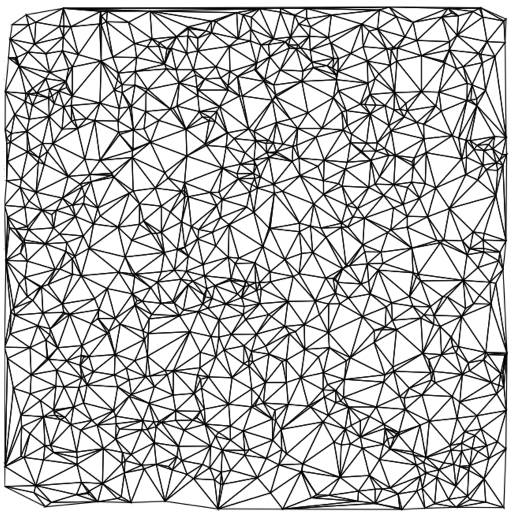
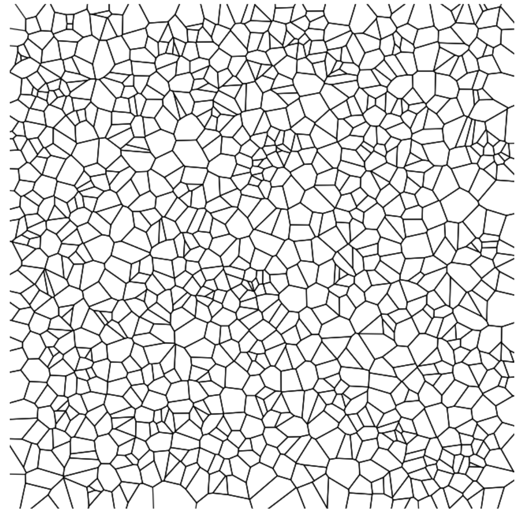
Geometry project

Voronoi Diagram We explore several ways of obtaining Voronoi diagrams from a given set of points. In order to compare performance, we are going to work on the same point set of size 2000.

- Method 1:
  First construct the Delaunay triangulation using the Bowyer–Watson algorithm in $O(Nlog(N))$ time. Then link the circumcenters of the neighboring triangles to obtain the edges of the Voronoi cells. Although efficient for obtaining an image, this method does not give us the cells themselves directly.

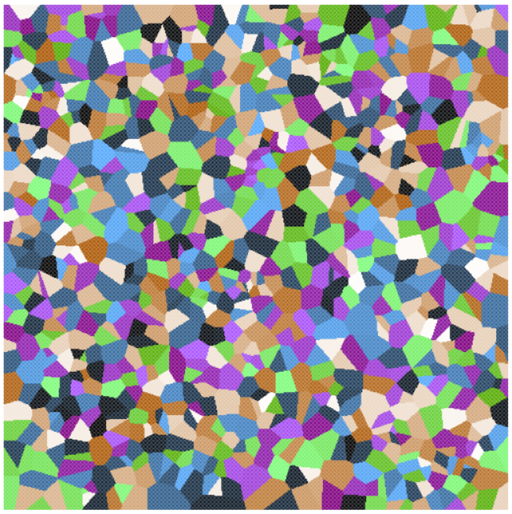


(a) Delaunay triangulation

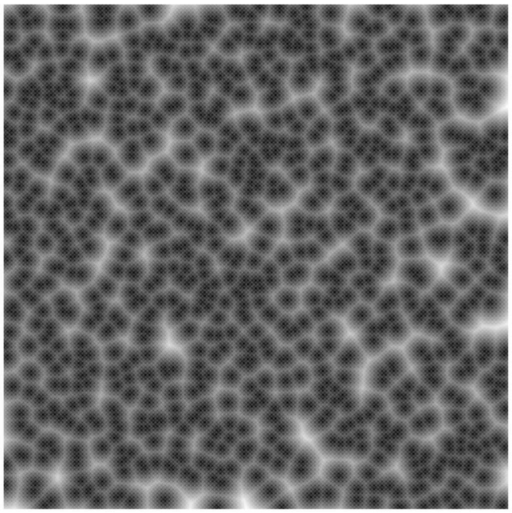


(b) Voronoi diagram

- Method 2:
  Use the Jump Flooding technique. Although it allows us to store more information, for example distances, at no additional cost, the method itself is not very efficient in comparison to the others.



