GIS - Task #8567

Delaunay 2D and 3D triangulation using catchment points and their elevations

28/06/2019 14:30 - Debojyoti Mallick

Status:	New	**Start date:**	28/06/2019
Priority:	Normal	**Due date:**	
Assignee:	Debojyoti Mallick	**% Done:**	30%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour

Description

The purpose of this task is to understand the possibilities of Delaunay triangulation for vector outputs and to choose which method produces the least number of anomalies.

History

#1 - 28/06/2019 16:38 - Debojyoti Mallick

- File Delaunay_QbBQt.png added

#2 - 01/07/2019 11:38 - Debojyoti Mallick

Debojyoti Mallick wrote:

> The purpose of this task is to understand the possibilities of Delaunay triangulation for vector outputs and to choose which method produces the least number of anomalies.

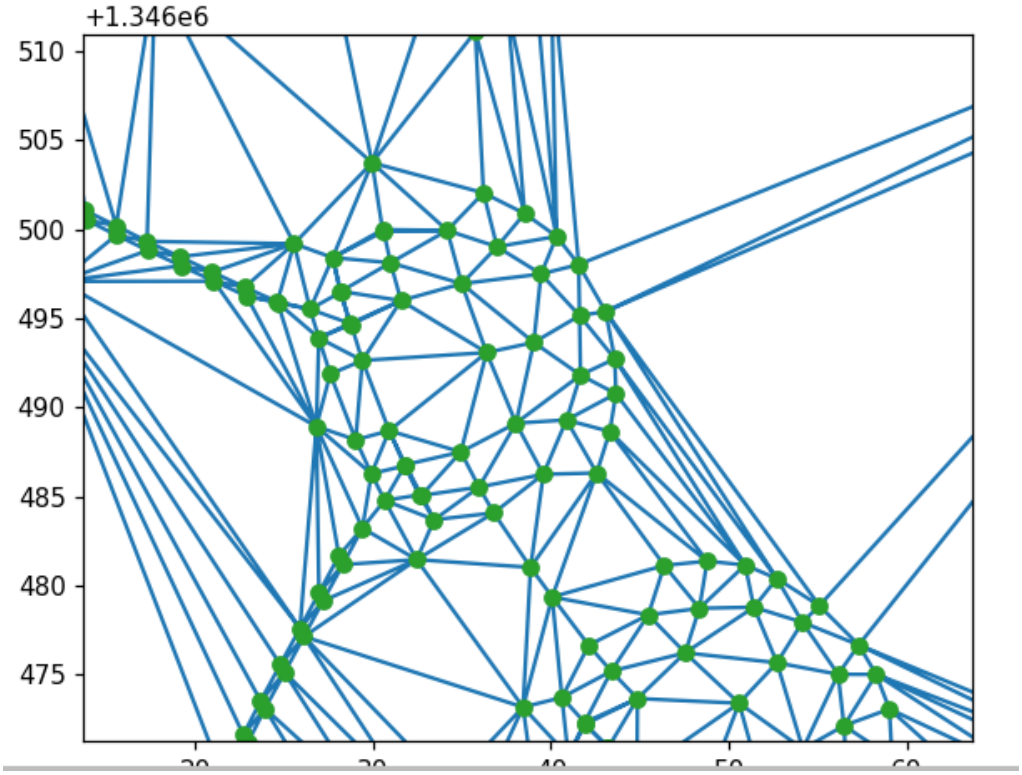
Found a matching QHull method for Delaunay Triangulation:

Matches Delaunay Methods in GRASS 7.* GIS , scipy.spatial and Cloud Compare.

Script reference of tests done using Jupyter labs.

```
%matplotlib widget

import matplotlib as mpl
import matplotlib.pyplot as plt
import mpl_toolkits.mplot3d as a3
import numpy as np
import scipy as sp
from scipy import spatial as sp_spatial
from shapely.geometry import mapping

from scipy.spatial import Delaunay

import geopandas
gdf = geopandas.read_file('data/Catchment_points.gpkg').to_crs(epsg=3857)

gdf.head()

gdf.plot(figsize=(10, 5), alpha=0.3)

points = list(gdf.xy)

tri = Delaunay(points, qhull_options="QbB Qt")

plt.triplot(gdf.x, gdf.y, tri.simplices.copy())
plt.plot(gdf.x, gdf.y, 'o' )
plt.show()
```

Results using the script above :



QHull method used:

QbB - scale the input to fit the unit cube
After scaling with option 'QbB', the lower bound will be -0.5 and the upper bound +0.5 in all dimensions. For different bounds change qh_DEFAULTbox in user.h (0.5 is best for Geomview).

For Delaunay and Voronoi diagrams, scaling happens after projection to the paraboloid. Under precise arithmetic, scaling does not change the topology of the convex hull. Scaling may reduce precision errors if coordinate values vary widely.

Qt - triangulated output
By default, qhull merges facets to handle precision errors. This produces non-simplicial facets (e.g., the convex hull of a cube has 6 square facets). Each facet is non-simplicial because it has four vertices.

Use option 'Qt' to triangulate all non-simplicial facets before generating results. Alternatively, use joggled input ('QJ') to prevent non-simplical facets. Unless 'Pp' is set, qhull produces a warning if 'QJ' and 'Qt' are used together.

