Triangulations / Quadrangulations

Jean-François Remacle[1] and Christophe Geuzaine[2]

[1] Université catholique de Louvain (UCLouvain)

[2] Université de Liège (ULiege)

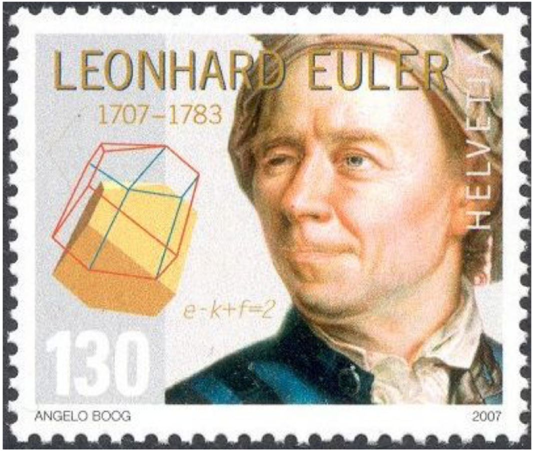
http://www.gmsh.info

November 23, 2020

Algebraic topology for meshes

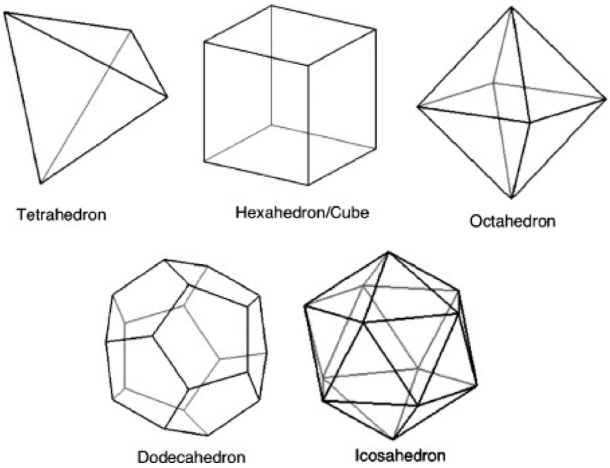
Delaunay triangulations in the plane

Euler's second most famous result



Platonic solids

There exist exactly 5 "ideal" polyedra:



There a are exactly 5 platonic solids

- Consider a polyhedron with n vertices, n_e edges, n_f planar facets. Euler Formula is written

$$n - n_e + n_f = 2. \tag{1}$$

There a are exactly 5 platonic solids

- Consider a polyhedron with n vertices, n_e edges, n_f planar facets. Euler Formula is written

$$n - n_e + n_f = 2. \quad (1)$$

- Let m denote the number of edges and vertices of each facet and k the degree of each vertex i.e. the number of facets adjacent to the vertex.

There a are exactly 5 platonic solids

- Consider a polyhedron with n vertices, n_e edges, n_f planar facets. Euler Formula is written

$$n - n_e + n_f = 2. \tag{1}$$

- Let m denote the number of edges and vertices of each facet and k the degree of each vertex i.e. the number of facets adjacent to the vertex.
- Each vertex has k adjacent faces and each face has m vertices. This implies that

$$mn_f = kn \quad \rightarrow \quad n_f = \frac{kn}{m}. \tag{2}$$

There a are exactly 5 platonic solids

- Consider a polyhedron with n vertices, n_e edges, n_f planar facets. Euler Formula is written

$$n - n_e + n_f = 2. \tag{1}$$

- Let m denote the number of edges and vertices of each facet and k the degree of each vertex i.e. the number of facets adjacent to the vertex.
- Each vertex has k adjacent faces and each face has m vertices. This implies that

$$mn_f = kn \quad \to \quad n_f = \frac{kn}{m}. \tag{2}$$

- Each edge has 2 adjacent faces and each face has m edges. This implies

$$mn_f = 2n_e \quad \to \quad n_e = \frac{mn_f}{2} = \frac{kn}{2}. \tag{3}$$

- Putting (1), (2) and (3) together gives

$$n\left(1 + \left(\frac{k}{m} - \frac{k}{2}\right)\right) = 2.$$

There a are exactly 5 platonic solids

- We can expand

$$n\left(1+\left(\frac{k}{m}-\frac{k}{2}\right)\right)=2.$$

  into

$$(2m+2k-mk)n=4.$$

There a are exactly 5 platonic solids

- We can expand

$$n\left(1+\left(\frac{k}{m}-\frac{k}{2}\right)\right)=2.$$

into

$$(2m+2k-mk)n=4.$$

- Since $n>0$ and $m>0$, we must have

$$2m+2k-mk>0.$$

There a are exactly 5 platonic solids

- We can expand
$$n\left(1+\left(\frac{k}{m}-\frac{k}{2}\right)\right)=2.$$
into
$$(2m+2k-mk)n=4.$$
- Since $n>0$ and $m>0$, we must have
$$2m+2k-mk>0.$$
- Since
$$2m+2k-mk=-(k-2)(m-2)+4>0$$
the condition is transformed into
$$(k-2)(m-2)<4.$$

There a are exactly 5 platonic solids

- We can expand

$$n\left(1+\left(\frac{k}{m}-\frac{k}{2}\right)\right)=2.$$

into

$$(2m+2k-mk)n=4.$$

- Since $n>0$ and $m>0$, we must have

$$2m+2k-mk>0.$$

- Since

$$2m+2k-mk=-(k-2)(m-2)+4>0$$

the condition is transformed into

$$(k-2)(m-2)<4.$$

- Since $k\geq 3$ and $m\geq 3$, the only possible values for (m,k) are $(3,3)$, $(4,3)$, $(5,3)$, $(3,4)$ and $(3,5)$.

There a are exactly 5 platonic solids

$$n\left(1+\left(\frac{k}{m}-\frac{k}{2}\right)\right)=2 \quad , \quad n_f=\frac{kn}{m}.$$

- Tetrahedron : $(m,k)=(3,3) \quad \rightarrow \quad n=4 \quad , \quad n_f=4.$

There a are exactly 5 platonic solids

$$n\left(1+\left(\frac{k}{m}-\frac{k}{2}\right)\right)=2 \quad , \quad n_f=\frac{kn}{m}.$$

- Tetrahedron : $(m,k)=(3,3) \quad \rightarrow \quad n=4 \quad , \quad n_f=4.$
- Hexahedron : $(m,k)=(4,3) \quad \rightarrow \quad n=8 \quad , \quad n_f=6.$

There a are exactly 5 platonic solids

$$n\left(1+\left(\frac{k}{m}-\frac{k}{2}\right)\right)=2 \quad , \quad n_f=\frac{kn}{m}.$$

- Tetrahedron : $(m,k)=(3,3) \quad \rightarrow \quad n=4 \quad , \quad n_f=4.$
- Hexahedron : $(m,k)=(4,3) \quad \rightarrow \quad n=8 \quad , \quad n_f=6.$
- Octahedron : $(m,k)=(3,4) \quad \rightarrow \quad n=6 \quad , \quad n_f=8.$

There a are exactly 5 platonic solids

$$n\left(1+\left(\frac{k}{m}-\frac{k}{2}\right)\right)=2 \quad , \quad n_f=\frac{kn}{m}.$$

- Tetrahedron : $(m,k)=(3,3) \quad \rightarrow \quad n=4 \quad , \quad n_f=4.$
- Hexahedron : $(m,k)=(4,3) \quad \rightarrow \quad n=8 \quad , \quad n_f=6.$
- Octahedron : $(m,k)=(3,4) \quad \rightarrow \quad n=6 \quad , \quad n_f=8.$
- Dodecahedron : $(m,k)=(5,3) \quad \rightarrow \quad n=20 \quad , \quad n_f=12.$

There a are exactly 5 platonic solids

$$n\left(1+\left(\frac{k}{m}-\frac{k}{2}\right)\right)=2 \quad , \quad n_f=\frac{kn}{m}.$$

- Tetrahedron : $(m,k)=(3,3) \quad \rightarrow \quad n=4 \quad , \quad n_f=4.$
- Hexahedron : $(m,k)=(4,3) \quad \rightarrow \quad n=8 \quad , \quad n_f=6.$
- Octahedron : $(m,k)=(3,4) \quad \rightarrow \quad n=6 \quad , \quad n_f=8.$
- Dodecahedron : $(m,k)=(5,3) \quad \rightarrow \quad n=20 \quad , \quad n_f=12.$
- Icosahedron : $(m,k)=(3,5) \quad \rightarrow \quad n=12 \quad , \quad n_f=20.$

Euler-Poincare Characteristic

- The topology of a 3D surface S can be described by a topological invariant that is its Euler-Poincare characteristic χ.

Euler-Poincare Characteristic

- The topology of a 3D surface S can be described by a topological invariant that is its Euler-Poincare characteristic χ.
  - Two surfaces S_1 and S_2 with the same χ are topologically equivalent: it is possible to deform S_1 onto S_2 smoothly.

Euler-Poincare Characteristic

- The topology of a 3D surface S can be described by a topological invariant that is its Euler-Poincare characteristic χ.
  - Two surfaces S_1 and S_2 with the same χ are topologically equivalent: it is possible to deform S_1 onto S_2 smoothly.
  - Assume that our surface is a sphere with n_b holes and n_h handles. We have :

$$\chi = 2 - n_h - 2n_h$$

Euler-Poincare Characteristic

- The topology of a 3D surface S can be described by a topological invariant that is its Euler-Poincare characteristic χ.
  - Two surfaces S_1 and S_2 with the same χ are topologically equivalent: it is possible to deform S_1 onto S_2 smoothly.
  - Assume that our surface is a sphere with n_b holes and n_h handles. We have :

$$\chi = 2 - n_h - 2n_h$$

  - A disk can be seen (topologically) as a sphere with one hole in it so $\chi = 2 - 1 = 1$.

