Numerical methods for Optimal Transportation

MokAlien
Numerical Optimal Transportation Workshop
U Paris Dauphine
Nov 10th, 2015

Adam Oberman, McGill University
Ruan Yanglong, McGill University

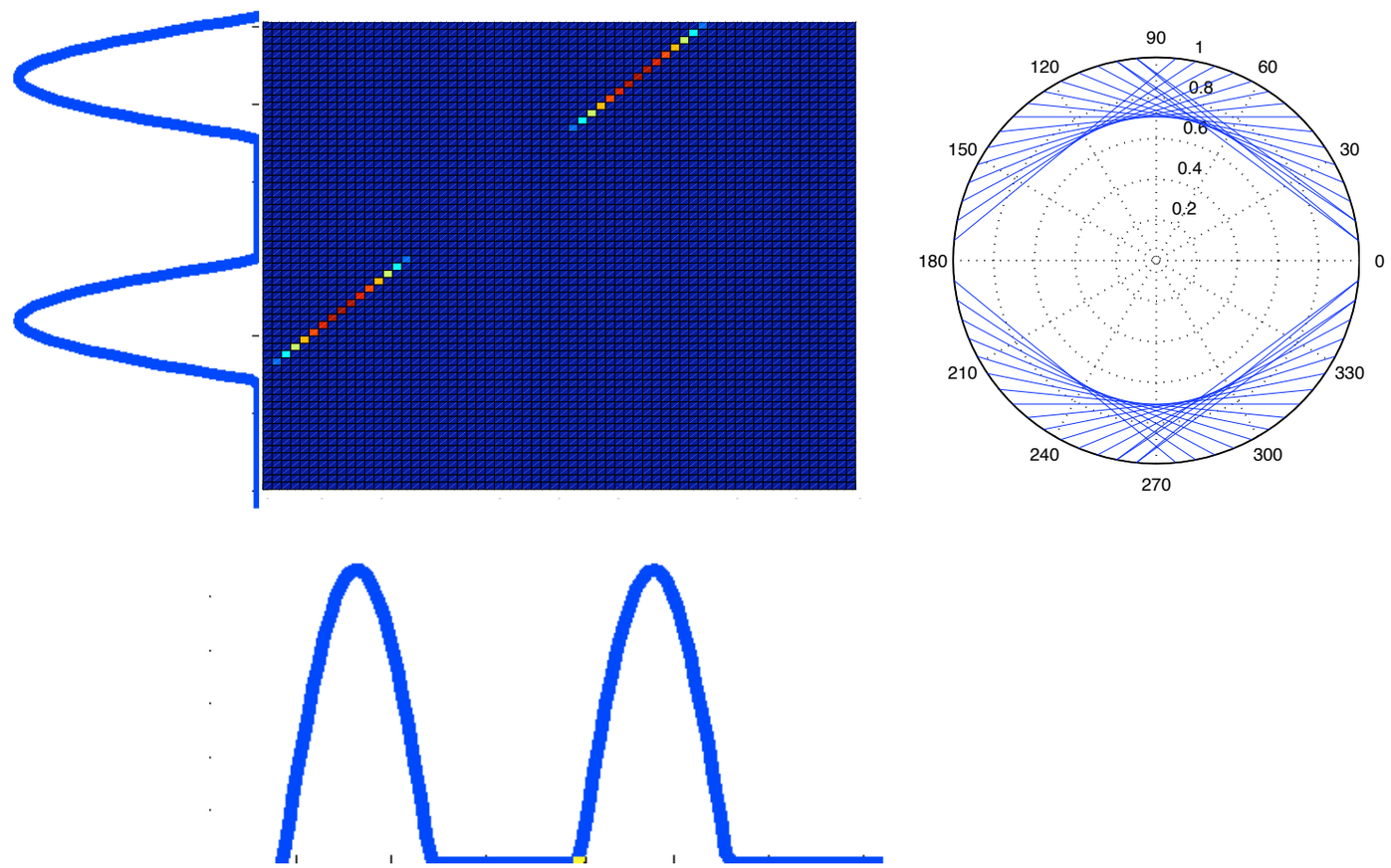
What theory is available
to build a numerical method
for Optimal Transportation?

Linear Programming for OT?

- measures are discrete.
- Size of LP is very large: quadratic in number of points
- Too costly for moderate size problems in two dimensions

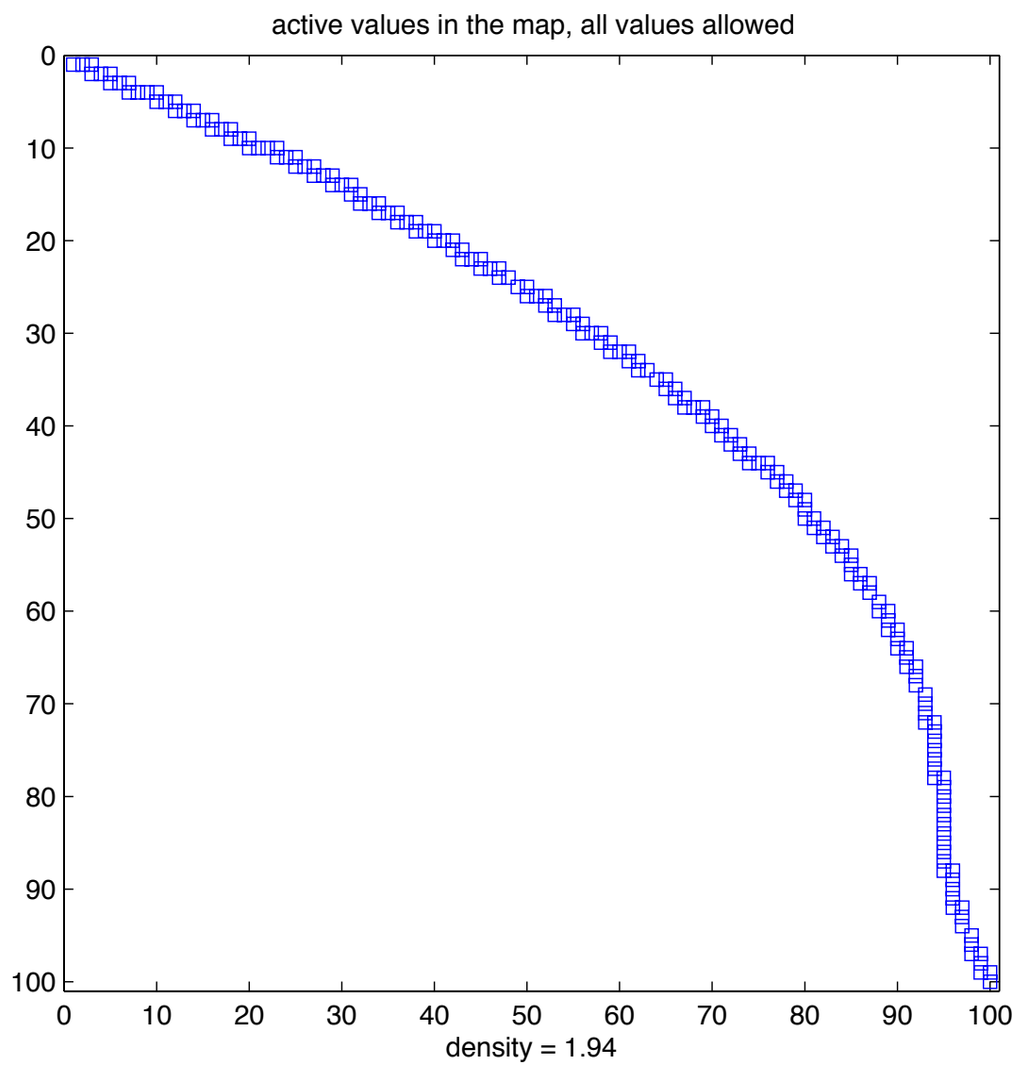
$$c(x, y) = |x - y|^2$$

[McCann, exact solns on line]



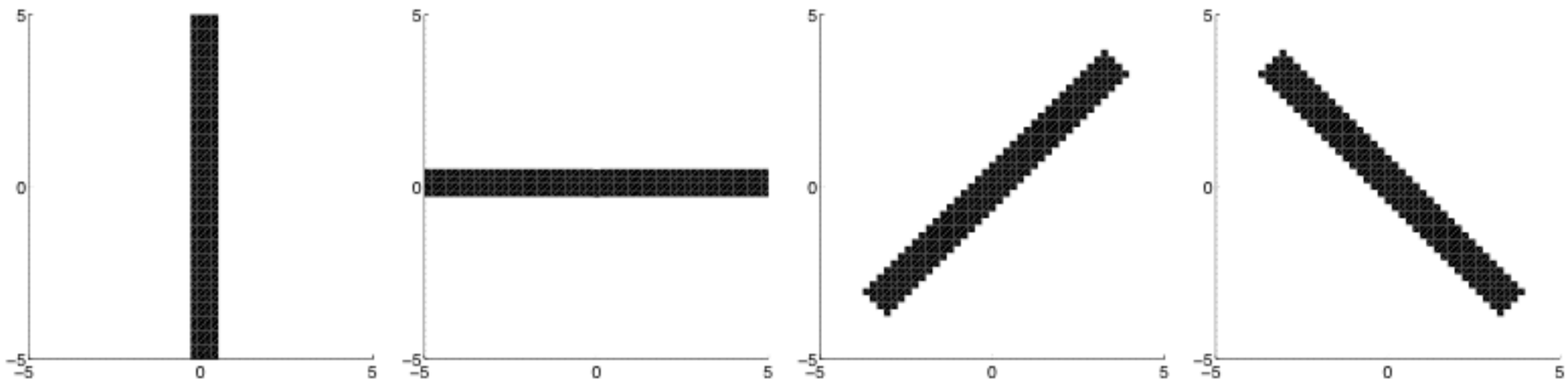
Linear Programming for OT?

In general, solution is very weak, not even a map!

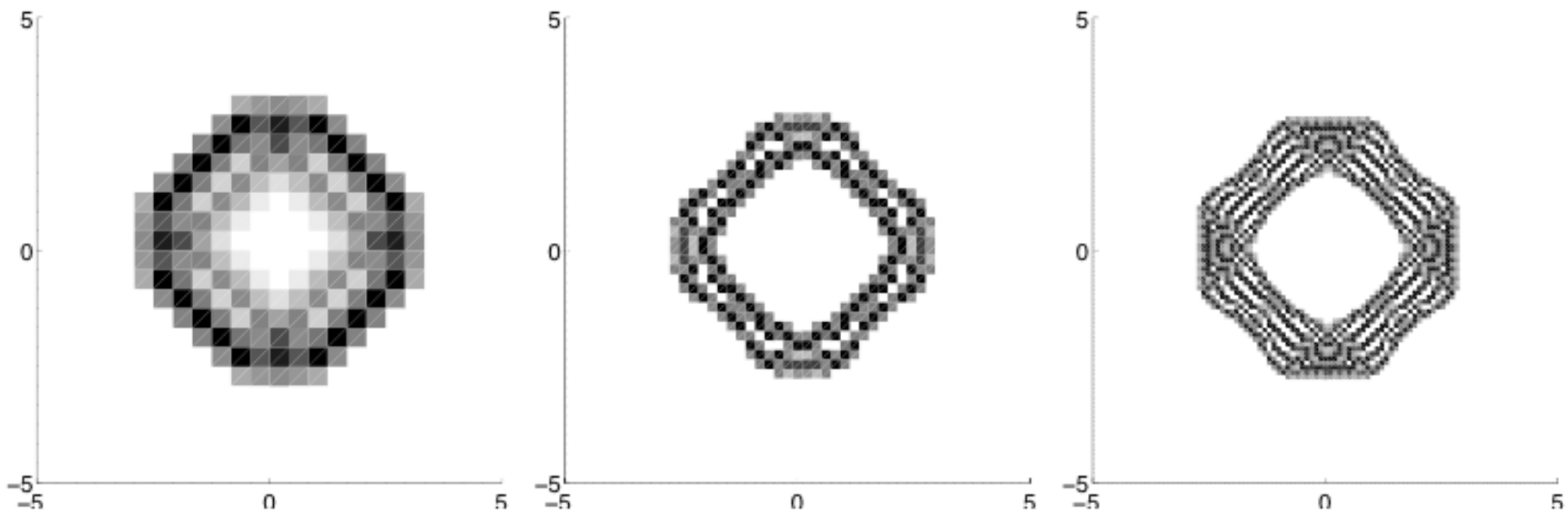


- Each point in source is split, and mapped to two (or more) points in the target.
- Accuracy hard to define, not even first order.