For Delaunay triangulations (qdelaunay), triangulation occurs after lifting the input sites to a paraboloid and computing the convex hull.

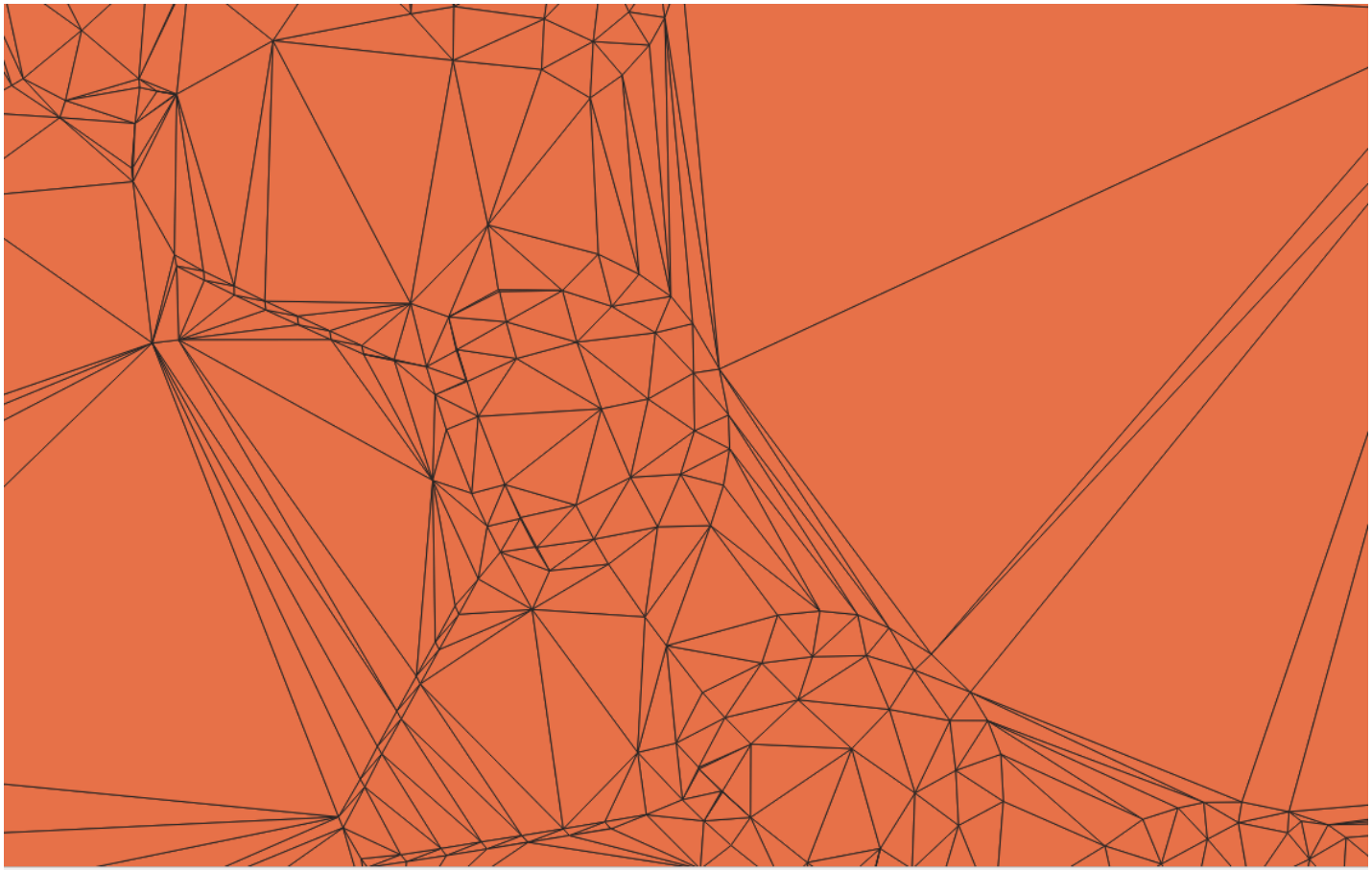
Option 'Qt' is deprecated for Voronoi diagrams (qvoronoi). It triangulates cospherical points, leading to duplicated Voronoi vertices.

Option 'Qt' may produce degenerate facets with zero area.

Facet area and hull volumes may differ with and without 'Qt'. The triangulations are different and different triangles may be ignored due to precision errors.

With sufficient merging, the ridges of a non-simplicial facet may share more than two neighboring facets. If so, their triangulation ('Qt') will fail since two facets have the same vertex set.

You can refer to the following image for triangulation results using GRASS:



#3 - 01/07/2019 11:38 - Debojyoti Mallick

- % Done changed from 0 to 30

#4 - 03/07/2019 11:44 - Debojyoti Mallick

- Assignee set to Debojyoti Mallick

Files

Figure 1 (1).png	106 KB	28/06/2019	Debojyoti Mallick
Delaunay_GRASS.png	183 KB	28/06/2019	Debojyoti Mallick
Delaunay_QbBQt.png	97.3 KB	28/06/2019	Debojyoti Mallick