Euler-Poincare Characteristic

- The topology of a 3D surface S can be described by a topological invariant that is its Euler-Poincare characteristic χ.
  - Two surfaces S_1 and S_2 with the same χ are topologically equivalent: it is possible to deform S_1 onto S_2 smoothly.
  - Assume that our surface is a sphere with n_b holes and n_h handles. We have :

$$\chi = 2 - n_h - 2n_h$$

  - A disk can be seen (topologically) as a sphere with one hole in it so $\chi = 2 - 1 = 1$.
  - The surface of a cylinder can be seen (topologically) as a sphere with two holes in it so $\chi = 2 - 2 = 0$.

Euler-Poincare Characteristic

- The topology of a 3D surface S can be described by a topological invariant that is its Euler-Poincare characteristic χ.
  - Two surfaces S_1 and S_2 with the same χ are topologically equivalent: it is possible to deform S_1 onto S_2 smoothly.
  - Assume that our surface is a sphere with n_b holes and n_h handles. We have :

$$\chi = 2 - n_h - 2n_h$$

  - A disk can be seen (topologically) as a sphere with one hole in it so $\chi = 2 - 1 = 1$.
  - The surface of a cylinder can be seen (topologically) as a sphere with two holes in it so $\chi = 2 - 2 = 0$.
- The Euler-Poincare formula is a generalization of Euler's formula for general 3D surfaces that may have a topology that is not the one of a sphere. Assume a polyhedron (n vertices, n_e edges and n_f facets) that covers a surface of topology χ, we have

$$n - n_e + n_f = \chi.$$

Euler-Poincare – Triangular meshes

Assume a triangular mesh mesh with n vertices, n_e edges and n_f triangular facets that covers a domain that has the topology of a sphere $(\chi = 2)$:

$$n - n_e + n_f = \chi.$$

- Each edge has exactly two neighboring triangles and each triangle has three edges:

$$3n_f = 2n_e$$

Euler-Poincare – Triangular meshes

Assume a triangular mesh mesh with n vertices, n_e edges and n_f triangular facets that covers a domain that has the topology of a sphere ($\chi = 2$):

$$n - n_e + n_f = \chi.$$

- Each edge has exactly two neighboring triangles and each triangle has three edges:

$$3n_f = 2n_e$$

- With Euler's formula:

$$n_f = 2(n-2) \quad , \quad n_e = 3(n-2).$$

Euler-Poincare – Triangular meshes

Assume a triangular mesh mesh with n vertices, n_e edges and n_f triangular facets that covers a domain with topology χ:

$$n - n_e + n_f = \chi.$$

Assume that n_h edges and vertices are located on the boundaries of the surface.

- Each triangle has 3 edges. Each internal edge has two triangles and each edge on the boundary is asjacent to on triangle:

$$3n_f = 2(n_e - n_h) + n_h$$

Euler-Poincare – Triangular meshes

Assume a triangular mesh mesh with n vertices, n_e edges and n_f triangular facets that covers a domain with topology χ:

$$n - n_e + n_f = \chi.$$

Assume that n_h edges and vertices are located on the boundaries of the surface.

- Each triangle has 3 edges. Each internal edge has two triangles and each edge on the boundary is asjacent to on triangle:

$$3n_f = 2(n_e - n_h) + n_h$$

- With Euler's formula:

$$n_f = 2(n - \chi) - n_h \quad , \quad n_e = 3(n - \chi) - n_h.$$

Euler-Poincare – Triangular meshes

Assume a triangular mesh mesh with n vertices, n_e edges and n_f triangular facets that covers a domain with topology χ:

$$n - n_e + n_f = \chi.$$

Assume that n_h edges and vertices are located on the boundaries of the surface.

- Each triangle has 3 edges. Each internal edge has two triangles and each edge on the boundary is asjacent to on triangle:

$$3n_f = 2(n_e - n_h) + n_h$$

- With Euler's formula:

$$n_f = 2(n - \chi) - n_h \quad , \quad n_e = 3(n - \chi) - n_h.$$

- There are asymptotically 3 times more edges than nodes and 2 times more triangles than nodes in a triangular mesh.

Euler-Poincare – Triangular meshes

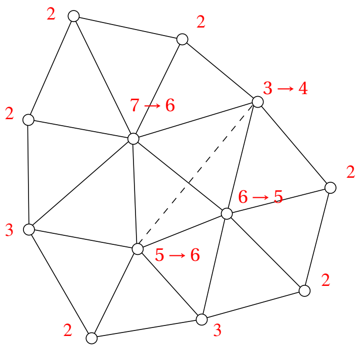
- A triangle has three vertices and each vertex is adjacent in average to n_{vf} triangles. This leads to
$$n_{vf}n = 3n_f = 3\,(2(n-\chi)-n_h) \quad \rightarrow \quad n_{vf} = 6 - \frac{3n_h + 3\chi}{n}.$$

Euler-Poincare – Triangular meshes

- A triangle has three vertices and each vertex is adjacent in average to n_{vf} triangles. This leads to

$$n_{vf} n = 3n_f = 3\,(2(n-\chi) - n_h) \quad \rightarrow \quad n_{vf} = 6 - \frac{3n_h + 3\chi}{n}.$$

- This means that, for large meshes, there is in average 6 triangles surrounding every vertex.



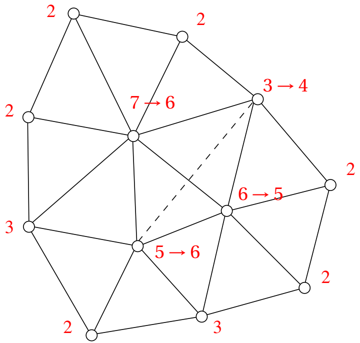
A triangulation T with $n = 12$ and $n_h = 9$. The average number of triangles adjacent to a vertex is $n_{vf} = 6 - \frac{3\times 9+6}{12} = 3,25$. This average can also be computed explicitely: $n_{vf} = \frac{39}{12} = 3,25$.

Euler-Poincare – Triangular meshes

- A triangle has three vertices and each vertex is adjacent in average to n_{vf} triangles. This leads to

$$n_{vf} n = 3 n_f = 3\left(2(n-\chi) - n_h\right) \quad \rightarrow \quad n_{vf} = 6 - \frac{3n_h + 3\chi}{n}.$$

- This means that, for large meshes, there is in average 6 triangles surrounding every vertex.
- There is, in average, exactly 6 triangles surrounding each vertex on a triangular mesh of a torus $(n_h = \chi = 0)$.



A triangulation T with $n = 12$ and $n_h = 9$. The average number of triangles adjacent to a vertex is $n_{vf} = 6 - \frac{3\times 9+6}{12} = 3,25$. This average can also be computed explicitely: $n_{vf} = \frac{39}{12} = 3,25$.

Regular triangulations

$$n_f = 2(n - \chi) - n_h.$$

- Closed surface, no boundaries, $n_h = 0$.

Regular triangulations

$$n_f = 2(n - \chi) - n_h.$$

- Closed surface, no boundaries, $n_h = 0$.
- Regular topology: exactly 6 triangles adjacent to a vertex:

$$3n_f = 6n \quad \to \quad n_f = 2n.$$

Regular triangulations

$$n_f = 2(n - \chi) - n_h.$$

- Closed surface, no boundaries, $n_h = 0$.
- Regular topology: exactly 6 triangles adjacent to a vertex:

$$3n_f = 6n \quad \rightarrow \quad n_f = 2n.$$

- Restriction:

$$2n = 2(n - \chi) \quad \rightarrow \quad \chi = 0.$$

Regular triangulations

$$n_f = 2(n - \chi) - n_h.$$

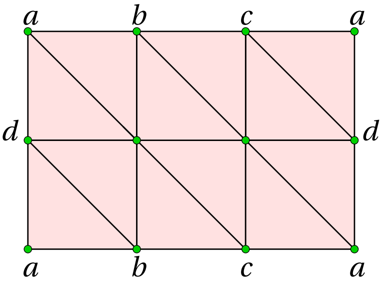
- Closed surface, no boundaries, $n_h = 0$.
- Regular topology: exactly 6 triangles adjacent to a vertex:

$$3n_f = 6n \quad \rightarrow \quad n_f = 2n.$$

- Restriction:

$$2n = 2(n - \chi) \quad \rightarrow \quad \chi = 0.$$

- Regular triangulations of closed surfaces are only possible for torus topologies ($\chi = 0$).



Regular triangulations with boundaries

$$n_f = 2(n - \chi) - n_h.$$

- We have n_h edges/vertices on the boundaries of the surface.

Regular triangulations with boundaries

$$n_f = 2(n - \chi) - n_h.$$

- We have n_h edges/vertices on the boundaries of the surface.
- Regular topology: exactly 6 triangles adjacent to an internal vertex and 3 triangles adjacent to a boundary vertex.

$$3n_f = 6(n - n_h) + 3n_h \quad \rightarrow \quad n_f = 2n - n_h.$$

Regular triangulations with boundaries

$$n_f = 2(n - \chi) - n_h.$$

- We have n_h edges/vertices on the boundaries of the surface.
- Regular topology: exactly 6 triangles adjacent to an internal vertex and 3 triangles adjacent to a boundary vertex.

$$3n_f = 6(n - n_h) + 3n_h \quad \rightarrow \quad n_f = 2n - n_h.$$

- Same restriction:

$$2n - n_h = 2(n - \chi) - n_h \quad \rightarrow \quad \chi = 0.$$

Regular triangulations with boundaries

$$n_f = 2(n - \chi) - n_h.$$

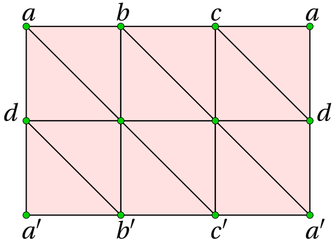
- We have n_h edges/vertices on the boundaries of the surface.
- Regular topology: exactly 6 triangles adjacent to an internal vertex and 3 triangles adjacent to a boundary vertex.

$$3n_f = 6(n - n_h) + 3n_h \quad \rightarrow \quad n_f = 2n - n_h.$$

- Same restriction:

$$2n - n_h = 2(n - \chi) - n_h \quad \rightarrow \quad \chi = 0.$$

- Regular triangulations of general surfaces are only available for $\chi = 0$ i.e. surface of a cylinder or torus.



