An Introduction to Delaunay-based Mesh Generation and Adaptation

Hang Si
si@wias-berlin.de

Weierstrass Institute for Applied Analysis and Stochastics (WIAS),
Berlin, Germany

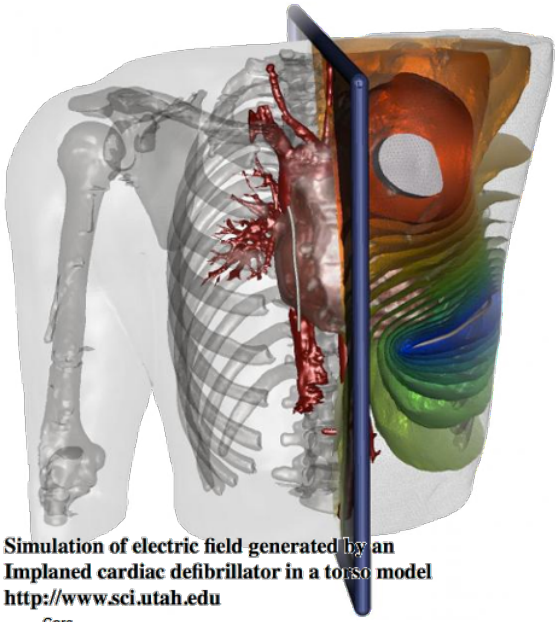
Stonybrook, NY, USA

Outline

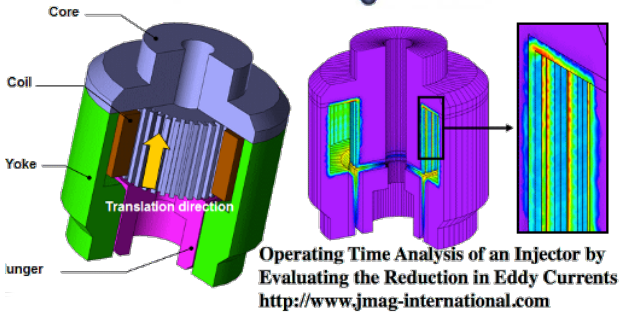
1. Introduction
2. Delaunay Triangulations
3. Constrained Mesh Generation
4. Mesh Adaptation
5. Further Topics

Motivation

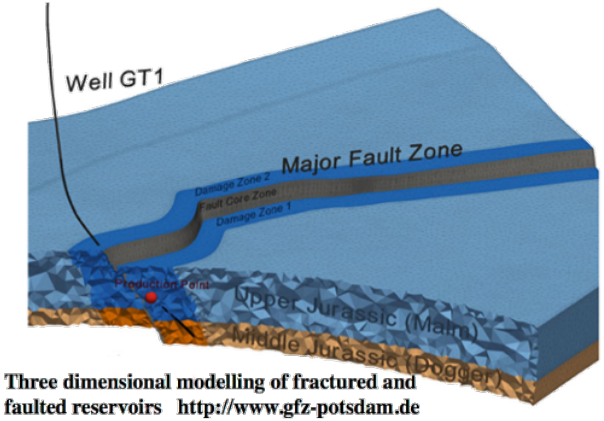
- Numerical simulation has become an important part of the mathematical modeling process and is possibly the unique way to reproduce or predicate the behavior of a complex real-life physic phenomenon.
- Mesh generation has become a key bottleneck in the numerical solution process.



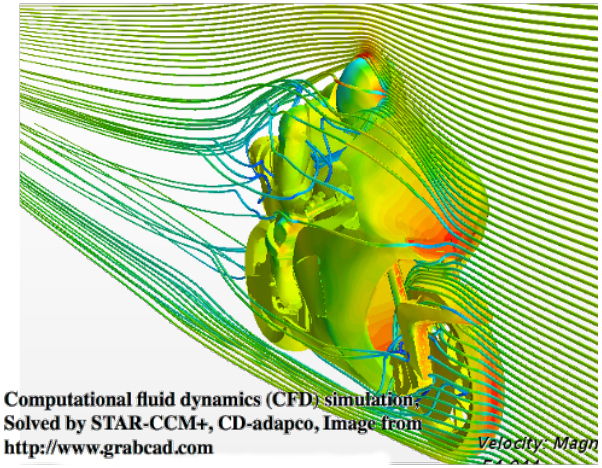
Simulation of electric field generated by an Implaned cardiac defibrillator in a torso model http://www.sci.utah.edu