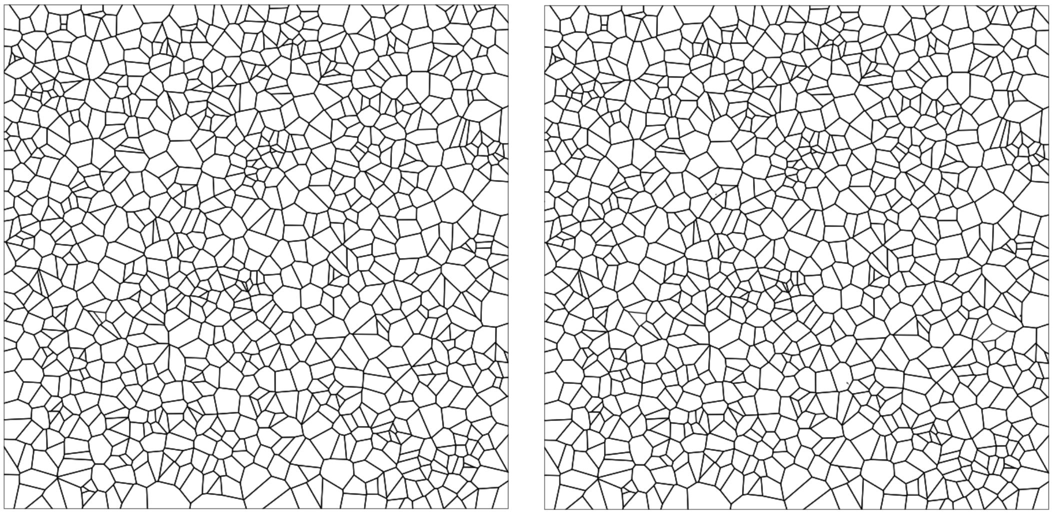
(a) Jump Flooding diagram



(b) Distance map

- Method 3:
  Obtain the Voronoi cell for each point P independently: first consider the whole area and then clip it by the bisector of PP_i for every other point P_i. The method is referred to as Voronoi parallel linear enumeration and will be used from now on for obtaining Voronoi diagrams.

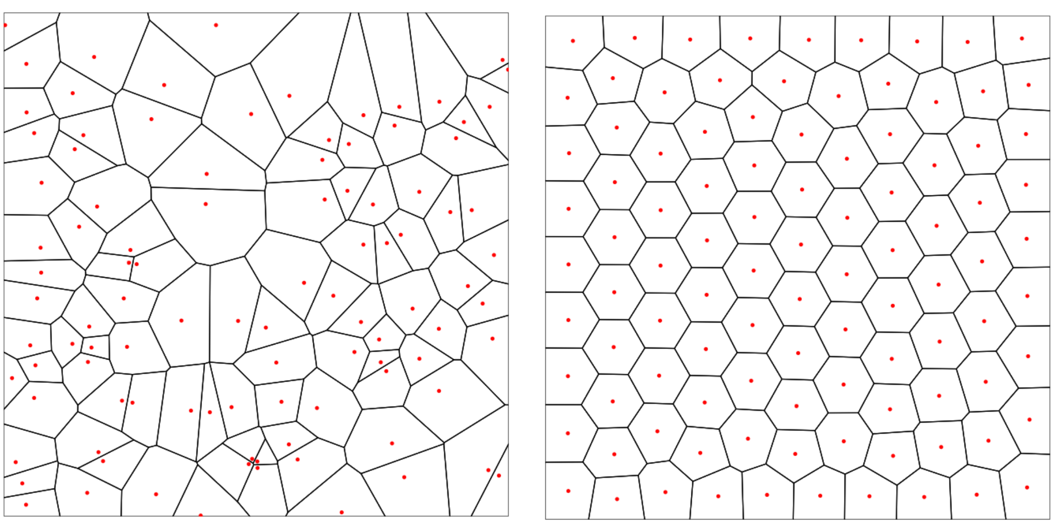
- Method 4:
  Apply the same principle as in the previous method but only for the k-nearest neighbors of every point. The observed speed up is around 3 times.