Quasi-regular triangulations

- Introduction of n_k , $k = -2, -1, 1, 2$ non-regular internal vertices of degree $6 - k$.

Quasi-regular triangulations

- Introduction of n_k , $k = -2, -1, 1, 2$ non-regular internal vertices of degree $6 - k$.
- Introduction of m_l , $l = -2, -1, 1, 2$ non-regular boundary vertices of degree $3 - k$.

Quasi-regular triangulations

- Introduction of n_k , $k = -2, -1, 1, 2$ non-regular internal vertices of degree $6 - k$.
- Introduction of m_l , $l = -2, -1, 1, 2$ non-regular boundary vertices of degree $3 - k$.
- This leads to

$$3n_f = \sum_k [(6-k)n_k + (3-k)m_k + 6(n - n_k - n_h) + 3(n_h - m_k)]$$

Quasi-regular triangulations

- Introduction of n_k , $k = -2, -1, 1, 2$ non-regular internal vertices of degree $6 - k$.
- Introduction of m_l , $l = -2, -1, 1, 2$ non-regular boundary vertices of degree $3 - k$.
- This leads to

$$3n_f = \sum_k [(6-k)n_k + (3-k)m_k + 6(n - n_k - n_h) + 3(n_h - m_k)]$$

Finally , using $n_f = 2(n - \chi) - n_h$, we get

$$6n - 6\chi - 3n_h = \sum_k [(6-k)n_k + (3-k)m_k + 6(n - n_k - m_k) + 3(n_h - m_k)]$$

that simplifies into

$$\chi = -\sum_k \frac{k}{6}(n_k + m_k).$$

Quasi-regular triangulations

$$\chi = -\sum_k \frac{k}{6}(n_k + m_k).$$

This formula has quite interresting implications

- It is possible to compute χ only by counting singularities

Quasi-regular triangulations

$$\chi = -\sum_k \frac{k}{6}(n_k + m_k).$$

This formula has quite interresting implications

- It is possible to compute χ only by counting singularities
- Each singularity of index k count as $-k/6$ in the Poincare characteristic.

Quasi-regular triangulations

$$\chi = -\sum_k \frac{k}{6}(n_k + m_k).$$

This formula has quite interresting implications

- It is possible to compute χ only by counting singularities
- Each singularity of index k count as $-k/6$ in the Poincare characteristic.
- A vertex with 5 neighboring triangles counts for $1/6$

Quasi-regular triangulations

$$\chi = -\sum_k \frac{k}{6}(n_k + m_k).$$

This formula has quite interresting implications

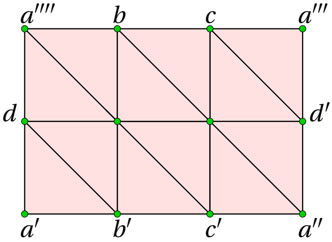
- It is possible to compute χ only by counting singularities
- Each singularity of index k count as $-k/6$ in the Poincare characteristic.
- A vertex with 5 neighboring triangles counts for $1/6$
- A vertex with 7 neighboring triangles counts for $-1/6$

Quasi-regular triangulations

$$\chi = -\sum_k \frac{k}{6}(n_k + m_k).$$

This formula has quite interresting implications

- It is possible to compute χ only by counting singularities
- Each singularity of index k count as $-k/6$ in the Poincare characteristic.
- A vertex with 5 neighboring triangles counts for $1/6$
- A vertex with 7 neighboring triangles counts for $-1/6$
- In the example, $\chi = 1$ and vertices a, a', a'' and a''' are irregular: a and a''' have indices $k = -1$ and a' and a'' have indices $k = -2$, which leads to $1/6 + 1/6 + 2/6 + 2/6 = 1$.



Euler-Poincare – Quadrangular meshes

Assume a quad-mesh with n vertices, n_e edges and n_f quad facets that covers a domain with topology χ:

$$n - n_e + n_f = \chi.$$

Assume that n_h edges and vertices are located on the boundaries of the surface.

- Each quad has 4 edges. Each internal edge has two adjacent quads and each edge on the boundary is adjacent to on quad:

$$4n_f = 2(n_e - n_h) + n_h$$

Euler-Poincare – Quadrangular meshes

Assume a quad-mesh with n vertices, n_e edges and n_f quad facets that covers a domain with topology χ:

$$n - n_e + n_f = \chi.$$

Assume that n_h edges and vertices are located on the boundaries of the surface.

- Each quad has 4 edges. Each internal edge has two adjacent quads and each edge on the boundary is adjacent to on quad:

$$4n_f = 2(n_e - n_h) + n_h$$

- With Euler's formula:

$$n_f = n - \chi - \frac{n_h}{2}$$

Euler-Poincare – Quadrangular meshes

Assume a quad-mesh with n vertices, n_e edges and n_f quad facets that covers a domain with topology χ:

$$n - n_e + n_f = \chi.$$

Assume that n_h edges and vertices are located on the boundaries of the surface.

- Each quad has 4 edges. Each internal edge has two adjacent quads and each edge on the boundary is adjacent to on quad:

$$4n_f = 2(n_e - n_h) + n_h$$

- With Euler's formula:

$$n_f = n - \chi - \frac{n_h}{2}$$

- Quad meshes are only possible if n_f is even!

Regular quadrangulations

$$n_f = n - \chi - \frac{n_h}{2}$$

- Closed surface, no boundaries, $n_h = 0$.

Regular quadrangulations

$$n_f = n - \chi - \frac{n_h}{2}$$

- Closed surface, no boundaries, $n_h = 0$.
- Regular topology: exactly 6 triangles adjacent to a vertex:

$$4n_f = 4n \quad \rightarrow \quad n_f = n.$$

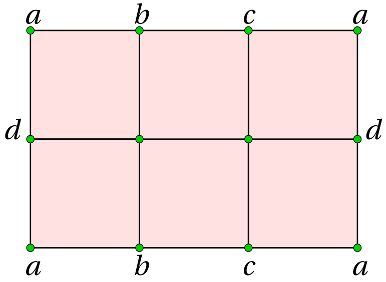
Regular quadrangulations

$$n_f = n - \chi - \frac{n_h}{2}$$

- Closed surface, no boundaries, $n_h = 0$.
- Regular topology: exactly 6 triangles adjacent to a vertex:

$$4n_f = 4n \quad \rightarrow \quad n_f = n.$$

- Regular quadrangulations of closed surfaces are only possible for torus topologies ($\chi = 0$).



Regular quadrangulations with boundaries

$$n_f = 2(n - \chi) - n_h.$$

- We have n_h edges/vertices on the boundaries of the surface.

Regular quadrangulations with boundaries

$$n_f = 2(n - \chi) - n_h.$$

- We have n_h edges/vertices on the boundaries of the surface.
- Regular topology: exactly 4 quads adjacent to an internal vertex and 2 quads adjacent to a boundary vertex.

$$4n_f = 4(n - n_h) + 2n_h \quad \rightarrow \quad n_f = n - \frac{n_h}{2}.$$

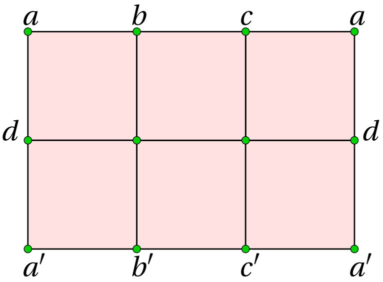
Regular quadrangulations with boundaries

$$n_f = 2(n - \chi) - n_h.$$

- We have n_h edges/vertices on the boundaries of the surface.
- Regular topology: exactly 4 quads adjacent to an internal vertex and 2 quads adjacent to a boundary vertex.

$$4n_f = 4(n - n_h) + 2n_h \quad \rightarrow \quad n_f = n - \frac{n_h}{2}.$$

- Regular quadrangulations of general surfaces are only available for $\chi = 0$ i.e. surface of a cylinder or torus.



Quasi-regular quadrangulations

- Introduction of n_k , $k = -2, -1, 1, 2$ non-regular internal vertices of degree $4 - k$.

Quasi-regular quadrangulations

- Introduction of n_k , $k = -2, -1, 1, 2$ non-regular internal vertices of degree $4 - k$.
- Introduction of m_l , $l = -2, -1, 1, 2$ non-regular boundary vertices of degree $2 - k$.

Quasi-regular quadrangulations

- Introduction of n_k , $k = -2, -1, 1, 2$ non-regular internal vertices of degree $4 - k$.
- Introduction of m_l , $l = -2, -1, 1, 2$ non-regular boundary vertices of degree $2 - k$.
- This leads to

$$4n_f = \sum_k \left[(4-k)n_k + (2-k)m_k + 4(n - n_k - n_h) + 2(n_h - m_k)\right]$$

Quasi-regular quadrangulations

- Introduction of n_k , $k = -2, -1, 1, 2$ non-regular internal vertices of degree $4 - k$.
- Introduction of m_l , $l = -2, -1, 1, 2$ non-regular boundary vertices of degree $2 - k$.
- This leads to

$$4n_f = \sum_k [(4-k)n_k + (2-k)m_k + 4(n - n_k - n_h) + 2(n_h - m_k)]$$

Finally , using $n_f = 2(n - \chi) - n_h$, we get

$$\chi = -\sum_k \frac{k}{4}(n_k + m_k).$$

Quasi-regular quadrangulations

$$\chi = -\sum_k \frac{k}{4}(n_k + m_k).$$

This formula has quite interresting implications

- It is possible to compute χ only by counting singularities

Quasi-regular quadrangulations

$$\chi = -\sum_k \frac{k}{4}(n_k + m_k).$$

This formula has quite interresting implications

- It is possible to compute χ only by counting singularities
- Each singularity of index k count as $-k/4$ in the Poincare characteristic.