Weak Convergence for Barycenters



Barycentre of four rectangles is a measure. Convergence is very weak, solutions develop oscillations as the grid resolution is increased.



What about PDE method?

$$\det(D^2 u(x)) = \frac{f(x)}{g(\nabla u(x))}$$

$$\nabla u(X) = Y.$$

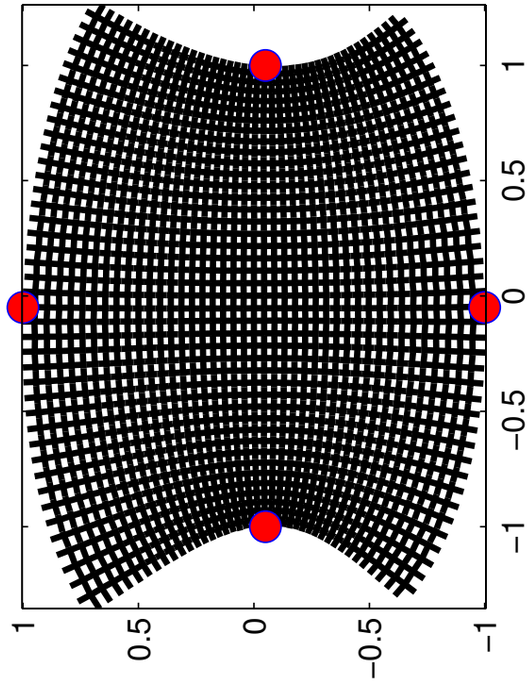
$$u \text{ is convex.}$$

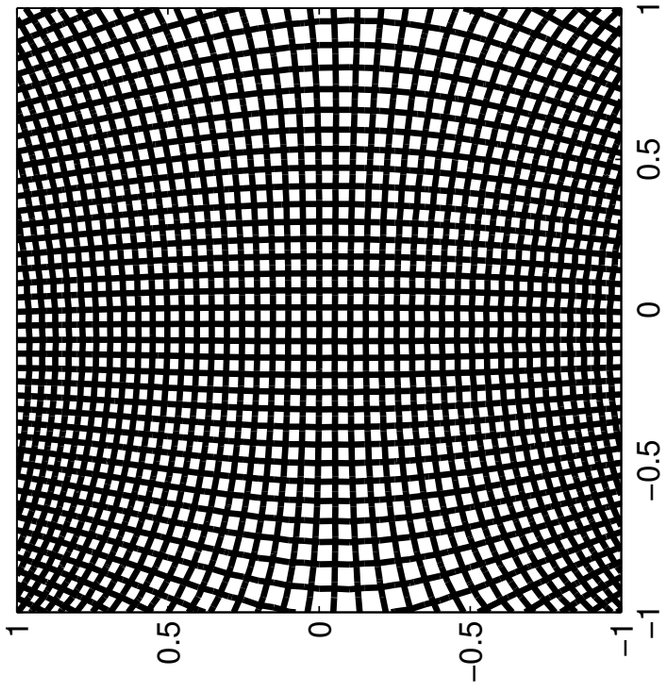
- Need to solve fully nonlinear elliptic PDE
- Need to enforce convexity conditions
- Missing standard boundary conditions, instead have a condition on the gradient map

First two points challenging, but developed tools over several years.

Is there PDE theory for mappings?

- Conformal mappings
  - angle preserving maps, have Riemann Mapping Theorem.
  - class of analytic maps not stable enough
- Nonlinear Elasticity Theory:
  - different boundary conditions
  - variational theory
- Conclusion: existing theory not much help.

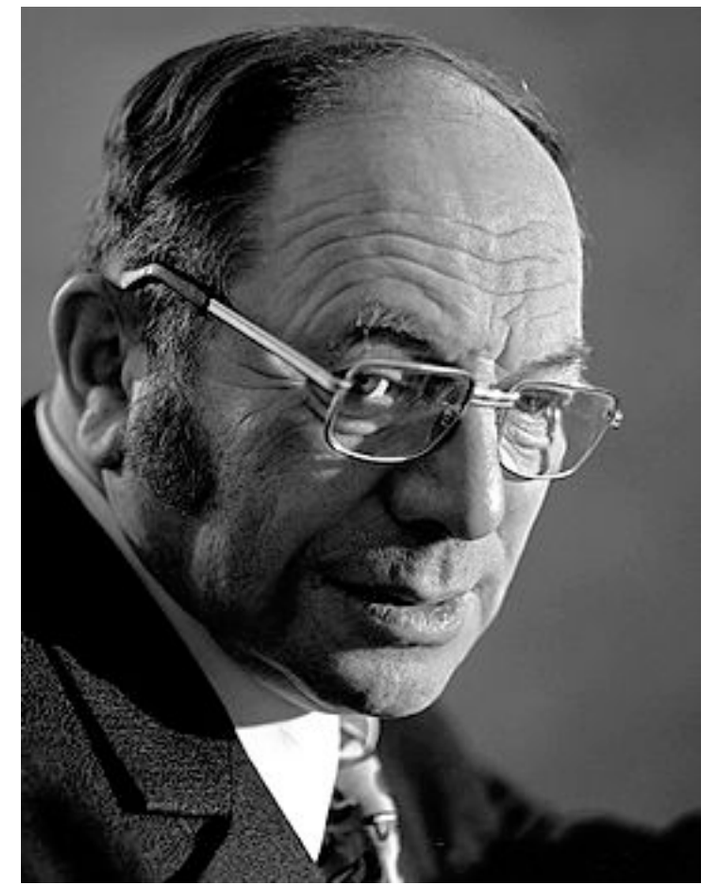




Computer Science Algorithms?

- Wasserstein distance called Earth Mover Distance in CS.
- Indyk developed very fast algorithm developed for Wasserstein-1 distance by counting points at different scales.
- This method, equivalent to an imbedding into L1, introduces a “distortion”.
- Naor proved that metric embeddings of this kind will always have a significant distortion.

Introduction and Theory



Leonid Kantorovich

Optimal Transportation

Given: μ, ν probability measures with bounded supports X, Y

$$d\mu(x) = f(x)dx, \qquad d\nu(y) = g(y)dy$$

cost function $c(x, y) : X \times Y \to \mathbb{R}$.

Goal: find a mapping to "rearrange" one measure into the other, which minimizes $c(x, y)$ weighted by the amount of mass transported.

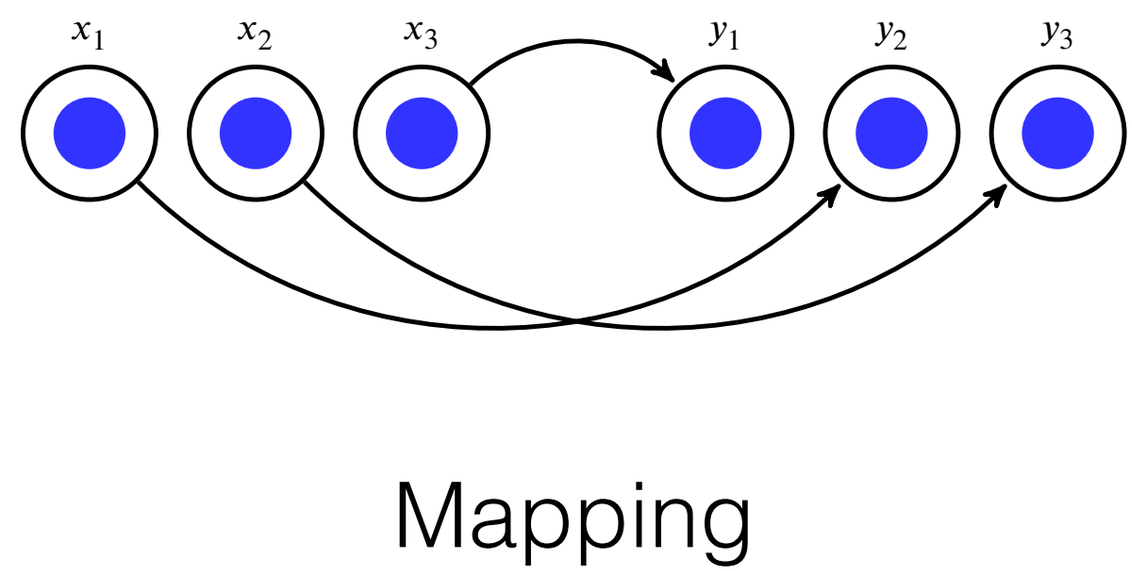
There are two different notions of "rearrangement"

The Monge Problem

Monge's problem is to minimize the total work corresponding to a map T over the set of all measurable maps T which transport μ onto ν.

$$\text{Minimize } I[T] = \int_X c(x, T(x)) d\mu(x)$$

$$\text{Subject to: } T\#\mu = \nu$$



Suppose the measures have smooth densities, and consider the class of measurable, one-to-one mappings T which rearrange μ into ν. If T is continuously differentiable, the change of variable formula from multivariable calculus leads to

$$f(x) = g(T(x))|\det \nabla T(x)|$$

The Monge Ampere PDE (with OT BC)

Case where $c(x, y) = |x - y|^2$

$$\text{Minimize } I[T] = \int_X |x - T(x)|^2 f(x) dx$$

$$\text{Subject to: } f(x) = g(T(x)) |\det \nabla T(x)|$$

When $c(x, y) = |x - y|^2$, the densities are smooth, and Y is convex, the Monge problem becomes a fully nonlinear Monge-Ampère PDE. The optimal map $T = \nabla u$ is given by the gradient of a convex function

$$\det(D^2 u(x)) = \frac{f(x)}{g(\nabla u(x))}$$

$$\nabla u(X) = Y.$$

$$u \text{ is convex.}$$

Remarks on derivation of Monge-Ampere PDE

- Fact: optimal map is cyclically monotone (meaning exchanges of images of points will not improve the quadratic cost)
- Convex Analysis Theorem: cyclically monotone maps are (sub-)gradients of convex functions
- Regularity theory: map is actually a gradient.
- Ref: Evans notes / Villani book
- Very special to the case of quadratic costs

The Kantorovich Formulation

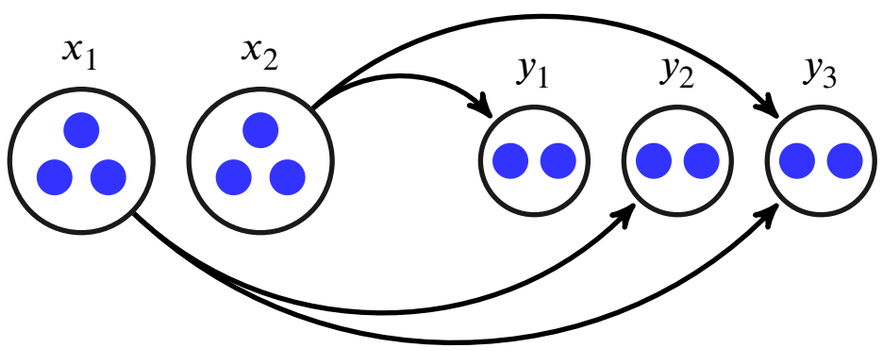
Given: μ, ν probability measures with bounded supports X, Y

Weak formulation as infinite dimensional Linear Program. Transference plan generalizes a mapping, in order to allow for mass to be split into multiple parts.

