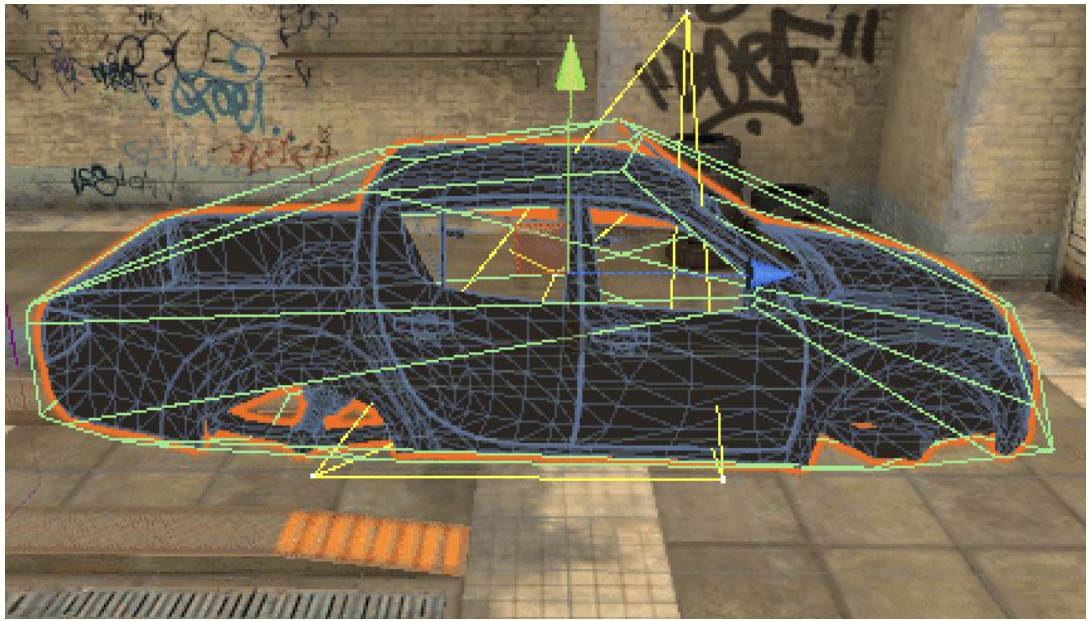


Computational Geometry: Triangulation

Analysis and Design of Advance Algorithms

Ing. Luis Salomon Flores Ugalde
Luis_SFU@tec.mx

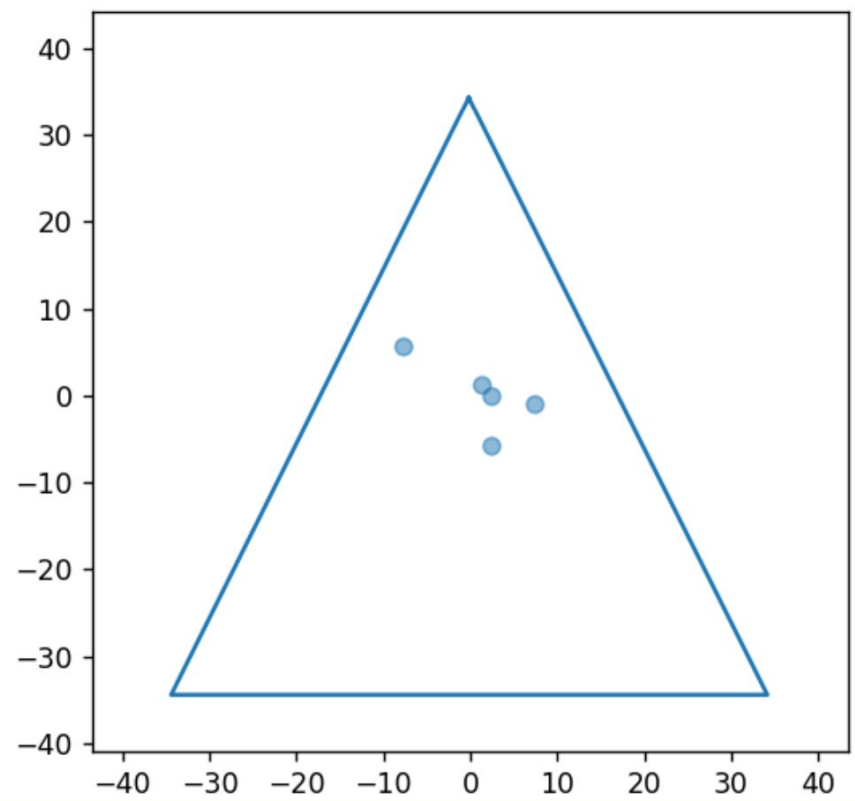
Translation of
Dra. Valentina Navárez Terán
presentation

Problem 1: Points and triangles

The problem:
With a set of points with coordinates x, y

How would you create a triangle so that all the points are found inside?