Quasi-regular quadrangulations

$$\chi = -\sum_k \frac{k}{4}(n_k + m_k).$$

This formula has quite interresting implications

- It is possible to compute χ only by counting singularities
- Each singularity of index k count as $-k/4$ in the Poincare characteristic.
- A vertex with 3 neighboring triangles counts for $1/4$

Quasi-regular quadrangulations

$$\chi = -\sum_k \frac{k}{4}(n_k + m_k).$$

This formula has quite interresting implications

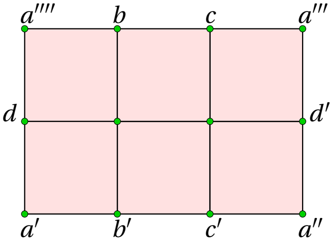
- It is possible to compute χ only by counting singularities
- Each singularity of index k count as $-k/4$ in the Poincare characteristic.
- A vertex with 3 neighboring triangles counts for $1/4$
- A vertex with 5 neighboring triangles counts for $-1/4$

Quasi-regular quadrangulations

$$\chi = -\sum_k \frac{k}{4}(n_k + m_k).$$

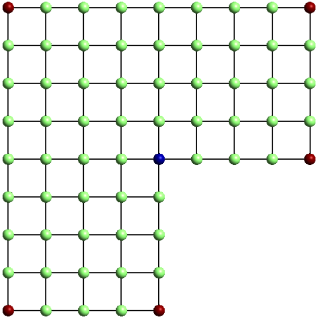
This formula has quite interresting implications

- It is possible to compute χ only by counting singularities
- Each singularity of index k count as $-k/4$ in the Poincare characteristic.
- A vertex with 3 neighboring triangles counts for $1/4$
- A vertex with 5 neighboring triangles counts for $-1/4$
- In the example, $\chi = 1$ and vertices a, a', a'' and a''' are irregular and of index $k = -1$, which leads to $1/4 + 1/4 + 1/4 + 1/4 = 1$.



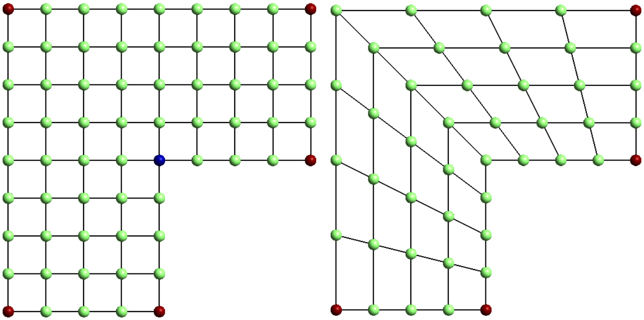
Quasi-regular quadrangulations

- Quadrilateral meshes of a non smooth domain. Five singularities of index $1/4$ (in red) and one singularity of index $-1/4$ (in blue) are required to have the sum of the indices to be one (left).



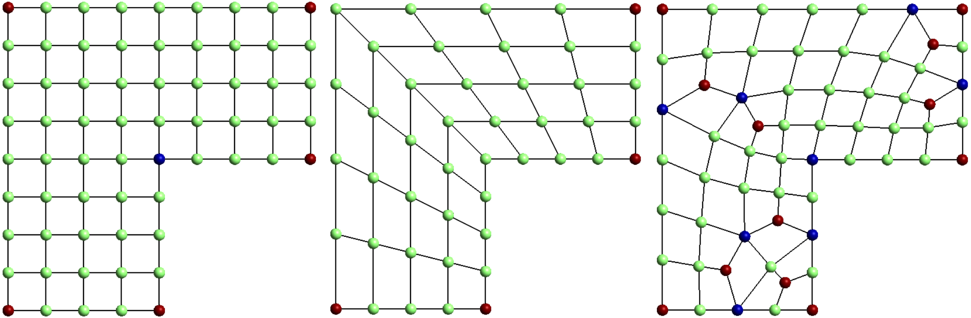
Quasi-regular quadrangulations

- Quadrilateral meshes of a non smooth domain. Five singularities of index $1/4$ (in red) and one singularity of index $-1/4$ (in blue) are required to have the sum of the indices to be one (left).
- It is also possible to use 4 irregular nodes only (right), leading to a different result.



Quasi-regular quadrangulations

- Quadrilateral meshes of a non smooth domain. Five singularities of index $1/4$ (in red) and one singularity of index $-1/4$ (in blue) are required to have the sum of the indices to be one (left).
- It is also possible to use 4 irregular nodes only (right), leading to a different result.
- Quadrilateral mesh with 8 vertices of index $-1/4$, and 12 of index $1/4$, leading to $\chi = 12/4 - 8/4 = 1$.

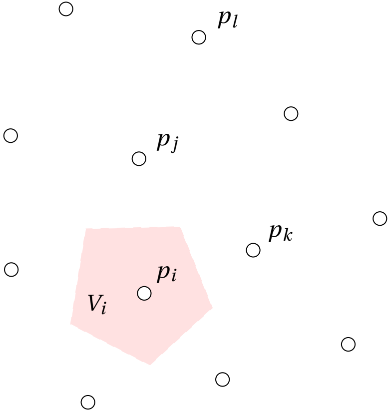


The Voronoï Diagram

Definition: Consider a finite set $S = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^2$ of n distinct points in the plane. The *Voronoi cell* V_i of $p_i \in S$ is the set of points x that are closer to p_i than to any other points of the set:

$$V_i = \left\{x \in \mathbb{R}^2 \mid \|x - p_i\| < \|x - p_j\|, \ \forall 1 \leq i \leq n, i \neq j\right\}$$

where $\|x - y\|$ is the euclidian distance between x and y.

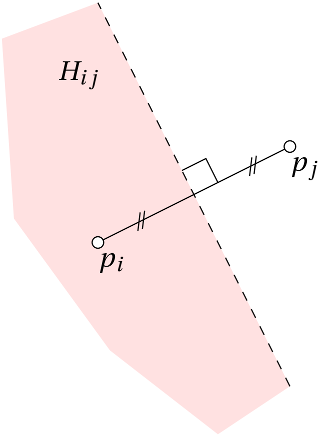


The Voronoï Diagram of 2 points p_i and p_j

The perpendicular bissector of p_ip_j divides $\mathbb{R}^2$ into two halfplanes H_{ij} and H_{ji}:

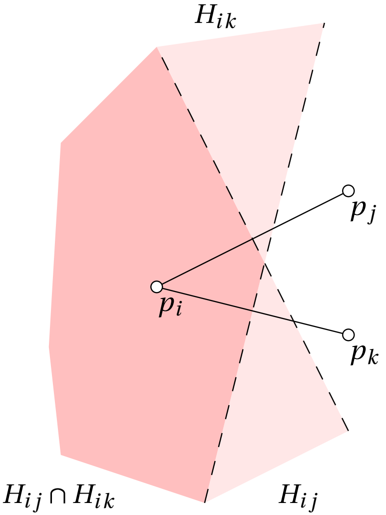
$$H_{ij} = \left\{x \in \mathbb{R}^2 \mid \|x - p_i\| < \|x - p_j\|\right\}.$$

We have $V_i = H_{ij}$.



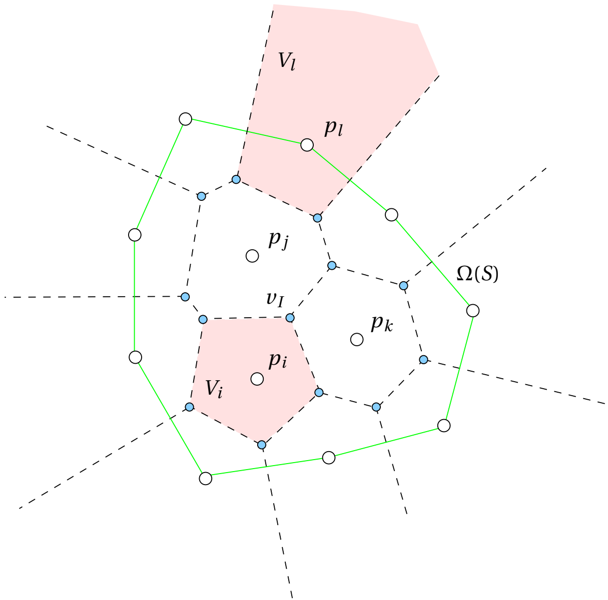
The Voronoï Diagram of 3 points

Let's make the problem a little more complicated and consider a set $S = \{p_i, p_j, p_k\}$ of 3 points. The Voronoi cell associated to p_i is the intersection of half planes H_{ij} and H_{ik}: $V_i = H_{ij} \cap H_{ik}$.



