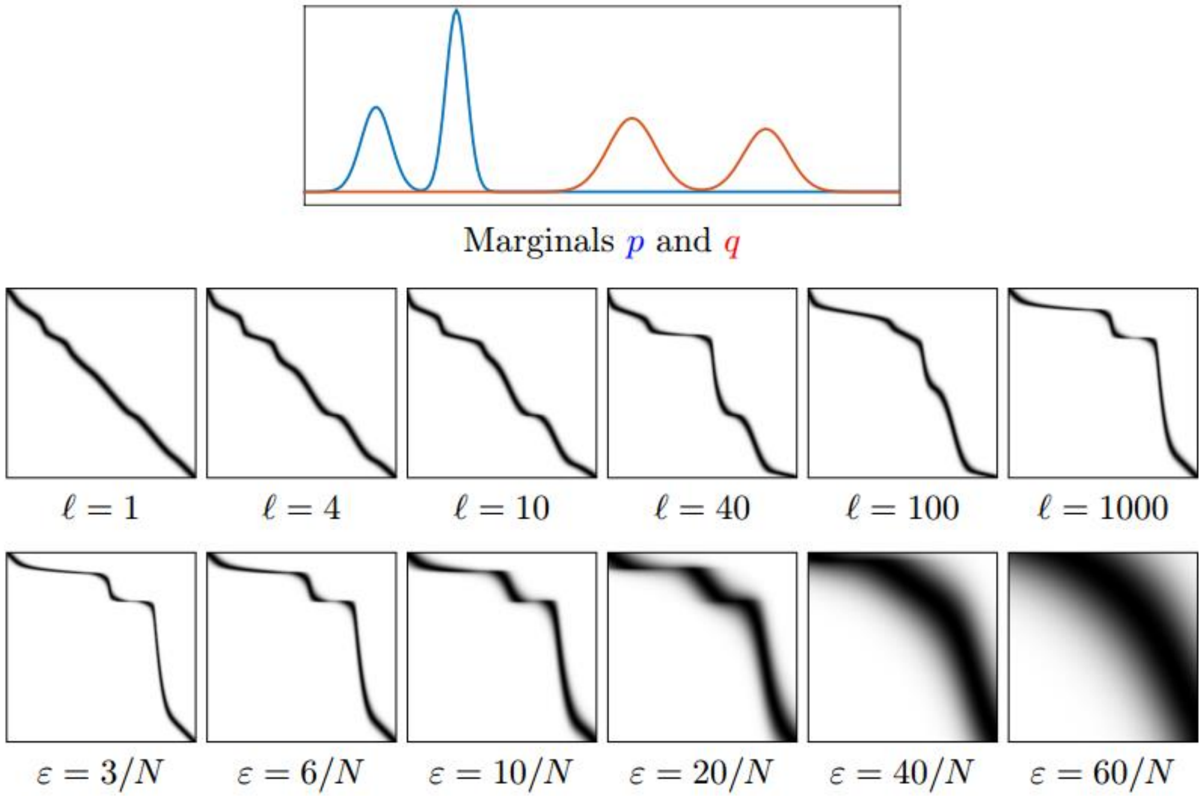
The Sinkhorn algorithm

- Projecting on constraints:
  - Constraints: $\sum_i M_{ij} = r_j$ and $\sum_j M_{ij} = c_i$
  - $M'_{ij} = \frac{M_{ij}}{\sum_i M_{ij}}.r_j$ and $M'_{ij} = \frac{M_{ij}}{\sum_j M_{ij}}.c_i$ corresponds to projection with KL
  - Row/column scaling
  - Corresponds to left/right multiplying M by diagonal matrix

The Sinkhorn algorithm

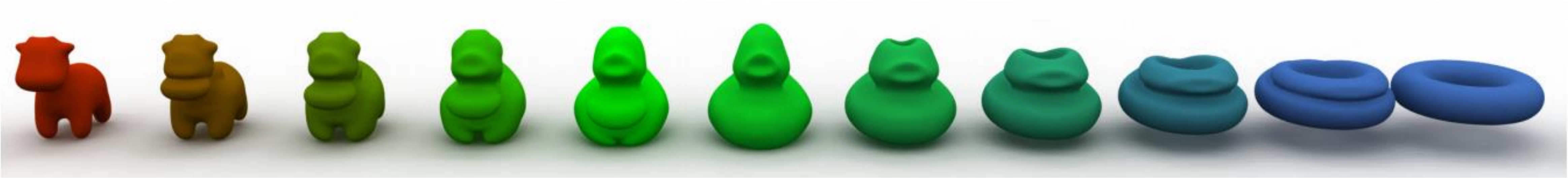
- We can thus apply Bregman projections: we iteratively project on each constraint
- We obtain the algorithm:

- $u^{(n)} = \frac{f}{\xi \, v^{(n)}}$
- $v^{(n+1)} = \frac{g}{\xi^T \, u^{(n)}}$
- $M = diag\left(u^{(n)}\right)\xi \; diag(v^{(n)})$



The Sinkhorn algorithm

- We realize that $\xi\, v^{(n)}$ can be computed efficiently
  - E.g., if $c(x,y) = \|x-y\|^2$, $\xi_{ij} = \exp\left(-\frac{\|x_i-x_j\|^2}{\epsilon}\right)$
  - Then $\xi\, v^{(n)}$ is just a Gaussian convolution
  - So, it is a separable operator, and efficiently done in high-dimension

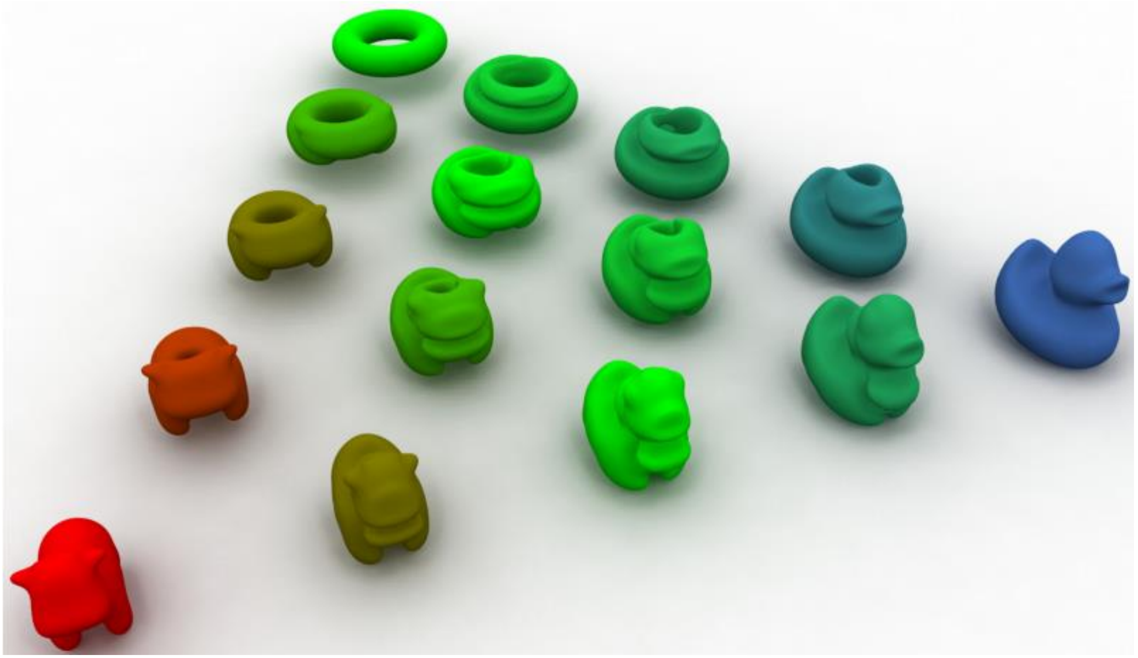


Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains [Solomon et al. 2015]

The Sinkhorn algorithm

- Generalized to compute displacement interpolation and barycenters

- $b_s^{(0)} = 1 \;\; \forall s$
- for $\ell = 0 \;\ldots L$
  - $a_s^{(\ell)} = \frac{p_s}{K\, b_s^{(l-1)}} \quad \forall s$
  - $p(\lambda) = \prod_s \left(K^T a_s^{(\ell)}\right)^{\lambda_s}$
  - $b_s^{(\ell)} = \frac{p(\lambda)}{K^T\, a_s^{(\ell)}} \qquad \forall s$



The Sinkhorn algorithm

- Issues
  - Unstable as regularization decreases
    - Computations in log-domain
  - Number of iterations should increase as regularization decreases
    - Multiscale computations
  - $W_\epsilon(f,f) \neq 0$
    - $\widetilde{W}_\epsilon(f,g) = W_\epsilon(f,g) - \frac{1}{2}(W_\epsilon(f,f) + W_\epsilon(g,g))$