(a) runtime: 47 milliseconds (b) runtime: 17 milliseconds

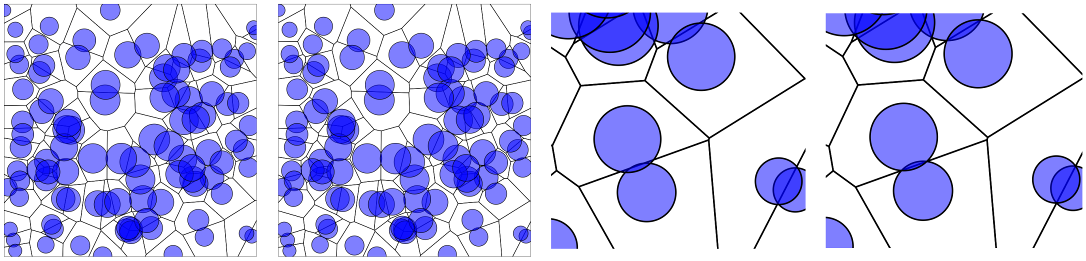
Runtime comparison with (on the right) and without (on the left) knn search

Lloyd's Iterations We implement a method to obtain a more even distribution from a randomly generated points. At each step the Voronoi diagram of the point set is built, then every point is replaced by the barycenter of its Voronoi cell. In order to obtain uniform distribution, the algorithm has to be repeated until convergence.



(a) Random point set (b) Distribution after 1000 iterations

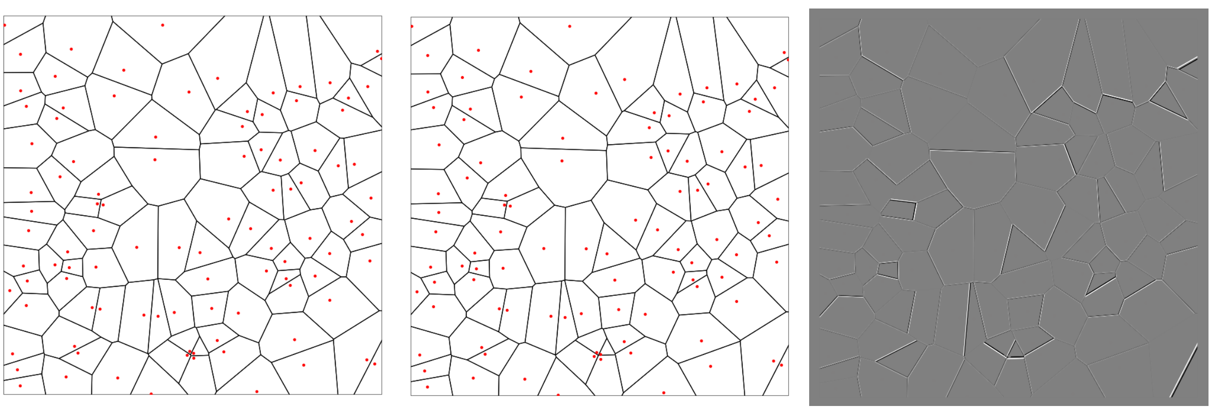
Power (Laguerre) Diagram The next step is to add weight to the points in order to influence the area of their cells. The algorithm is similar to the one used for Voronoi diagrams, however instead of taking the bisectors, we consider a perpendicular segment divider, based on the weights at both ends.