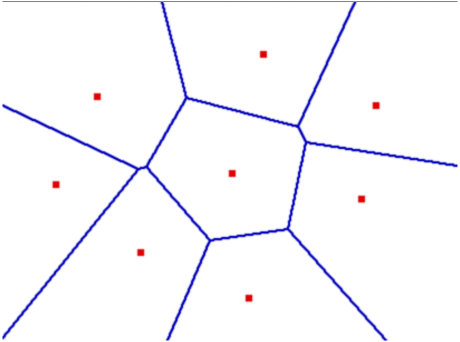
The Voronoï Diagram

The Voronoi diagram $V(S)$ is the unique subdivision of the plane into n cells. Its is the union of all Voronoi cells V_p:

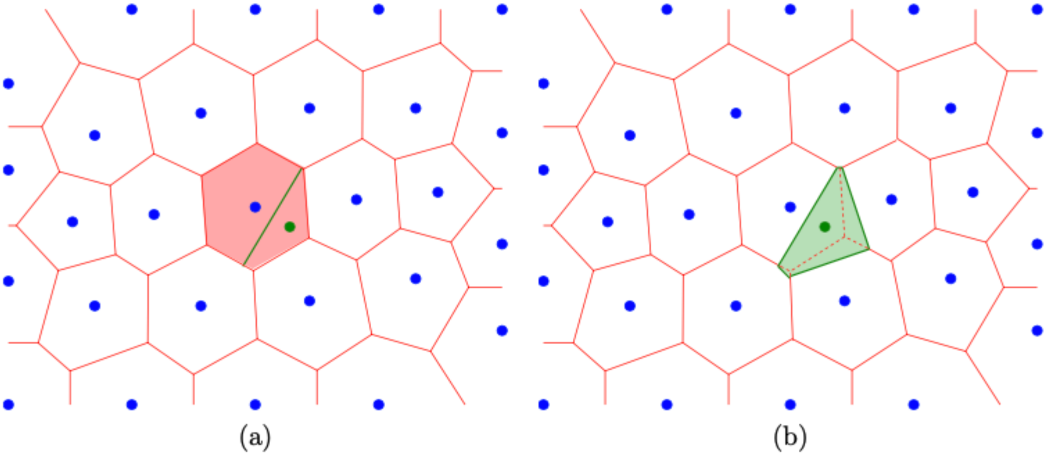


Green and Sibson's algorithm ($\mathcal{O}(n^2)$)

- Incremental: adding a point only modifies the diagram locally
- Let $S_n = \{p_1, p_2, \ldots, p_n\}$ and $V(S_n)$. Add p_{n+1} to form $V(S_{n+1})$ with $S_{n+1} = \{p_1, p_2, \ldots, p_{n+1}\}$.
  1. Find voronoi cell V_i such that $p_{n+1} \in V_i$.
  2. Draw orthogonal bissector of $p_{n+1}p_i$ and compute x_1 and x_2 its intersections with V_i (only 2 intersections because V_i is convex.
  3. x_1x_2 is the Voronoï edge that separates V_{n+1} and V_i. Start with x_2 that sits on a Voronoï edge of $V(S)$ that separates V_i with V_j.
  4. Replace i by j and goto 2 until x_2 goes back to x_1.
  5. The Voronoï cell V_{n+1} relative to p_{n+1} has been created. Remove the parts of all V_i's that have been "eaten" by V_{n+1}.

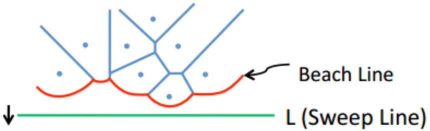


Green and Sibson's algorithm



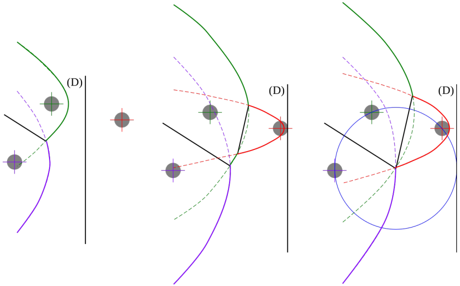
Fortune's algorithm ($\mathcal{O}(n\log(n))$)

- Line sweep (like intersection of lines) e.g. from left to right. Main issue, a part of the diagram on the left of the line depends on points on the right of the line.
- Fortune solves the issue by introducing a "beach line" that is (i) made of parabolas and that is (ii) delayed with respect to the sweep line.
- For each point left of the sweep line, one can define a parabola of points equidistant from that point and from the sweep line; the beach line is the boundary of the union of these parabolas.
- As the sweep line progresses, the vertices of the beach line, at which two parabolas cross, trace out the edges of the Voronoi diagram.
- The algorithm maintains as data structures a binary search tree describing the combinatorial structure of the beach line, and a priority queue listing potential future events that could change the beach line structure.



Fortune's algorithm ($\mathcal{O}(n\log(n))$)

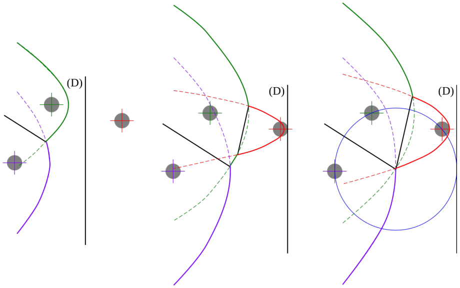
- Sweep line L passes through a first point p_1 and initiates a parabola P_1 s.t. $d(L, P_1) = d(p_1, P_1)$.
- Sweep line L passes through a second point p_2 and initiates a parabola P_2. Intersection point I between P_1 and P_2 verifies $d(I, p_1) = d(I, p_2)$ so I belongs to the Voronoï edge between p_1 and p_2.
- Sweep line L passes through a third point p_3 and initiates a parabola P_3. If points are in general position, there exist a circle C containing the 3 points. When L is tangent to C, its center is a Voronoï vertex. At that point, a part of P_1 must be removed from the beachline.



Fortune's algorithm ($\mathcal{O}(n\log(n))$)

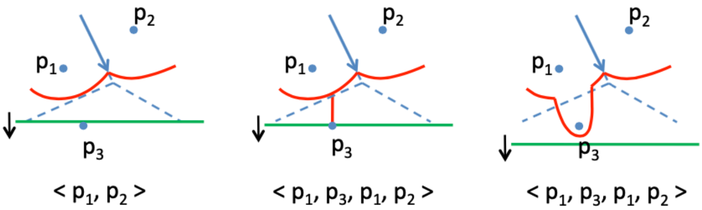
Two types of events:

- **Point event** A new parabola P_i is created whenever the sweep lines encounters seed p_i.
- **Circle/Vertex event** Disparition of a piece of parabola when the sweep line encounters a vertex i.e. the circumcircle of three "seeds".
- Both the point event and the vertex event can be handled in $\mathcal{O}(\log(n))$ time.
- Fortune's algorithm computes the Voronoï diagram in $\mathcal{O}(n\log(n))$ time. The storage space requirement is $\mathcal{O}(n)$.



Point Event

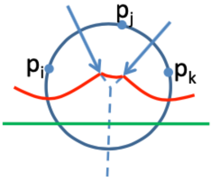
Point Event:



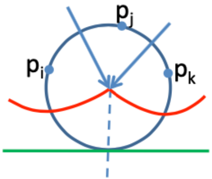
- To process a point event:
  - Determine the arc of the beach line directly above the new point
  - Split the arc into two by inserting a new infinitesimally small arc at this point
  - As the sweep proceeds this arc will start to widen

Circle/Vertex Event

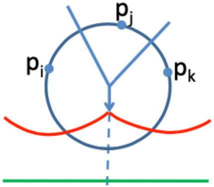
1. P_i, P_j, and P_k whose arcs appear consecutively on the beach line. The circumcircle lies partially below the sweep line



2. Circumcircle is empty and the center is equidistant to p_i, p_j, p_k, and L. The center is a Voronoi vertex.

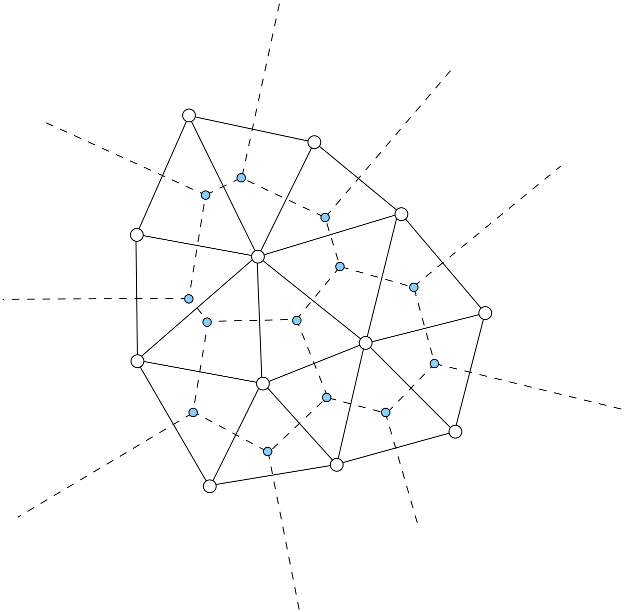


3. The arc of p_j disappears from the beach line



The Delaunay triangulation

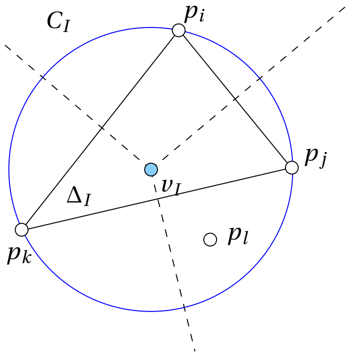
The Delaunay triangulation DT(S) is the geometric dual of the Voronoï diagram



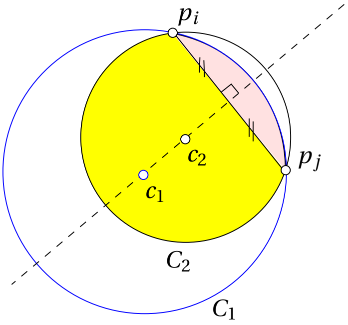
The empty circle property

The circumcircle of any triangle in the Delaunay triangulation is empty i.e. it contains no point of S.

- Consider the Delaunay triangle $\Delta_I = p_i p_j p_k$. Assume now that point $p_l \in C_I$ where C_I is the circumcircle of Δ_I.
- By definition, the triple point v_I is at equal distance to p_i, p_j and p_k and no other points of S are closer to v_I than those three points.
- Then, if a point like p_l exist in S, v_I is not a triple point and triangle Δ_I cannot be a Delaunay triangle.