Consider a probability measure, π, on the product space $X \times Y$, whose marginals are μ and ν.

transference plans $\Pi(\mu, \nu)$ $\quad \pi[A \times Y] = \mu(A), \quad \pi(X \times B) = \nu(B)$

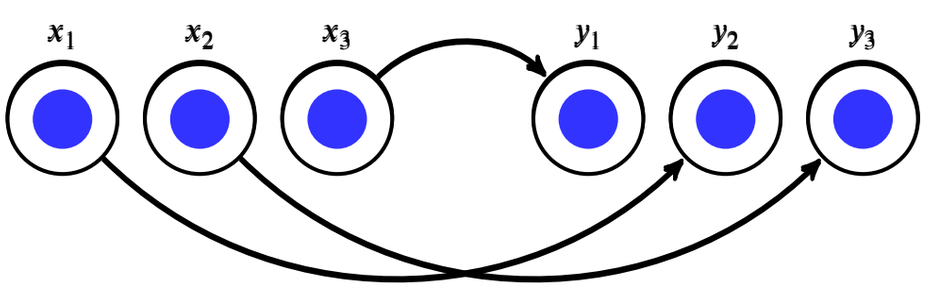
for all measurable subsets A of X and B of Y.



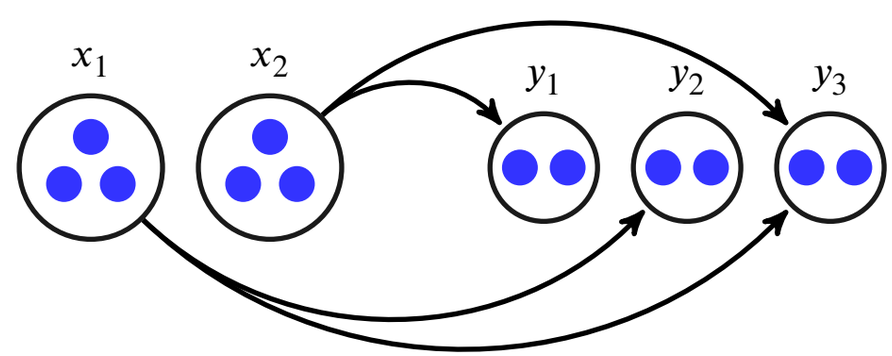
$$\text{Minimize } I[\pi] = \int_{X \times Y} c(x, y) d\pi(x, y), \qquad \text{for } \pi \in \Pi(\mu, \nu)$$

Comapare Monge to Kantorovich

- Monge: looks for an optimal map (strong).
- Kantorovich: look for an optimal transference plan (weak).
- For certain costs, including the quadratic cost, (more generally, there is a twist condition) the optimal transference plan is actually a map (so recover the Monge Formulation).

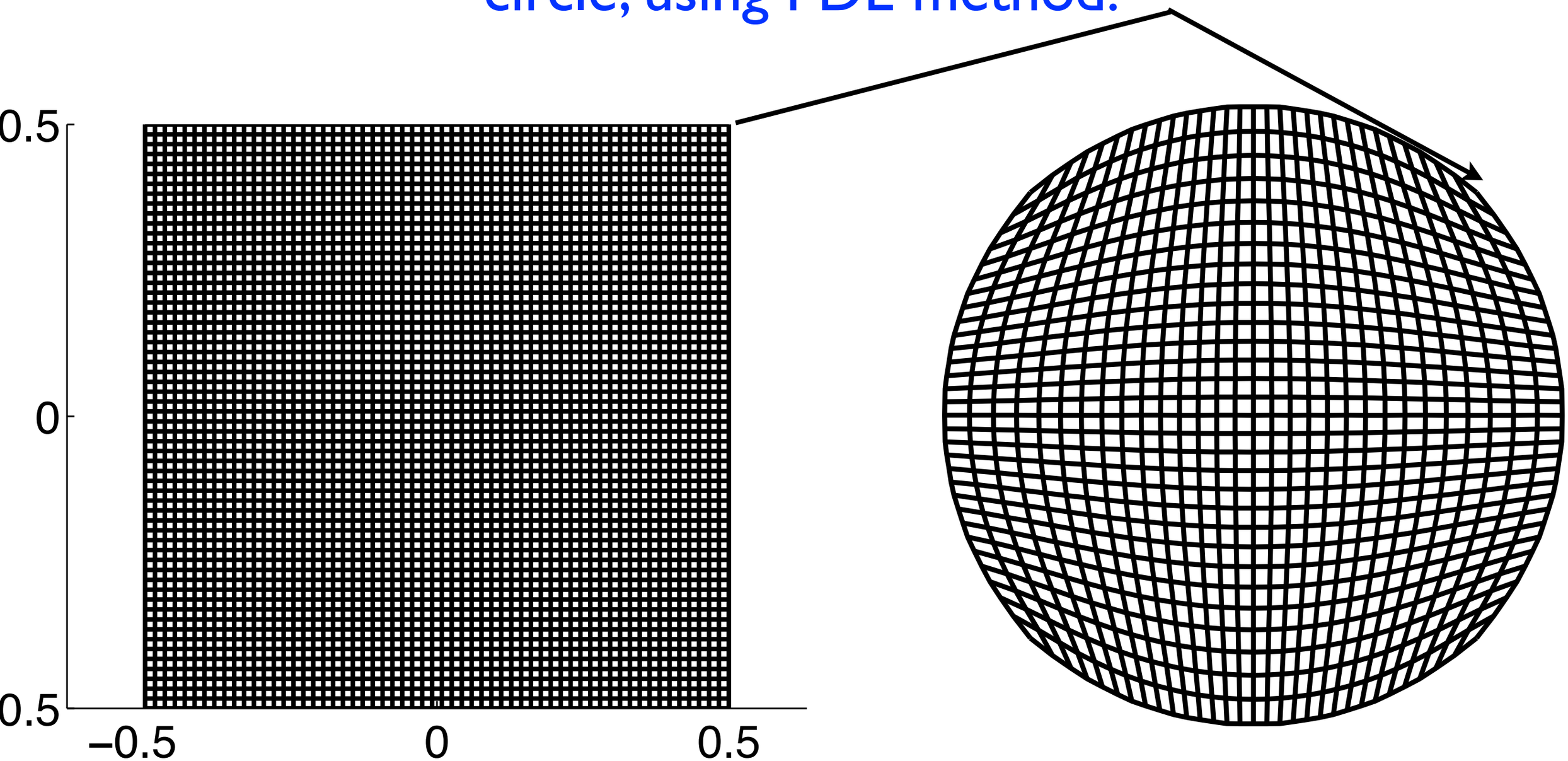


Mapping (strong)



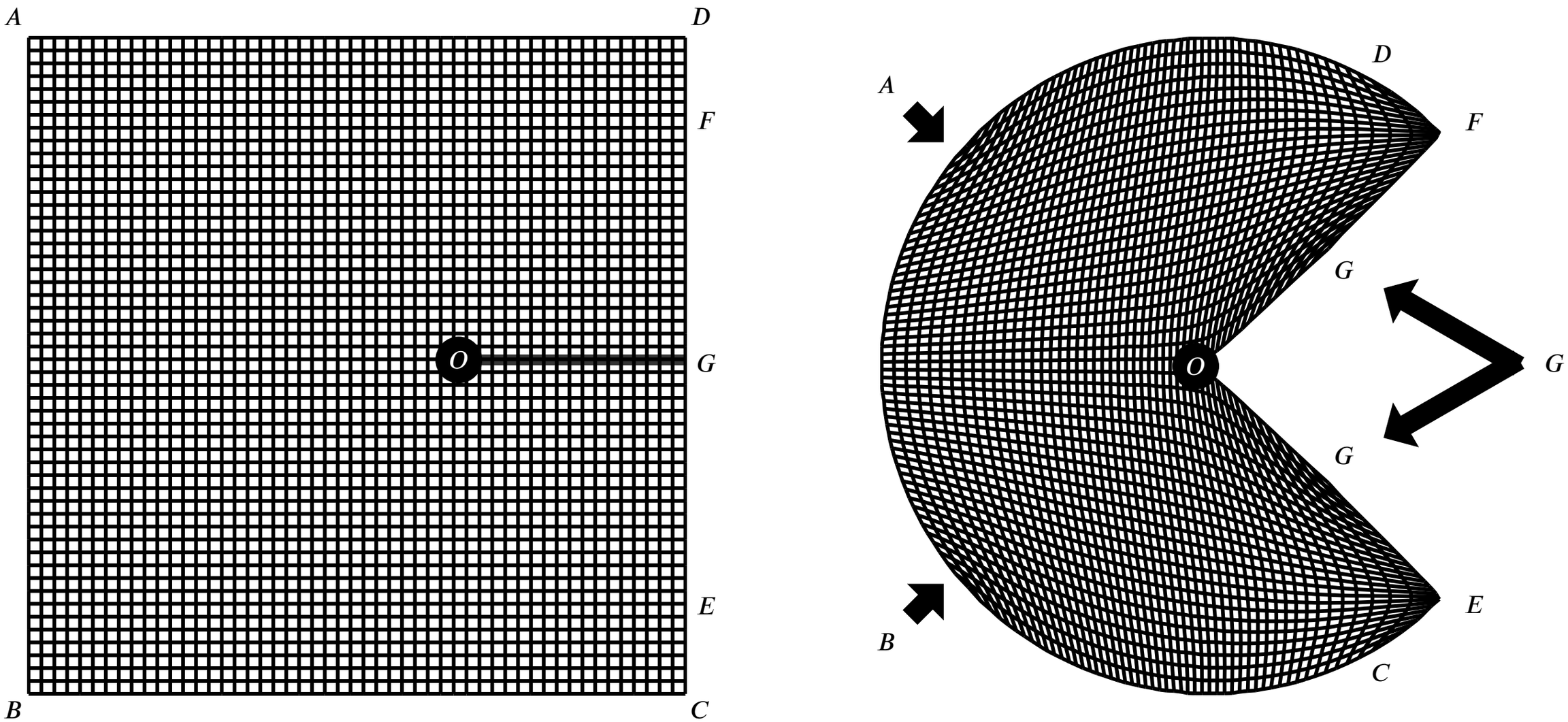
Transference plan (weak)

Optimal map between (uniform densities on) the square and circle, using PDE method.



How to read the map: four squares on the left gets mapped to a distorted square on the right. Centre goes to centre, corners go to diagonals.

Optimal map between (uniform densities on) the square and Pac-Man using LP method



How to read the map: Labels go to labels. The line segment on right gets split into two parts on the mouth of Pac-Man.

