Computational Geoemtry

Programming Assignments II
Instructors: Ioannis Z. Emiris and Anna Karasoulou
Spring 2020

Exercise 1. Compute Voronoi diagrams of different sets of vertices of your choice using the routine Voronoi (and its companion voronoi plot 2d for visualization) from the module scipy.spatial. Plot your results.

Exercise 2. Using the routine Delaunay in the module scipy.spatial compute the Delaunay triangulation of different sets of vertices of your choice and plot your results.

Exercise 3. Compute the shortest path of different set of vertices of your choice in a triangulation. By a path in this setting, we mean a chain of edges of this triangulation. Use the methods in the package scipy.sparse.csgraph.

Exercise 4. Experiment yourself with the *.encloses_point* and *.encloses* methods of the sympy.geometry module usingf polygons or circles to check if they contain certain points of your choice. Do the same with *contains_point* or *contains_points* from the Path class from the libraries of matplotlib.path.

Exercise 5. The problem of finding the Voronoi cell that contains a given location is equivalent to the search for the nearest neighbor. We can always perform this search with a brute force algorithm, but in general there are more elegant and less complex approaches to this problem like the kd-trees. In the scipy use the class KDTree to perform some experiments of your choice.