Delaunay Edges

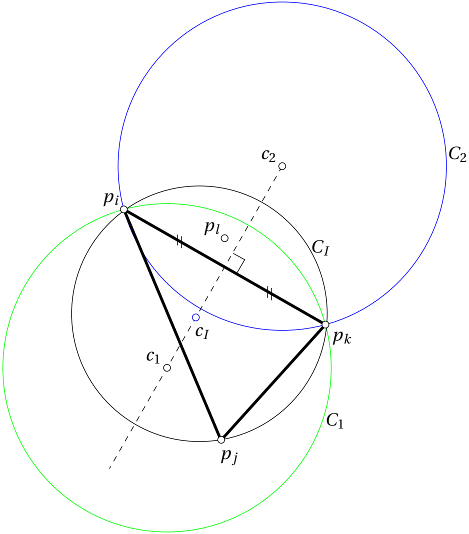


- Two circles C_1 and C_2 sharing an edge p_ip_j. The centers of the circles c_1 and c_2 lie on the perpendicular bissector of segment p_ip_j (in dashed lines).
- Edge p_ip_j divides disk C_1 into two disk sectors and one of the two sectors completely lies inside C_2. On the Figure, the pink sector of C_1 is inside C_2 and the yellow sector of C_2 lies inside C_1.

Delaunay Edges

An edge $p_i p_j$ of a triangulation is a *Delaunay edge* if there exist a circle that contains p_i and p_j and that is empty i.e. that contain no point of S.

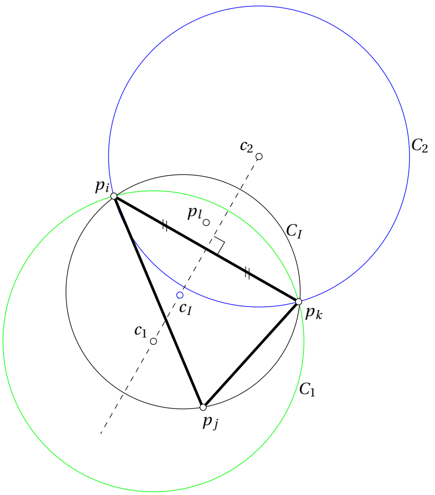
A mesh is a Delaunay Triangulation if and only if all its edges are Delaunay edges.



Delaunay Edges

Let us first show that a Delaunay triangulation has only Delaunay edges.

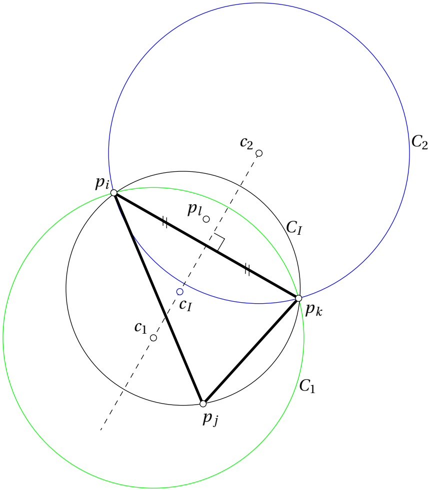
- Assume a Delaunay triangulation $T(S)$ and an edge p_ip_j that is not Delaunay.
- This means that there exist no circle passing through p_i and p_j that is empty.
- Consider Delaunay triangle $\Delta_I = p_ip_jp_k$ that contains edge p_ip_j.
- Its circumcircle is empty by definition because T is a Delaunay triangulation.
- This is in contradiction with the hypothesis that there exist no circle passing through p_i and p_j and that is not empty.



Delaunay Edges

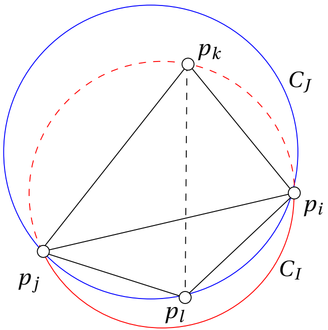
Now let's proof that if every edge of a triangulation is Delaunay, then every triangle is Delaunay as well.

- Assume that triangle $\Delta_I = p_i p_j p_k$ is not Delaunay (p_l is inside its circle), but all its 3 edges $p_i p_j$, $p_i p_k$ and $p_j p_k$ are Delaunay.
- Point p_l cannot be inside triangle Δ_I. It is then situated inside one of the three circular sectors delimited by p_i, p_j and p_k.
- Assume that p_l and p_j are on opposite sides of $p_i p_k$. By hypothesis, there exist a circle passing through p_i and p_k and that is empty. The center of such a circle lies on the orthogonal bissector of $p_i p_k$. Any circle like C_1 with its center c_1 that is below c_I contains p_j any circle C_2 that is above c_I contains p_l, which is in contradiction with the hypothesis that there exist a circle passing through $p_i p_k$ and that is empty.



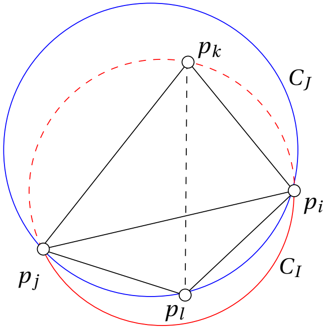
Local Delaunayhood

- Given a triangulation $T(S)$ and an edge p_ip_j in the triangulation that is adjacent to two triangles $\Delta_I = p_ip_jp_k$ and $\Delta_J = p_ip_lp_j$. We call edge p_ip_j *locally Delaunay* if p_l lies outside the circumcircle of Δ_I.
- Edge p_ip_j is not locally Delaunay on the Figure.
- It is easy to see that this condition is symmetric: if point p_l lies inside circle C_I, then point p_k lies inside circle C_J. We'll prove that below.



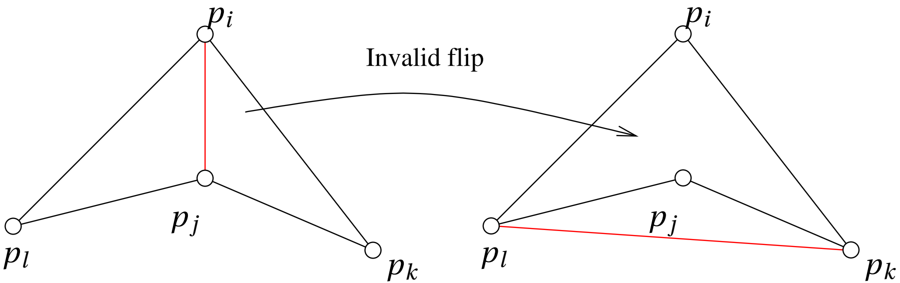
Edge Flip

- Consider again the situation of two triangles adjacent to edge $p_i p_j$ as depicted in the Figure.
- Flipping edge $p_i p_j$ consist in replacing triangles $p_i p_j p_k$ and $p_j p_i p_l$ by triangles $p_l p_k p_i$ and $p_k p_l p_j$.
- Edge $p_i p_j$ has been flipped and replaced by edge $p_k p_l$.



Edge Flip

The edge flip operator can only be applied to a pair of triangles that form a convex quadrilateral. If it is concave, then flipping the edge leads to an invalid configuration with two overlapping triangles



- An edge that is not locally Delaunay is flippable and the new edge resulting of the flip operation is locally Delaunay.
- If all edges of triangulation $T(S)$ are locally Delaunay, then T is the Delaunay triangulation $\mathrm{DT}(S)$.

The Flip Algorithm

Flip until you drop:

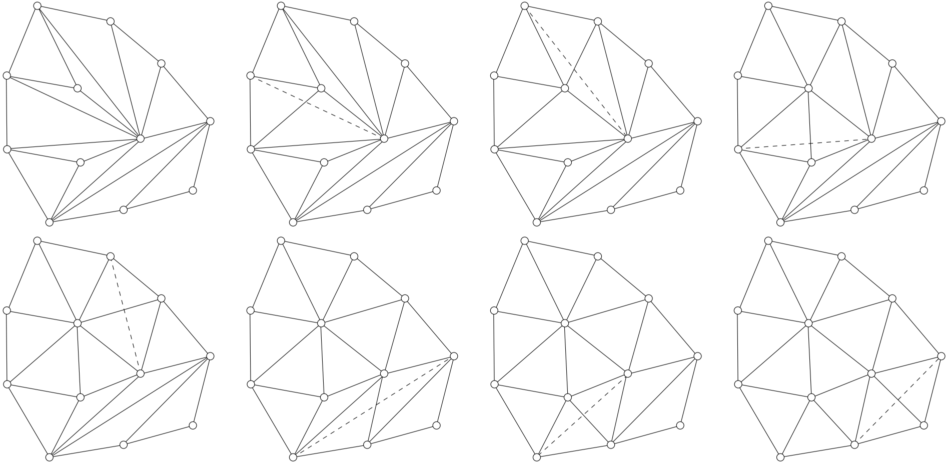
- Insert all the internal edges of $T(S)$ in a `stack`.
- Do while the `stack` is not empty
  - Take edge p_ip_j at the top of the `stack`. This edge is adjacent to triangles $p_ip_jp_k$ and $p_jp_ip_l$. If p_ip_j is not locally Delaunay, then flip it and add edges p_ip_k,p_kp_j, p_jp_l and p_lp_i in the stack. If one of those edges was already present in the stack, update its neighbors.
  - Remove p_ip_j from the `stack`.

Two questions should be asked at that point: (i) does this algorithm produce the Delaunay triangulation of S and (ii) if it achieves to create DT(S), what is its complexity (does it simply terminate)?

The edge flip algorithm converges to DT(S) in at most $\mathcal{O}(n^2)$ flips

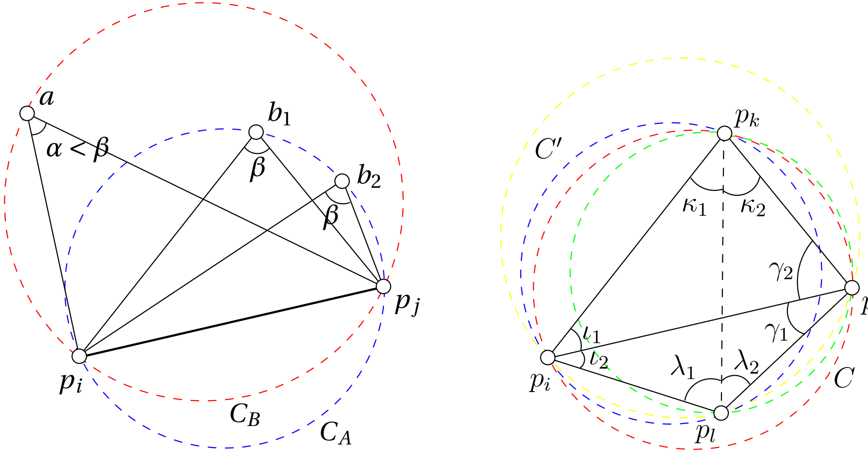
This result is outmost importance. It means that every triangulation $T(S)$ is "connected" to the Delaunay triangulation DT(S) by at most $\mathcal{O}(n^2)$ flips. It also means that any two triangulations T and T' are flip connected.