Linear Programming Discretization

Numerical Methods for OT

- PDE Methods for Monge-Ampere (no OT)
  - Many methods, see review by Feng-Neilan-Glowinsky, see also new methods by Mirebeau.
- PDE Methods
  - Benamou Brenier [00]
  - Benamou-Froese-Oberman [14]
- Earth Mover Distance
  - Indyk 08. But Naor proves introduces distortion
- Entropic Regularization
  - Cuturi 13, Carlier-Oberman-Oudet 14, Cuturi-Peyre-Rolet 15,
  - Solomon + 7 [15], Pixar
  - Benamou-Carlier-Cuturi-Nenna-Peyre [2015] Bregman Iterations
- Multiscale solvers: Schmitzer [13,15] two methods

Discretization of Optimal Transportation problem

- Discretize the measures on a uniform grid.
- Dirac deltas at the centre of the square, weights equal to the measure of the square

Each x_i is the center of a hypercube of width h,

$$R_h(x_i) = \left\{ x \in \mathbb{R}^d \;\middle|\; \|x - x_i\|_\infty = \frac{h}{2} \right\}.$$

Define the approximate measures μ^h, ν^h, to be a weighted sums of Dirac masses whose weights corresponds to the integral of the measures over the hypercubes of width h centred at x_i.

$$\begin{aligned} \mu^h &= \sum_{i=1}^{n} \mu_i^h \delta_{x_i}, \quad \mu_i^h = \mu(R_h(x_i)) \\ \nu^h &= \sum_{i=1}^{m} \nu_i^h \delta_{y_i}, \quad \nu_i^h = \nu(R_h(y_i)) \end{aligned} \tag{6}$$

The discrete cost function is given by

$$c_{ij} = c(x_i, y_j), \tag{7}$$

- With this discretization, the infinite dimensional problem becomes finite dimensional LP, the assignment problem.

Finite Dimensional Linear Program for OT

Given discrete probability measures

$$\mu = \sum_{i=1}^{n} \mu_i \delta_{x_i}, \quad \nu = \sum_{j=1}^{m} \nu_j \delta_{y_j}, \qquad \mu_i, \nu_j \geq 0, \ \sum_{i=1}^{n} \mu_i = \sum_{j=1}^{m} \nu_j = 1.$$

The cost function is given by the non-negative $n \times m$ matrix, $c = (c_{ij})$.

$$\Pi(\mu, \nu) = \left\{ \pi = (\pi_{ij}) \mid \sum_{j=1}^{m} \pi_{ij} = \mu_i, \quad \sum_{i=1}^{n} \pi_{ij} = \nu_j, \quad \pi_{ij} \geq 0 \right\}$$

The transportation linear program is given by

$$\text{Minimize } I[\pi] = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} \pi_{ij}, \qquad \text{for } \pi \in \Pi(\mu, \nu)$$

Stability implies convergence

- Weak convergence follows from textbook stability theory as in [Villani Theorem 5.20]
- Remark: no parameters in this method

Theorem 3 (Stability of optimal transportation). *Assume X and Y are compact, and that $c(x, y)$ is continuous. Let μ^{h_k} and ν^{h_k} be a sequence of measures that converge weakly to μ and ν, respectively. For each k let π^{h_k} be an optimal transference plan between μ^{h_k} and ν^{h_k}. Then, up to extraction of a subsequence,*

$$\pi^{h_k} \text{ convergences weakly to } \pi$$

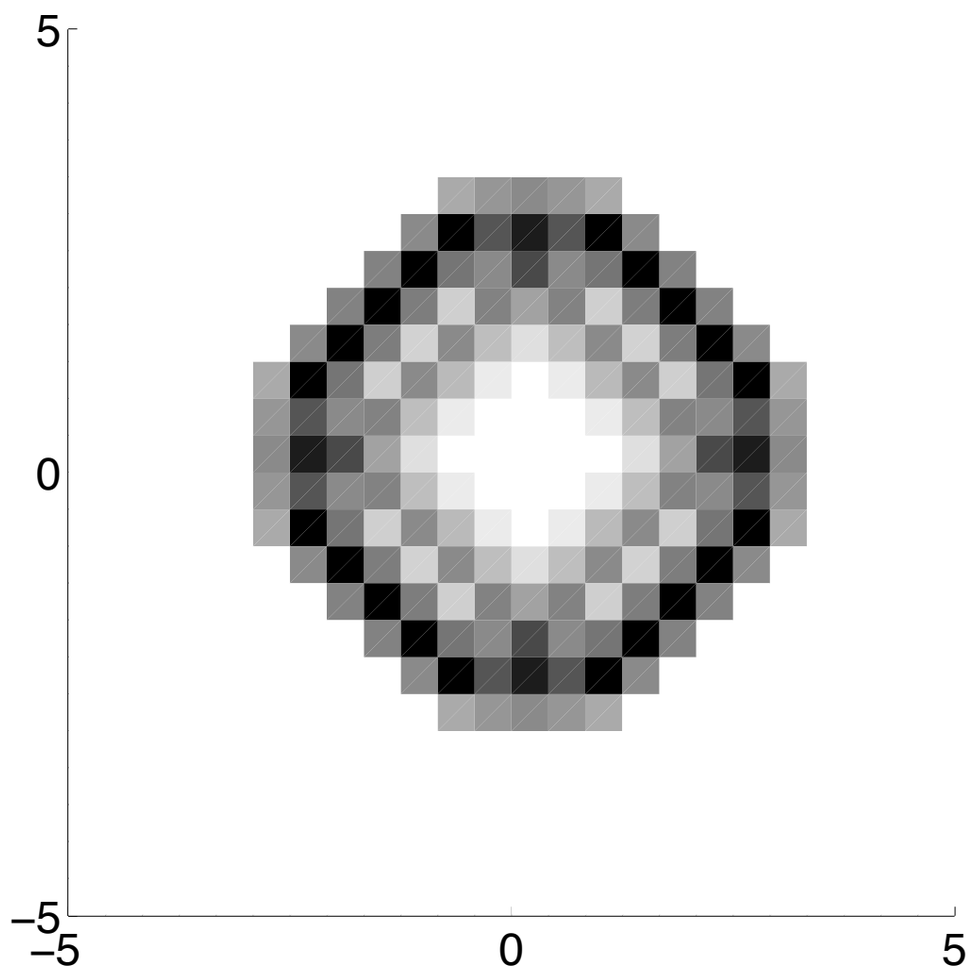
where π is an optimal transference plan between μ and ν.

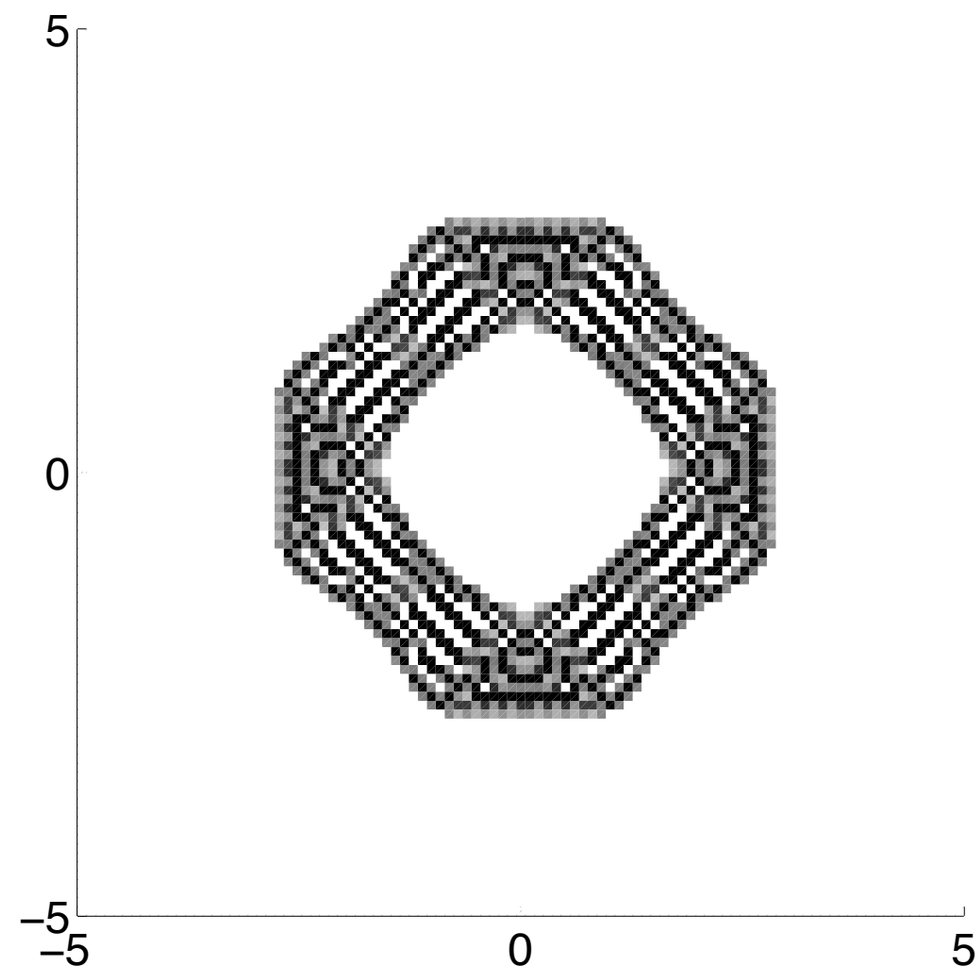
Theorem 4 (Convergence of Linear Programming Solutions). *Suppose that X and Y are compact, and that $c(x, y)$ is continuous. Let π^h be a solution of (LP) given by the discretization (6) (7) (8). Then, up to extraction of a subsequence, π^h convergences weakly to an optimal transference plan solution of (KLP).*

Weak convergence: solution is a plan (not a map)

Weak solutions are a major problem for Linear programming solutions. Very poor accuracy compared to PDE solvers.

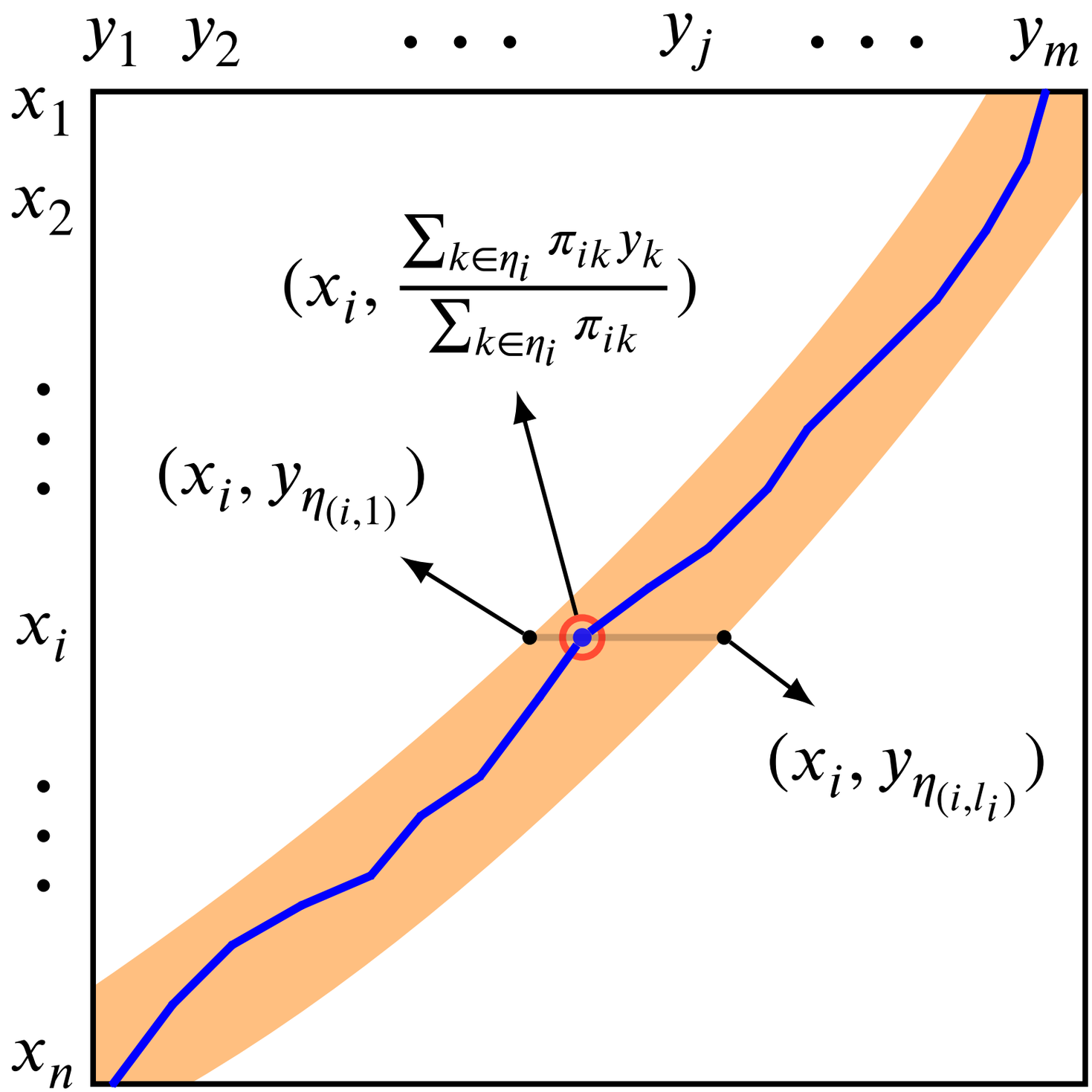
Similar problem with earlier work on Barycenters. The solutions oscillate at fine scales. This doesn't get better as resolution increased.