The points given are **NOT** part of any of the vertices of the triangle

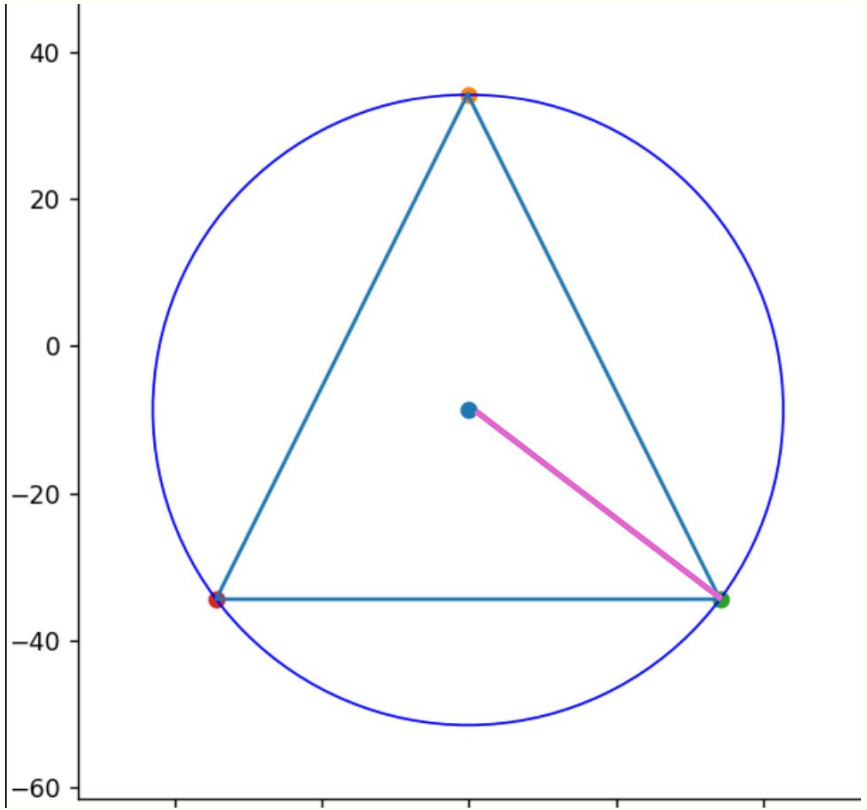


Problem 2: The circumscribed circle

The problem:
Given 3 points.

How would you create a circle so that the 3 points are PART of the circumference?

Note that the **center** of the circle **could** be **outside** of the triangle formed by the 3 points.

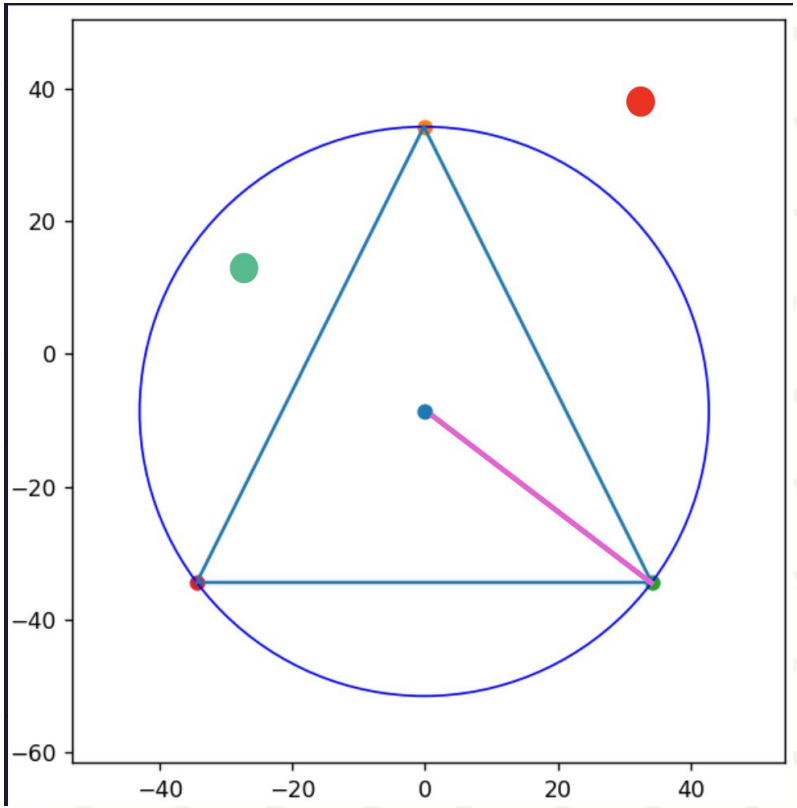


Problem 3: Points inside of circles.