(a) Voronoi diagram (b) Laguerre diagram (c) Voronoi, zoom in (d) Laguerre, zoom in

For this particular example we have used weights following a Gaussian distributed with respect to the image center. On the figures above they are represented in blue - every "weight circle" C_i has a radius $\sqrt{w_i}$, where w_i is the weight assigned to its center P_i. These circles have been displayed over both the Voronoi diagram from before and the new Laguerre diagram, however as discussed the weights only impact the latter. Zooming in on both confirms that for the Laguerre diagram the cell edges coincide with the radical axis between the circles of the corresponding cells, which is not the case for the Voronoi one. Therefore, although looking identical, the two diagrams differ in a meaningful manner and their great visual similarities are only due to the specific point set and assigned weights.

Semi-discrete Optimal Transport We implement a method to find the optimal set of weights.



(a) initial diagram (b) after optimal transport (c) superposition of both

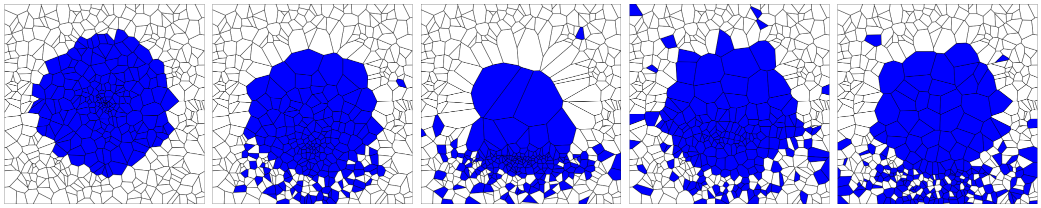
In the example above we try to go from uniform to Gaussian distributed weights. As before, the resulting diagrams look very similar, however overlapping them shows that they are in fact distinct.

Fluid simulation All of the algorithms describes so far allow us to create fluid simulations for a set of particles, represented by point coordinates with the respective physics quantities like mass or density. We proceed iteratively by evaluating the system after a time step and recording the resulting image frames.

- Particle simulation
  The first approach is to consider each particle individually and compute its trajectory, given the known formulas for gravity, pressure and viscosity.

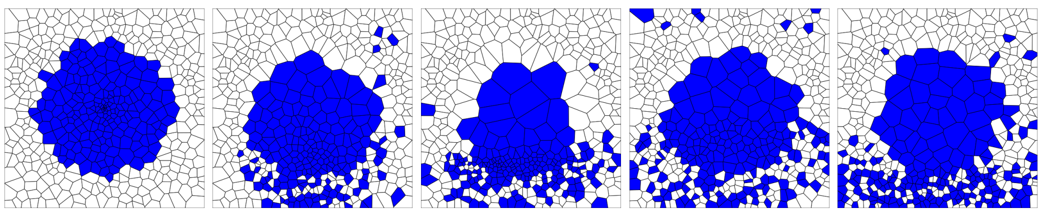
(a) frame 1 (b) frame 100 (c) frame 200

- Simulation using optimal transport and Laguerre diagrams
  Instead of considering the particles as points, we can use their Laguerre cells with weights such the cell areas are as evenly distributed as possible.



(a) frame 1 (b) frame 50 (c) frame 100 (d) frame 150 (e) frame 200

- Simulation using Lloyd's iterations and Voronoi diagrams
  Since the mass of each particle has been randomly generated, while the pressure remains constant, using optimal transport method as described does not produce the best visual results. Instead we can opt for a regular Voronoi diagram and apply Lloyd's iterations for a more even cell area distribution. It is important to note however that these resulting images are less accurate to reality.



(a) frame 1 (b) frame 50 (c) frame 100 (d) frame 150 (e) frame 200

Finally, for a smoother image, we can display only the "weight circles" instead of the entire Voronoi cell.

(a) frame 1 (b) frame 50 (c) frame 100 (d) frame 150 (e) frame 200

The entire fluid simulation recordings can be found in `/_results`. The same folder also contains a full record of all obtained images and videos, as well as their respective runtimes.