Linear Programming Accuracy

Barycentric Projection



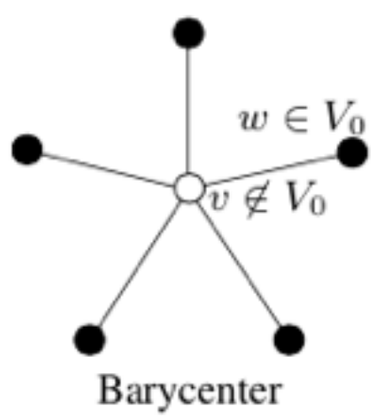
Improve Accuracy Using Barycentric Projection

Definition 5 (Barycentric projection of transference plan)**.** Define for each i with $\mu_i > 0$, $\bar{y}_i$ to be the Euclidean barycenter of the points $(y_1, \ldots, y_m)$ with weights $(\pi_{i1}, \ldots \pi_{im})$,

$$\bar{y}_i = \frac{\sum_{k=1}^{m} \pi_{ik} y_k}{\sum_{k=1}^{m} \pi_{ik}}.$$

Then define the barycentric projection of the transference plan π^h by

$$\bar{\pi}^h = \sum_{i=1}^{n} \mu_i^h \delta_{(x_i, \bar{y}_i)}.$$



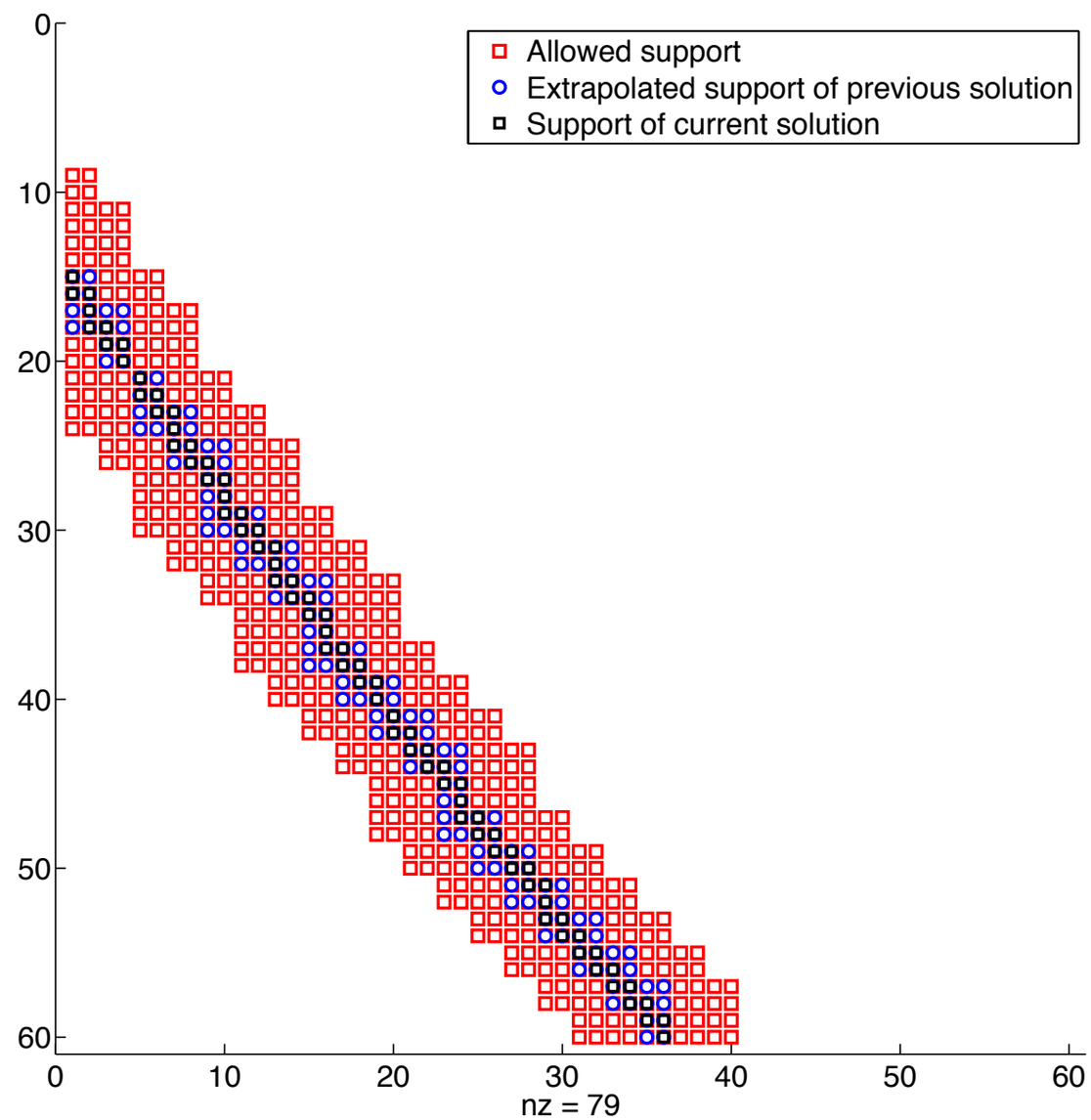
Barycentric projection is discussed in [AGS06]. In particular, by Theorem 5.4.4 and by Lemma 12.2.3, we can conclude that, for convex costs, the barycentric projection of the approximations converges to the barycentric projection of the limiting transference plan. In particular, when the unique limit is a map π, then $\bar{\pi}^h$ converges to π.

Linear Programming Efficiency

- So far, method converges, and can make it accurate.
- But the full LP can only solve small sized problems.
- Next: Sparse LP solver, based on refining the grid

Efficient Method - Sparse LP

- Use the fact that the solution is known to be a map. Solve a series of sparse LP, using grid refinement.
- Complicated to code. Refine grid, extrapolate support, pad support, then solve sparse LP.
- Size of LP is linear instead of quadratic in the number of masses



Efficient Method - Sparse LP

3.1. **The sparse linear program.** Let $\pi = \pi_{ij}$ be the solution of (LP), and let K_0 be the basis set, the indices of the nonzero entries of π. If $K_0 \subset K$, for a known set, K, then we can recover π by solving the reduced linear program,

$$\text{Minimize} \quad \sum_{(i,j)\in K} c_{ij}\pi_{ij},$$

$$\text{Subject to:} \quad \begin{cases} \displaystyle\sum_{\{j|(i,j)\in K\}} \pi_{ij} = \mu_i, & i \in [1,\ldots,n] \\ \displaystyle\sum_{\{i|(i,j)\in K\}} \pi_{ij} = \nu_j, & j \in [1,\ldots,m] \\ \pi_{ij} \geq 0, & (i,j) \in K \end{cases}$$

- Number of nonzeros (sparsity) is |K| instead of nm

Performance: Implementation

- Matlab for discretization. Called several LP solvers: CLPEX, MOSEK, Gurobi
- Compared speed and memory with full LP and ER.
- Our method is a lot of work to code.
- Entropic Regularization: has the advantage that it is a few lines of code. However, the regularization needs to be handled carefully.

Highlight: best of each:

- Full LP: 8,000 masses, 500 sec, 7GB memory,
- ER: 18,000 masses, 300 sec, 6 GB.
- Sp-LP 500,000 masses: 287 sec. 4 GB.

Performance: Linear Efficiency

N (grid)	$2N^2$	MGLP		FLP		ER	
		CPU	Memory	CPU	Memory	CPU	Memory
32	2048	0.9	14	14.0	400	3.8	900
48	4608	1.6	40	95.6	2000	20.1	1200
64	8192	3.2	80	527.1	7000	64.4	1900
96	18432	7.0	160	*	*	310.1	6200
128	32768	13.5	300			*	*
192	73728	35.3	700				
256	131072	58.9	1100				
384	294912	165.5	2500				
512	524288	287.6	4000				

TABLE 1. Comparison of run time and memory usage for the multigrid LP (MGLP), the full LP (FLP), and the entropic regularization (ER) method, the latter with precision to 10^{-4} and regularization parameter $\epsilon = 10^{-3}$ ([BCC+15]). Precision for the LP solvers are both 10^{-8}. Memory usage for the LP solvers are as reported by Gurobi, and is estimated for the ER method.

More on the grid refinement

Details of Grid refinement

(0) The full linear program (LP), with $c_{ij} = c_{ij}^h$ and $\mu = \mu^h, \nu = \nu^h$ as given by the quadrature rules (6) and (7), is solved on a coarse grid, $X^h \times Y^h$.

(1) From the discrete solution, π_{ij}^h, recover the spatial support set,

$$S_h = \left\{ R_h(x_i^h) \times R_h(y_j^h) \mid \pi_{ij}^h > 0 \right\}, \qquad S_h \subset X^h \times Y^h.$$

Grow the spatial support set, by including allowing neighbours (in space),

$$\bar{S}_h = \{ \text{ neighbours } S_h \}.$$

Extract the corresponding indices

$$\overline{K}_h = \{ \text{ indices of center points in } \bar{S}_h \}.$$

(2) Refine the grids in each domain by a factor of two, labelling the new grids $X^{h/2}$, $Y^{h/2}$. In each coordinate of $x = (x_1, \ldots, x_d)$, the interval $[x_i - h, x_i + h]$ is halved, to become $[x_i - h, x_i], [x_i, x_i + h]$, and the two new points are generated at the midpoints of the interval. Thus, the hypercube $R_h(x)$ is divided evenly into 2^d hypercubes, and the grid point x generates 2^d new points on the refined grid, each at the centres of the smaller hypercubes.

(3) The allowed spatial support is interpolated onto the finer grid $S_{h/2} = \bar{S}_h$ and the corresponding indices are extracted

$$K_{h/2} = \{ \text{ child indices of } \overline{K}_h \}.$$

(4) The sparsely supported linear programming problem (LPR) is solved with indices $K_{h/2}$ and with $c^{h/2}$ and $\mu^{h/2}, \nu^{h/2}$ given by the quadrature rules (6) and (7).

(5) Repeat starting at (1), until a fine enough solution is computed.

No proof that this works, but "padding" is insurance

- Why might this work?
- 1. Sparsity
- 2. Padding
- 3. LP stability

- For the costs we compute, theory tells us that we have a map, so we expect a sparse solution.
- Intuition tells us that the sparsity should be monotone as we refine the grid. However, it's possible (maybe with highly oscillatory masses) that the support moves
- So we "pad" the support to a larger set.
- At each finer scale solve, if the support goes beyond the padding, we could go back. (But this has never happened).