The problem:
Given several triangles, each one with its respective circle (calculated from problem 2) and a point.

How would you create a list of triangles whose circles contain a given point?

Example: the green point is inside the circle, the red one is not.

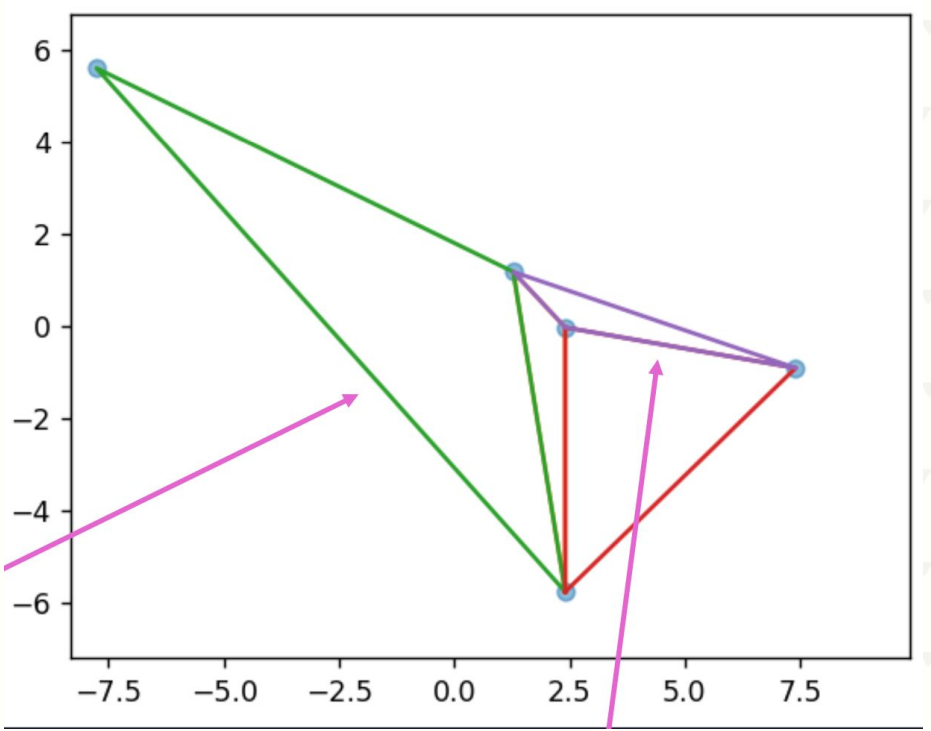


Problem 4: nonshared sides.

The problem:
Given several triangles, each one formed by 3 segments...

How would you identify which of the segments are NOT shared by other triangles?

Create a list of segments that meet this characteristic.



Not shared.

Shared.

Problem 5: Graphics

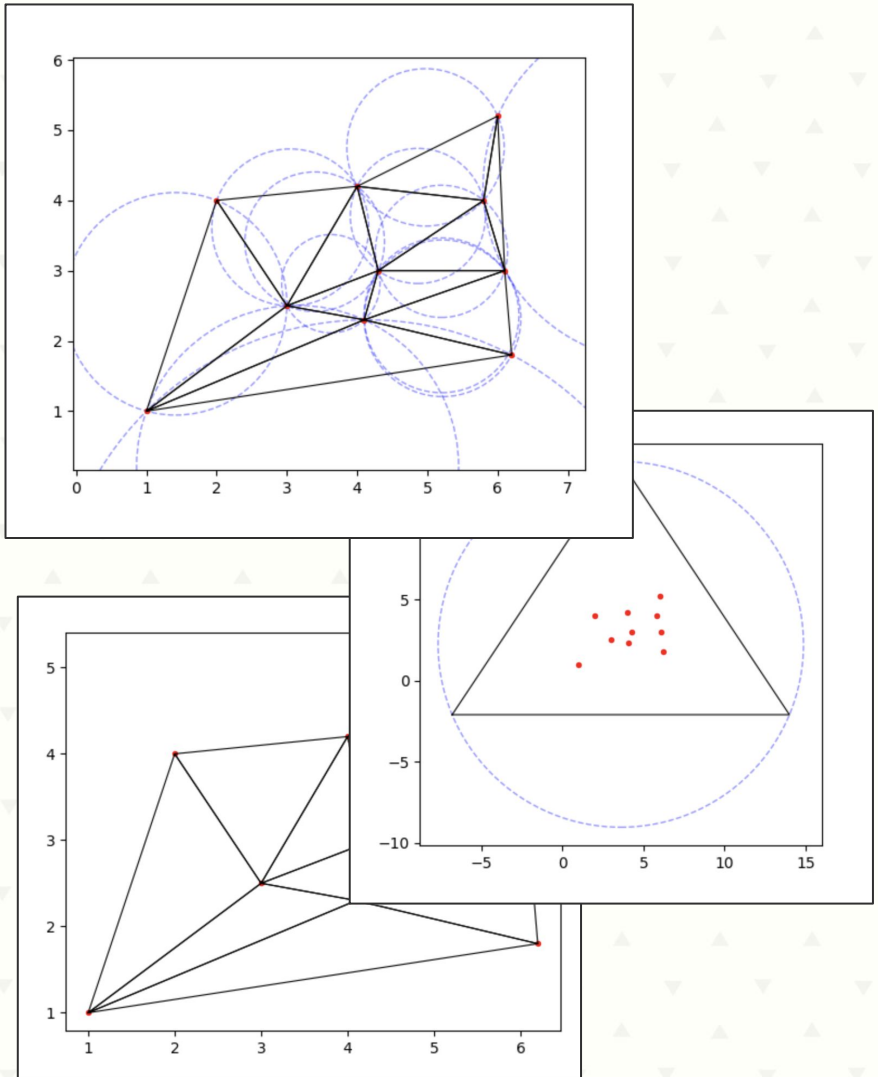
The problem:
How do you graph the results?