The Flip Algorithm



The MaxMin property

The Delaunay triangulation DT(S) is angle-optimal: it maximizes the minimum angle among all possible triangulations.

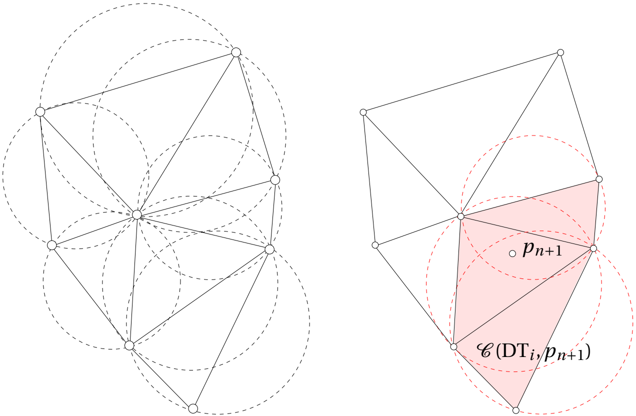


Thales theorem (left) and MaxMin property illustrated (right)

Bowyer-Watson Algorithm

Let DT_n be the Delaunay triangulation of a point set $S_n = \{p_1, \ldots, p_n\} \subset \mathbb{R}^2$ that are in general position. We describe an incremental process allowing the insertion of a given point $p_{n+1} \in \Omega(S_n)$ into DT_n and to build the Delaunay triangulation DT_{n+1} of $S_{n+1} = \{p_1, \ldots, p_n, p_{n+1}\}$.

$$\mathrm{DT}_{n+1} = \mathrm{DT}_n - \mathcal{C}(\mathrm{DT}_n, p_{n+1}) + \mathcal{B}(\mathrm{DT}_n, p_{n+1}). \tag{4}$$

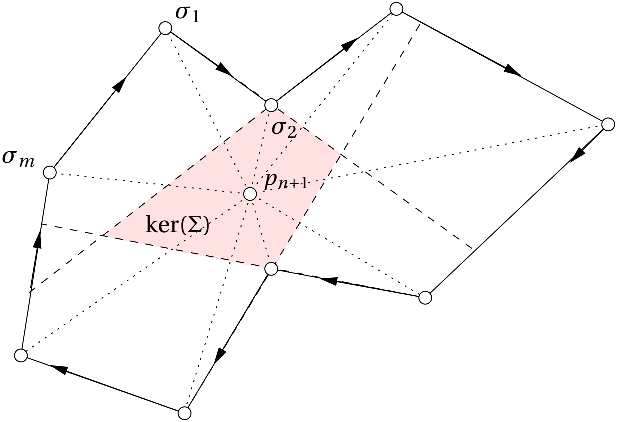


Bowyer-Watson Algorithm

Consider a polygon Σ with m corners $\sigma_1, \dots, \sigma_m$ that is bounded by m edges $\sigma_i, \sigma_{(i+1)\%m}$, $1 \le i \le m$.

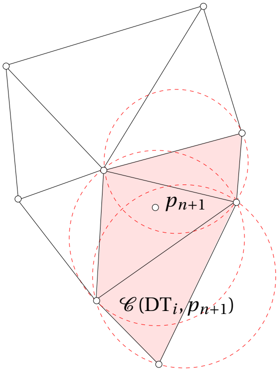
The kernel $\ker(\Sigma)$ is the set of point $x \in \mathbb{R}^2$ that are visible to every σ_j i.e. the line segment $x\sigma_j$ them do not intersect any edges of the polygon.

The kernel $\ker(\Sigma)$ can be computed by intersection of the halfplanes that correspond to all oriented edges of the polygon (see Figure).

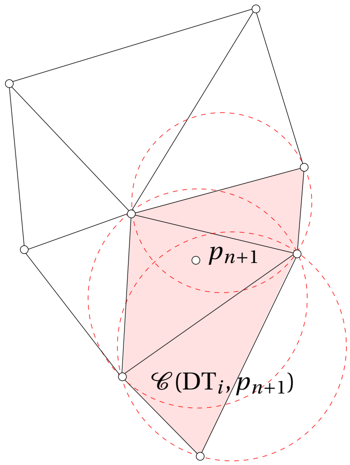


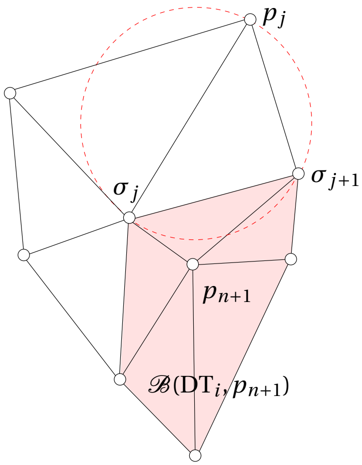
Bowyer-Watson Algorithm

The Delaunay cavity $\mathcal{C}(T_n, p_{n+1})$ is the set of m triangles $\Delta_1, \ldots, \Delta_m \in \mathrm{DT}_n$ for which their circumcircle contains p_{n+1}.
The Delaunay cavity contains the set of triangles that cannot belong to T_{n+1}. The region covered by those invalid triangles should be emptied and re-triangulated in a Delaunay fashion.
The Delaunay cavity has some interresting properties.
Theorem: The Delaunay cavity $\mathcal{C}(T_n, p_{n+1})$ is a non empty connected set of triangles which the union form a star shaped polygon with p_{n+1} in its kernel.



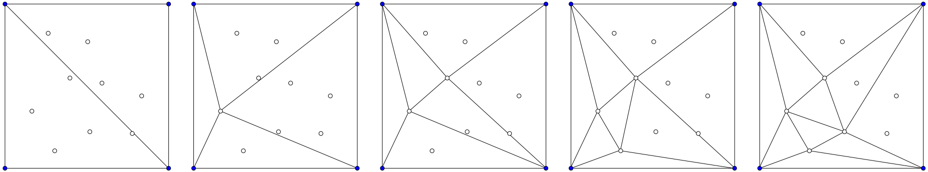
Bowyer-Watson Algorithm

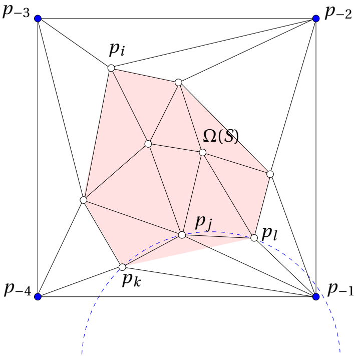




Bowyer-Watson Algorithm

Super triangles :

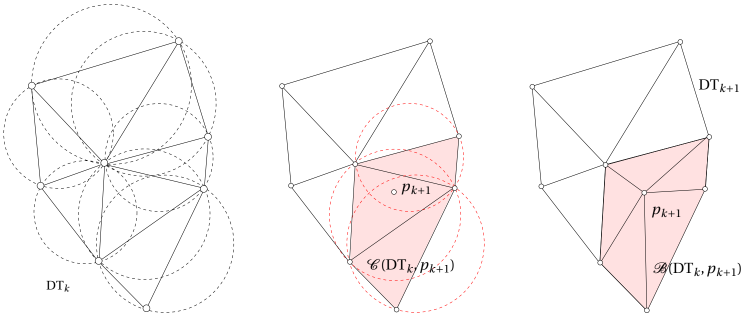




Triangulation of n points in $n \log(n)$ complexity

- Use Bowyer-Watson algorithm (not the best choice in 2D)

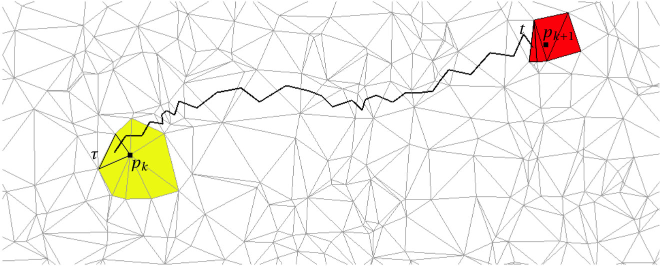
$$\mathsf{DT}_{k+1} = \mathsf{DT}_k - \mathcal{C}(\mathsf{DT}_k, p_{k+1}) + \mathcal{B}(\mathsf{DT}_k, p_{k+1})$$



Triangulation of n points in $n \log(n)$ complexity

- Use Bowyer-Watson algorithm (not the best choice in 2D)
- **Sort the points**, N. Amenta, S. Choi, and G. Rote. *Incremental constructions con brio.*, 2003.

Without sort: $\mathcal{O}(n^{1/d})$ "walking" steps per insertion → overall (best) complexity of $\mathcal{O}(n^{1+\frac{1}{d}})$



Triangulation of n points in $n\log(n)$ complexity

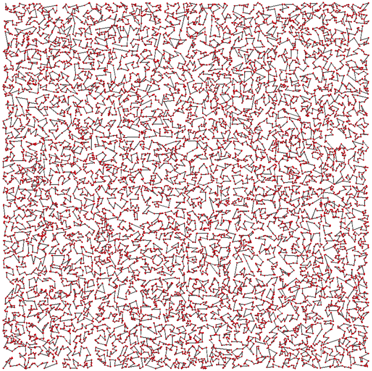
- Use Bowyer-Watson algorithm (not the best choice in 2D)
- **Sort the points**, N. Amenta, S. Choi, and G. Rote. *Incremental constructions con brio.*, 2003.
- **Robust predicates with static filters**, H. Si. *Tetgen, a delaunay-based quality tetrahedral mesh generator.*, 2015.