LP Stability

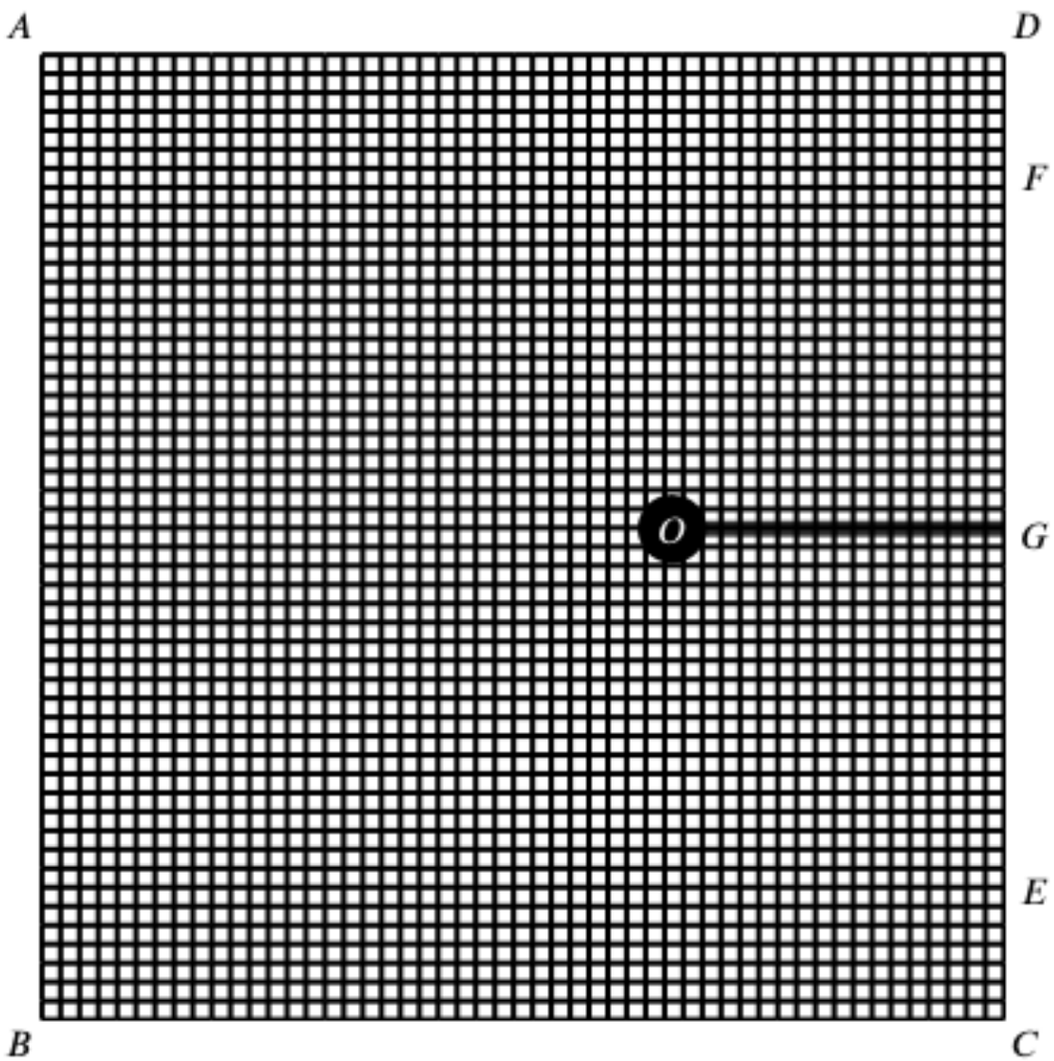
- We can "double" the LP, to find scales, and perturb with refined costs. Get a nearby LP

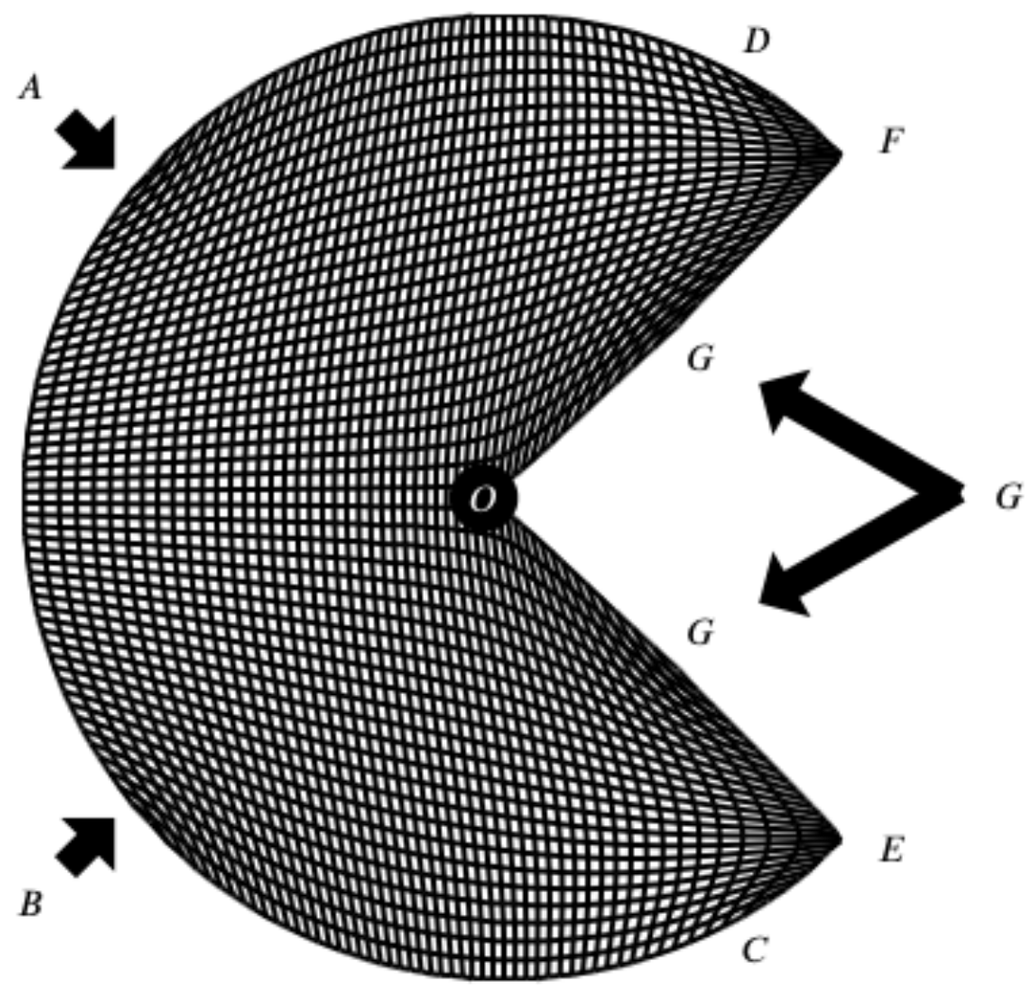
Theorem 6 (Stability of linear programming). *If the standard form linear program* (9) *has a unique optimal solution* x^*, *then the support of the optimal solution* $x^*(c)$ *is unchanged for nearby values* c. *If it has has a nondegenerate optimal basic solution* x^* , *then the support of the optimal solution* $x^*(b)$ *is unchanged for nearby values* b.

Lemma 8. *Consider the linear programming solution* $\pi^{h/2}$. *There is a perturbation of the cost function* $\hat{c}^{h/2}$, *for which the support of the solution,* $\hat{\pi}^{h/2}$, *of the perturbed problem is contained in the indices* $K_{h/2}$ *obtained using Step* (3) *from the coarse solution* π^h. *The perturbed cost can be made arbitrarily close to* $c^{h/2}$ *by taking* $h \to 0$.