Create a function to help you easily visualize the results and save them as images.

For example in matplotlib:

```
For points: scatter
For lines: plot
For circles: circle, gca, add_path

For ratio of graph and limits
plt.aces().set_aspect('equal')
plt.ylim(-150, 150)
plt.xlim(-150, 150)
To save:
savefig
```



Problem 6: Delaunay Triangulation

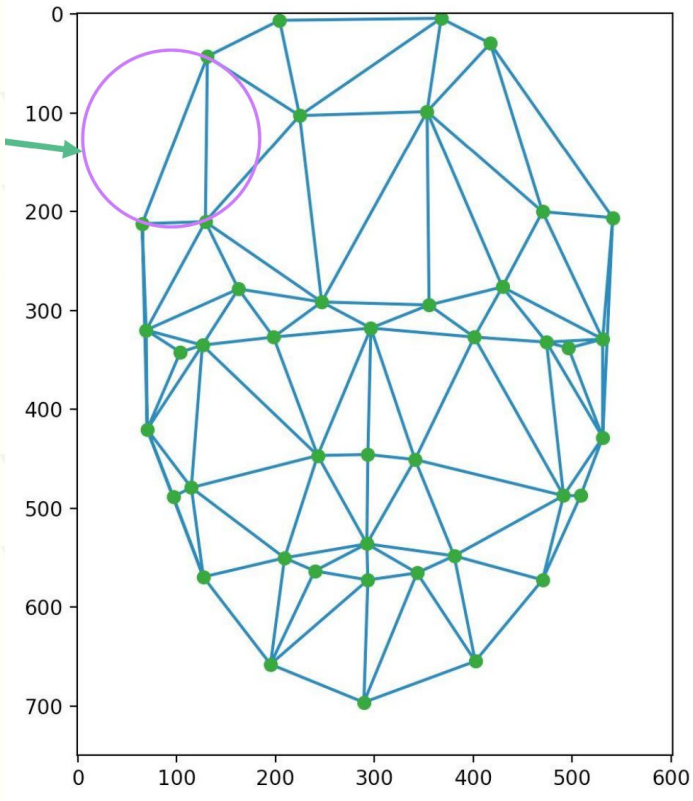
The problem:
Given a set of points:

The delaunay triangulation is a set of triangles such that the circumscribed circle of any of them contains no points.

How is it calculated?

Bowyer-Watson Algorithm, which uses the answers of the prior problems.

This holds true for all triangles



Problem 6: Delaunay Triangulation
Bowyer-Watson

```
big_triangle = a triangle that contains all the points
tris = list of triangles, initialized with only big_triangle

for each point p:
      bad_tris = list of triangles in tris whose circumcircle
      contains p

      uniques = list of edges from triangles in bad_tris that are
      not shared by any other triangle in bad_tris

      remove all bad_tris from tris

      for each edge in uniques:
            new_tri = triangle formed by connecting point p with the
            two vertices of edge

            Add new_tri to tris

remove from tris all triangles that include any vertex from
big_triangle

return tris
```

Problem 1

Problem 2 and 3

Problem 4

Problem 5: Graph the results to check.