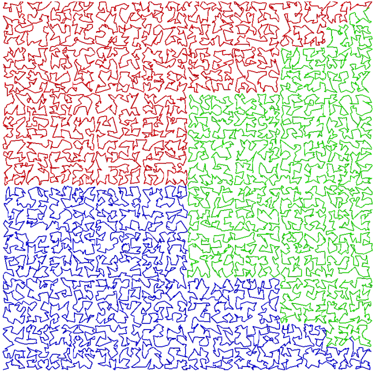
n	10^3	10^4	10^5	10^6	10^3	10^4	10^5	10^6
	2D				3D			
N_{walk}	23	73	230	727	17	38	85	186
t(sec)	$3.6\ 10^{-3}$	$9.1\ 10^{-2}$	3.98	187	$1.2\ 10^{-2}$	$1.8\ 10^{-1}$	3.42	73
	2D (BRIO)				3D (BRIO)			
N_{walk}	2.3	2.4	2.5	2.5	2.9	3.0	3.1	3.1
t(sec)	$2\ 10^{-3}$	$1.5\ 10^{-2}$	$1.5\ 10^{-1}$	1.47	$9.0\ 10^{-3}$	$7.5\ 10^{-2}$	$7.8\ 10^{-1}$	7.81

Triangulation of n points in $n\log(n)$ complexity

- Use Bowyer-Watson algorithm (not the best choice in 2D)
- **Sort the points**, N. Amenta, S. Choi, and G. Rote. *Incremental constructions con brio.*, 2003.
- **Robust predicates with static filters**, H. Si. *Tetgen, a delaunay-based quality tetrahedral mesh generator.*, 2015.
- **Multitreading**: distribute the Hilbert curve in M threads.

$$\mathsf{DT}_{k+1} = \mathsf{DT}_k + \sum_{i=0}^{M-1} \left[-\mathcal{C}(\mathsf{DT}_k, p_{k+i\frac{n}{M}}) + \mathcal{B}(\mathsf{DT}_k, p_{k+i\frac{n}{M}}) \right].$$





Hilbert curves

A curve $x(t)$ is defined as the mapping

$$x(t) \ , \ t \in [0,1] \to x \in \mathbb{R}^3.$$

Curves are perceived as one dimensional objects. Yet, it can be shown that a continuous curve can pass through every point of a unit square. The Hilbert space filling $\mathcal{H}(t)$ curve is a one dimensional curve which visits every point within a two dimensional space. It may be thought of as the limit

$$\mathcal{H}(t) = \lim_{k\to\infty} \mathcal{H}_k(t)$$

of a sequence of curves $\mathcal{H}_k$ (see Figure 1).

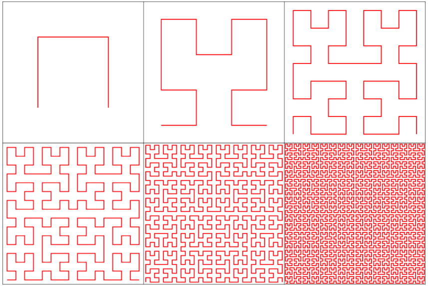


Figure: Sequense of Hilbert curves $\mathcal{H}_k$.

Hilbert curves

Curves $\mathcal{H}_1$ and $\mathcal{H}_2$ are depicted on Figure 2.

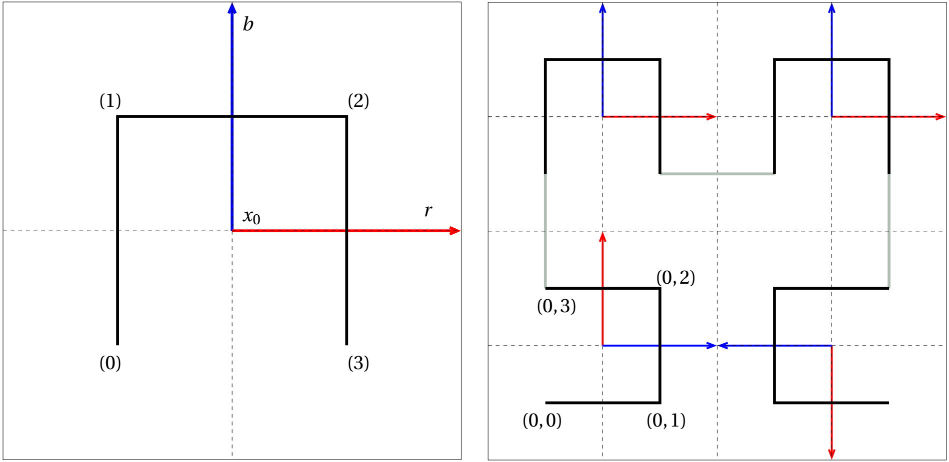


Figure: Curves $\mathcal{H}_1$ and $\mathcal{H}_2$.

Hilbert curves

Hilbert curves provide an ordering for points on a plane. Forget about how to connect adjacent sub-curves, and instead focus on how we can recursively enumerate the quadrants.
A local frame is associated to each quadrant: it consist in its center x_0 two orthogonal vectors b and r (see Figure 2). At the root level, enumerating the points is simple: proceed around the four quadrants, numbering them

$$(0) = x_0 - \frac{b+r}{2} \quad (1) = x_0 + \frac{b-r}{2} \quad (2) = x_0 + \frac{b+r}{2} \quad (3) = x_0 - \frac{b-r}{2}.$$

We want to determine the order we visit the sub-quadrants while maintaining the overall adjacency property. Examination reveals that each of the sub-quadrants curves is a simple transformation of the original pattern. Figure 2 illustrate the first level of that recursion.

Hilbert curves

Quadrant (0) is itself divided into four quadrants $(0, 0)$, $(0, 1)$, $(0, 2)$ and $(0, 3)$. Its center is simply set to (0) and two vectors b and r are changed as

$$b \leftarrow r/2 \text{ and } r \leftarrow b/2.$$

For quadrant $(0, 1)$ and $(0, 2)$ we have

$$b \leftarrow b/2 \text{ and } r \leftarrow r/2.$$

and finally for quadrant $(0, 3)$:

$$b \leftarrow -r/2 \text{ and } r \leftarrow -b/2.$$

creates 4 sub quadrants. If we consider a maximal recursion depth of d, each of the final subquadrants will be assigned to a set of d “coordinates” i.e. $(k_0, k_1, \ldots, k_d)$, k_j being 0,1,2 or 3.

Algorithm in Listings **??** compute the Hilbert coordinates of a given point x, y, starting from an initial quadrant define by its center x_0, y_0 and two orthogonal directions.

Hilbert curves

Each point x of $\mathbb{R}^2$ has its coordinates on the Hilbert curve. Sorting a point set with respect to Hilbert coordinates allow to ensure that two successive points of the set are close to each other. In the context of the Bowyer-Watson algorithm, this kind of data locality could potentially decrease the number of local searches N_{search} that were required to find the next invalid triangle.

Sets of 1000 and 10000 sorted points are presented on Figure 3. On the Figure, two successive points in the sorted list are linked with a line.

The main cost of sorting points is on the sorting algorithm itself and not on the computation of the Hilbert curve coordinates: sorting over a million points takes less than a second on a standard laptop. Table 1 present timings and statistics for the same point sets as in table ??, but while having sorted the points using the Hilbert curve.

n	10^3	10^4	10^5	10^6
N_{search}	2.34	2.46	2.50	2.50
N_{cavity}	4.06	4.13	4.16	4.17
t(sec)	0.0097	0.090	0.92	9.2

Table: Results of the `delaunayTrgl` algorithm applied to random points. Points were initially sorted through using a Hilbert sort.

Hilbert curves

The number of serarches is not increasing anymore with the size of the set. This is important: the complexity of the Delaunay triangulation algorithm now is linear in time. Of course, sorting points has a $n \log n$ complexity so that the overall process is in $n \log n$ as well. Yet, the relative cost of sorting the points is negligible with respect to the cost of the triangulation itself.

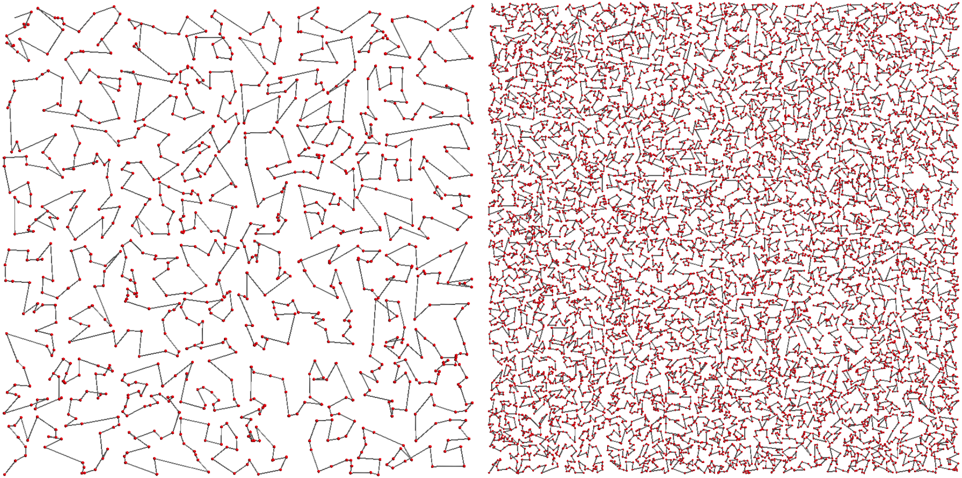


Figure: Hilbert sort of sets of 1000 and 10000 random points.