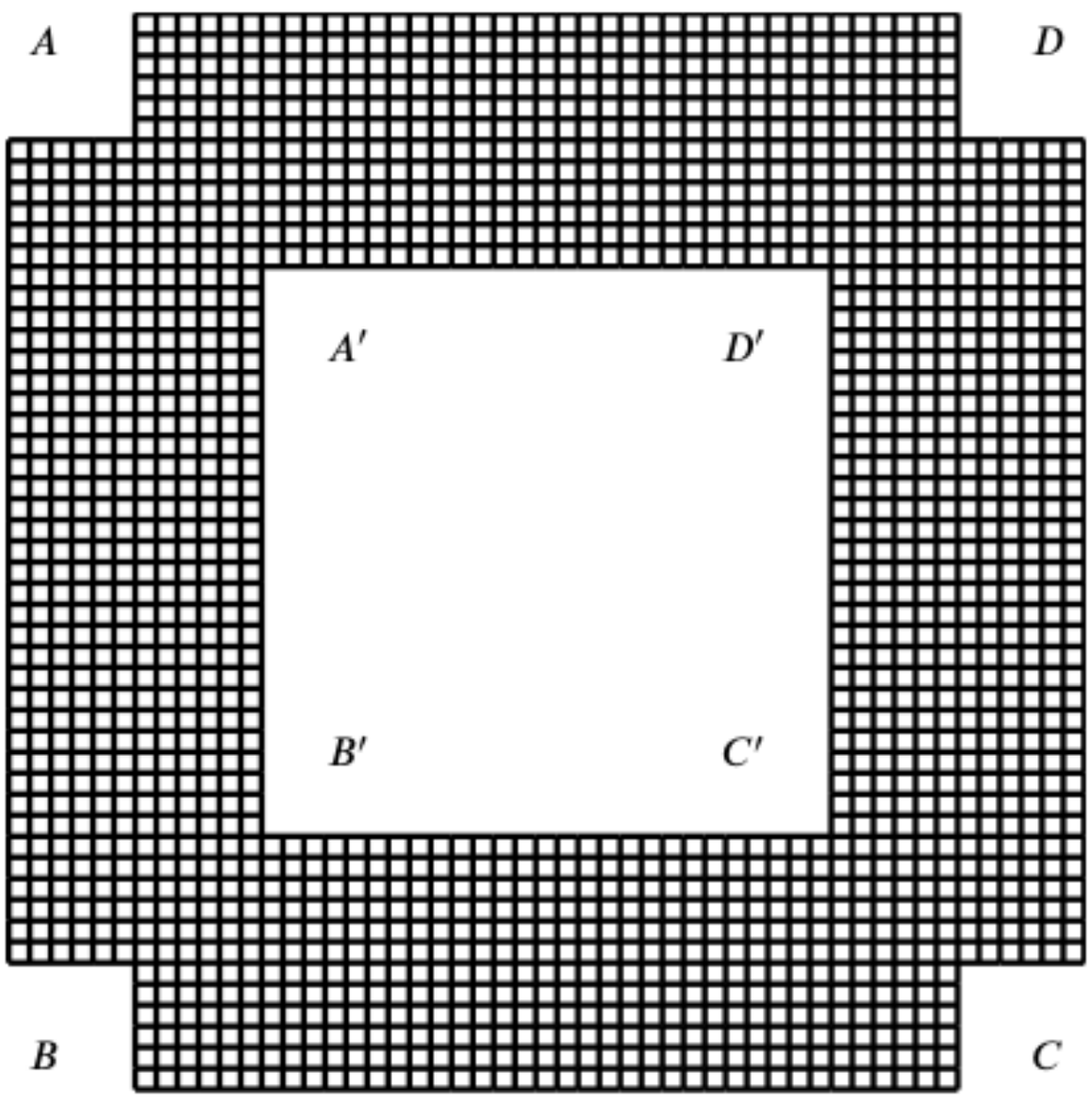
Numerical Results
OT Maps

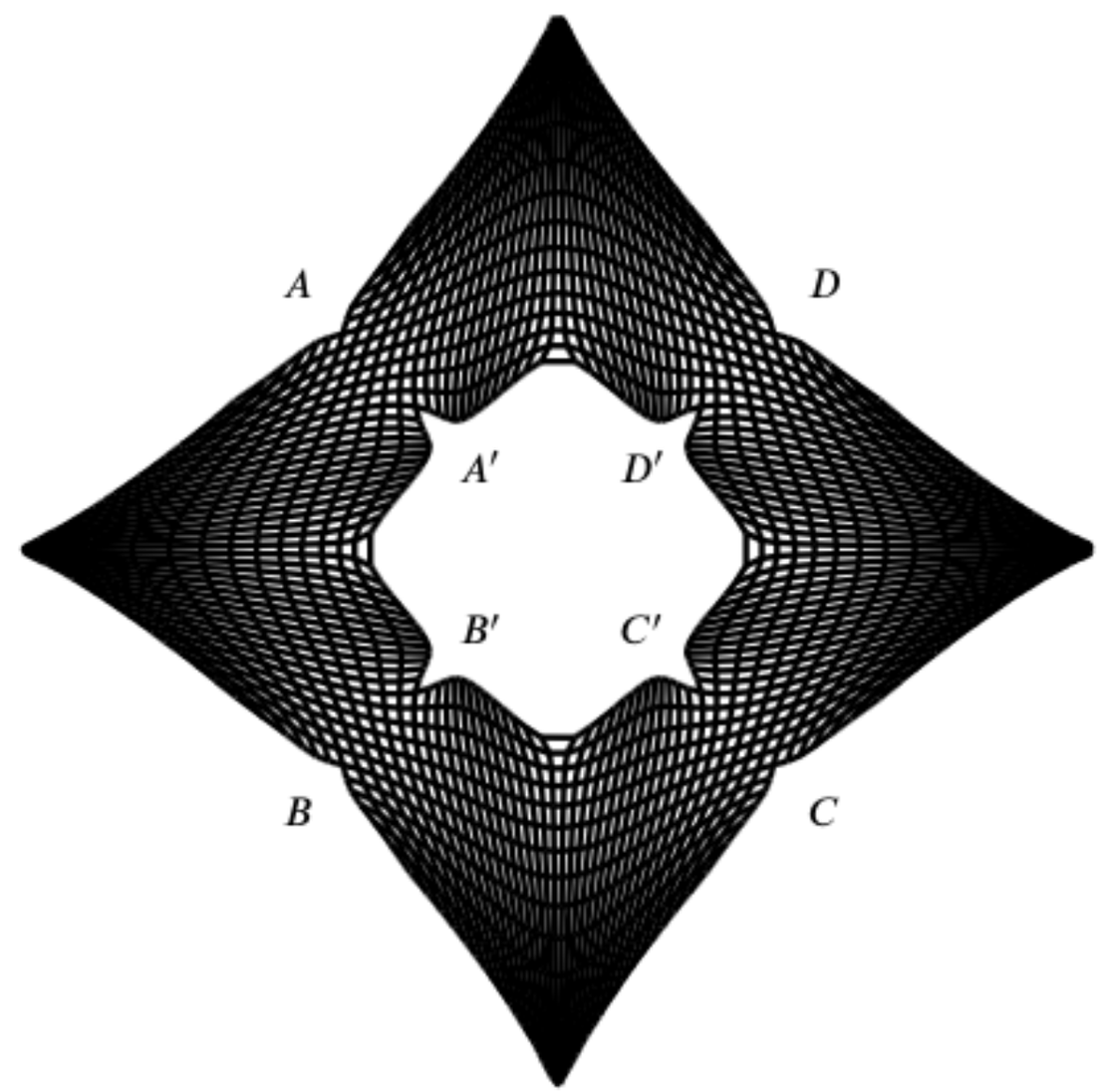
OT map using Quadratic cost: open mouth PacMan





OT map using Quadratic cost: from non-convex tile to "diamond"



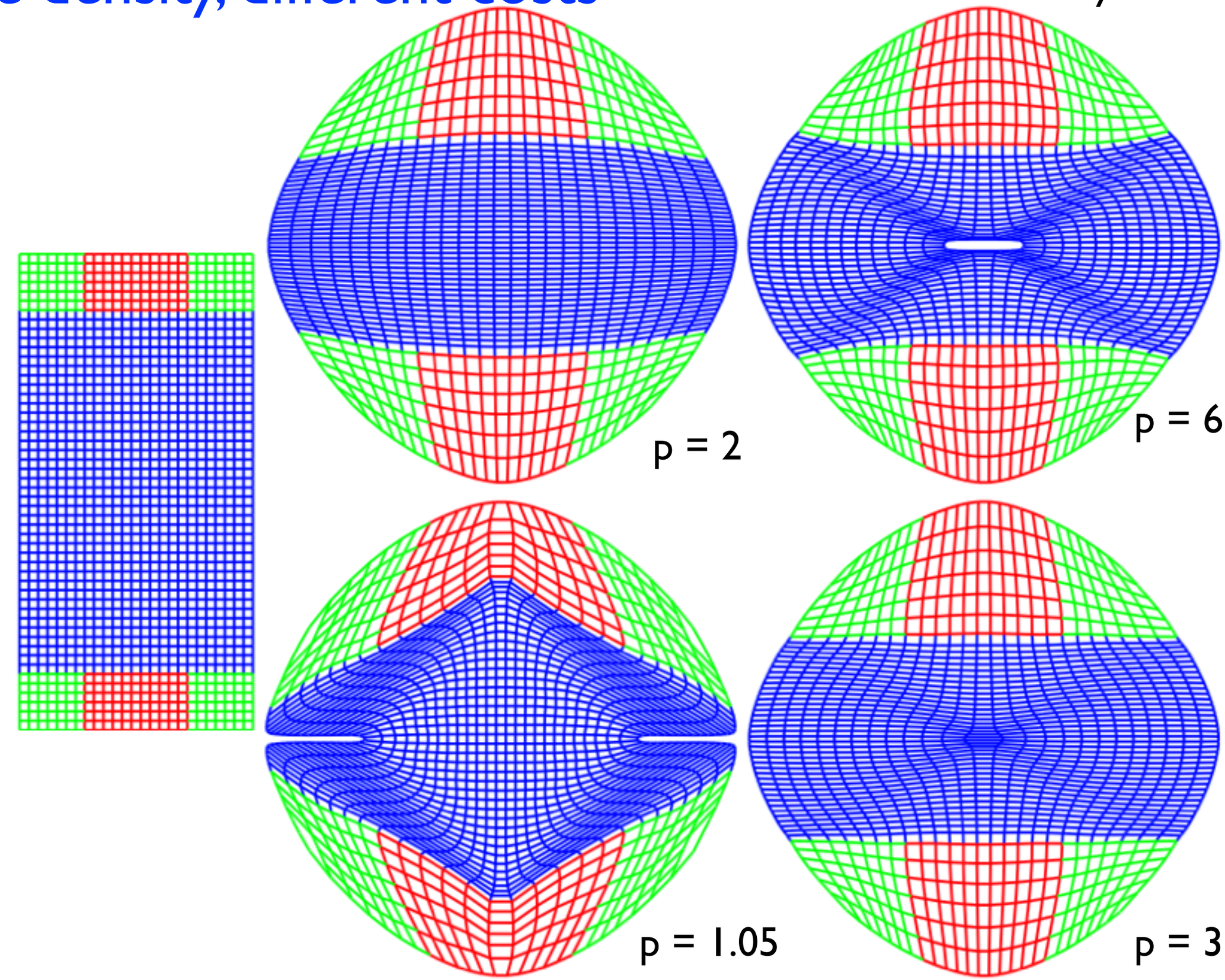


Computational Result.
variable density, different costs

Target: Blob with uniform density.

Costs: $d(x,y)^p$

Source: Rectangle piecewise constant density.



Numerical Results
Barycenter

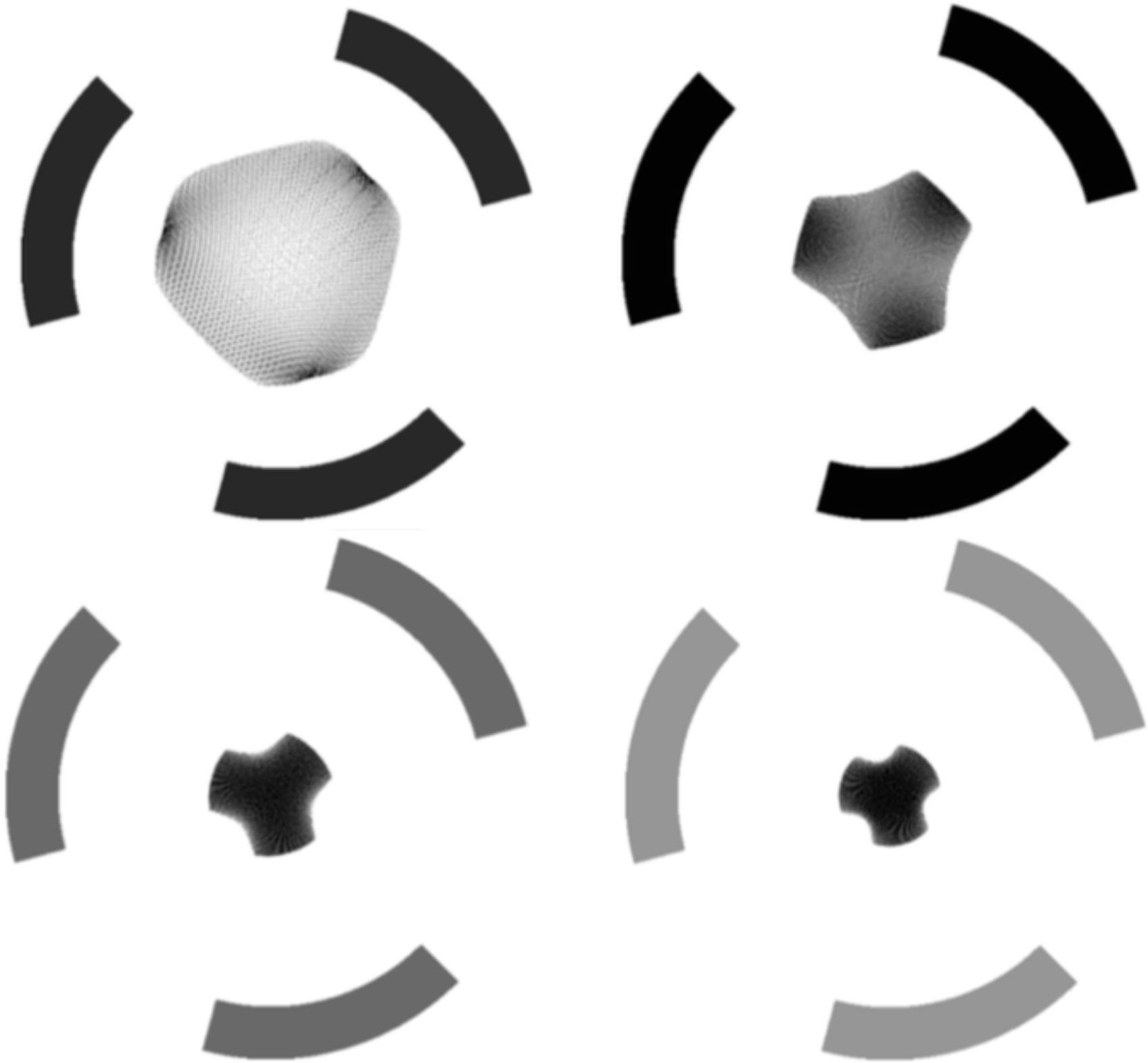
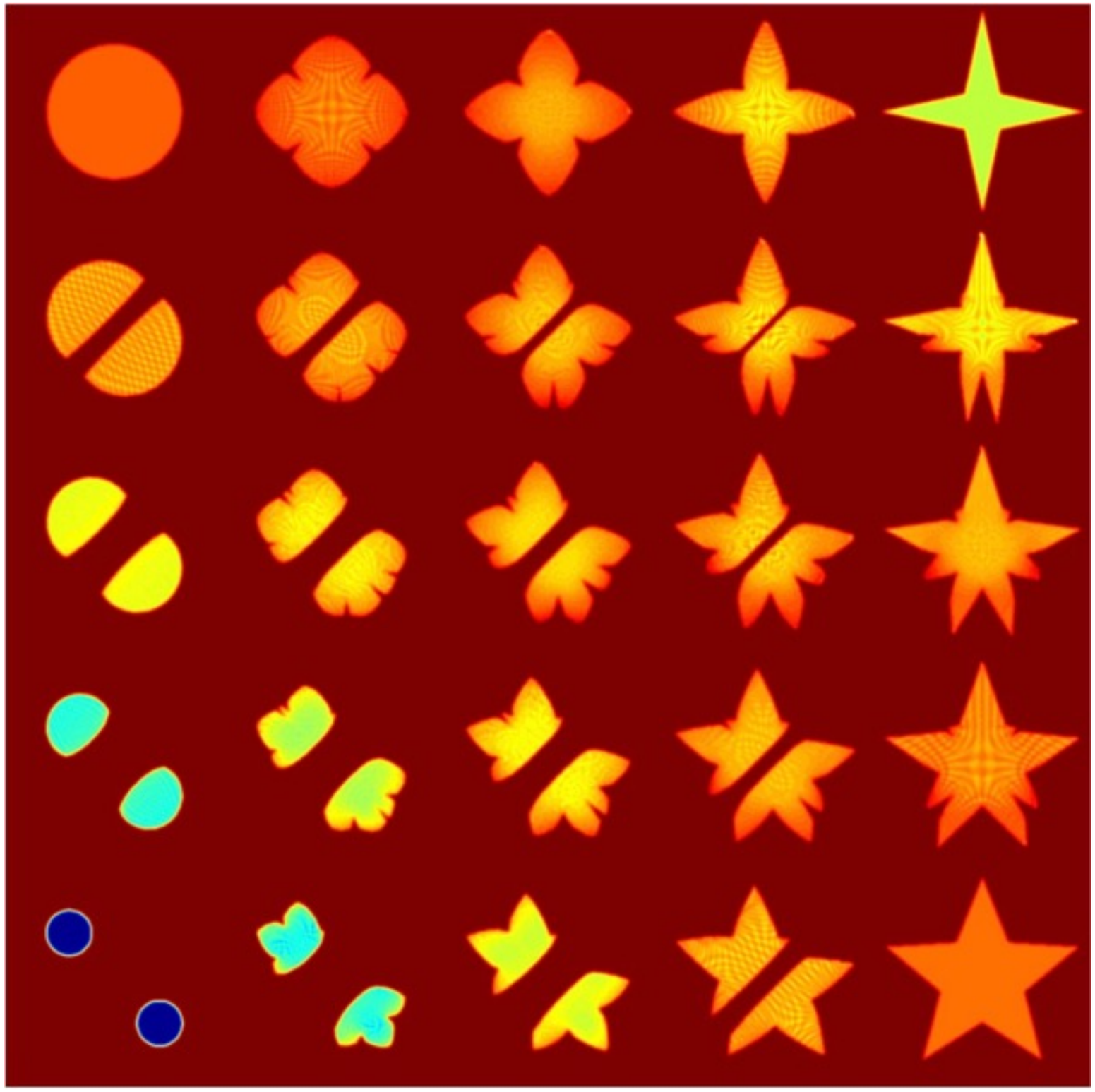


FIGURE 10. Barycenter of three sections of an annulus. From left to right and top to bottom: cost $c(x, y) = |x - y|^p$, $p = 1.05, 2, 5, 9$. Grid size 512^2.

Barycentre Problem: weighted average of measures, with distance given by Wasserstein

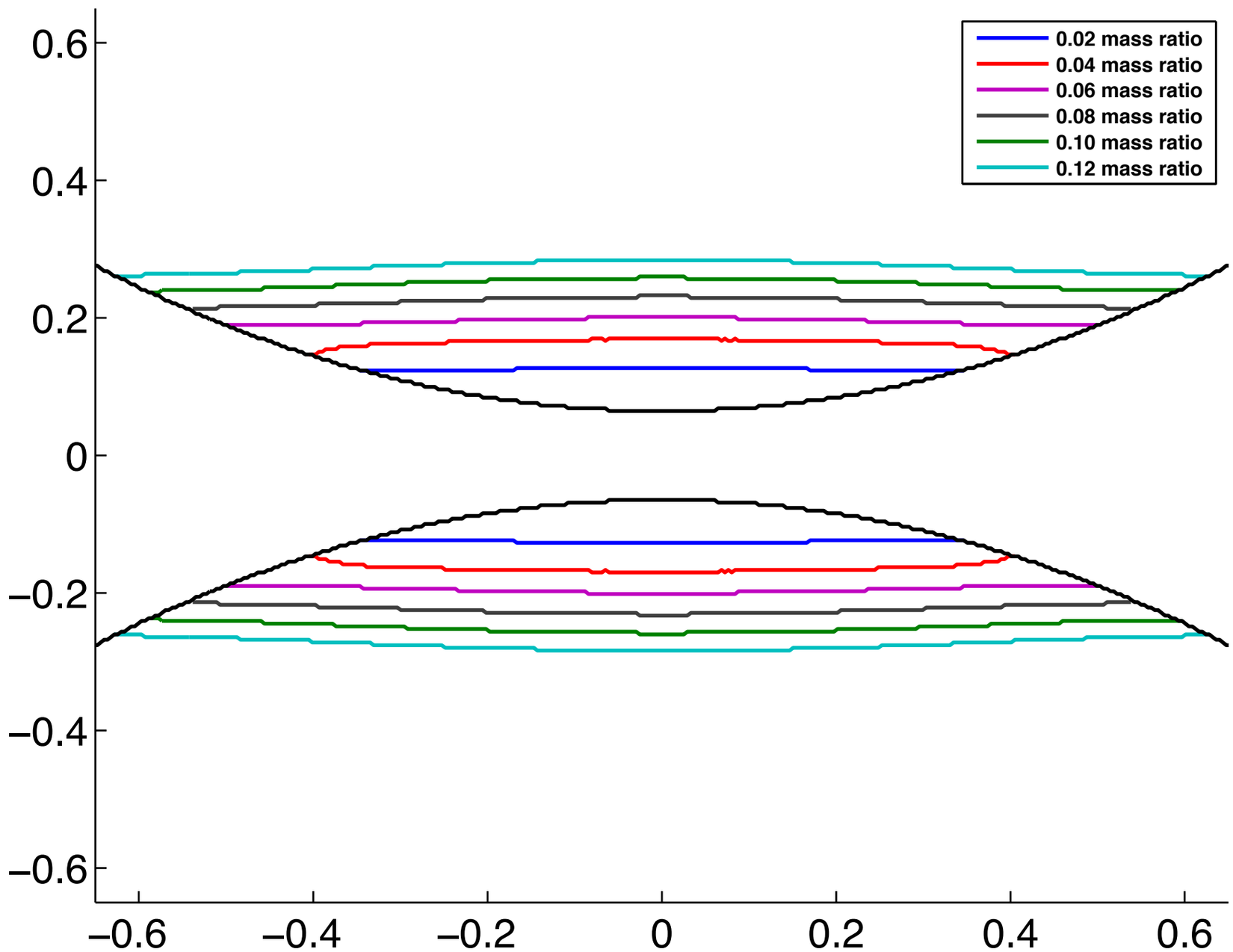


4 different given measures are positioned in the corners.

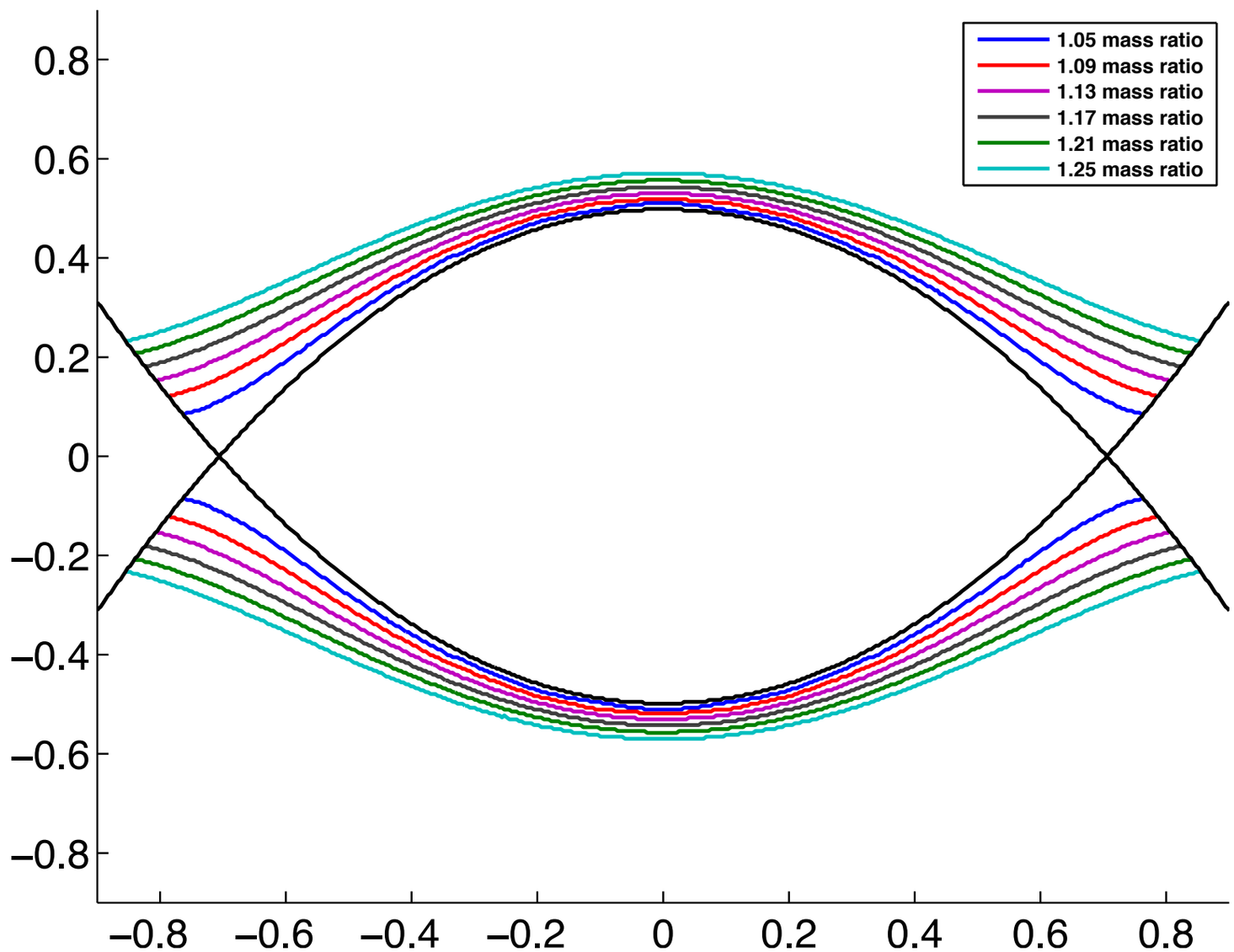
21 other measures are each the solution of a barycenter problem with weights given by position. (e.g. equal weights for the middle

Numerical Results
POT

Partial OT: no overlap.
note: free boundary is smooth



Partial OT: overlapping paraboloids: note: free boundary is smooth



Partial OT: overlapping squares: note: free boundary jumps off the corner