Operating Time Analysis of an Injector by Evaluating the Reduction in Eddy Currents http://www.jmag-international.com



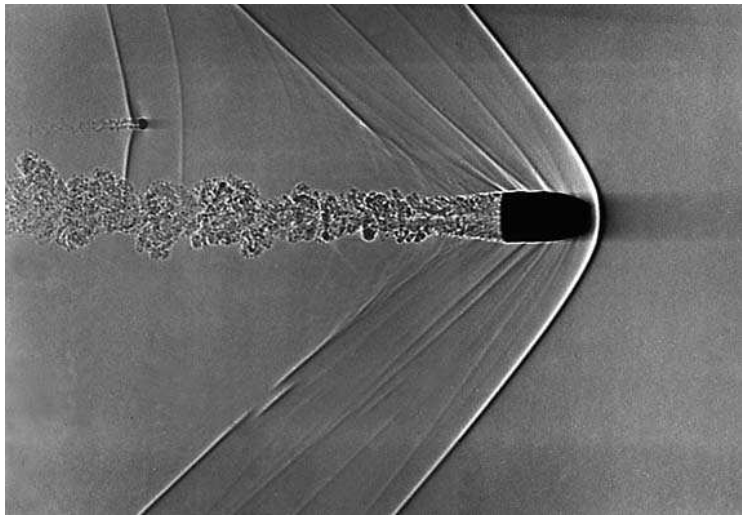
Three dimensional modelling of fractured and faulted reservoirs http://www.gfz-potsdam.de

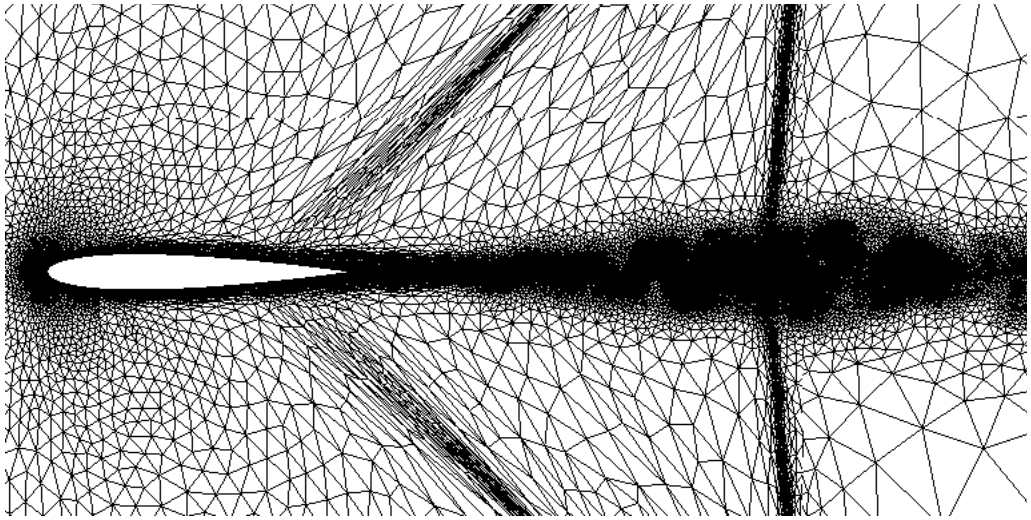


Computational fluid dynamics (CFD) simulation; Solved by STAR-CCM+, CD-adapco, Image from http://www.grabcad.com

Mesh Adaptation

- Many physical problems exhibit anisotropic features. Examples include particular convection-dominated problems whose solutions have, e.g., boundary layers, shocks, edge or corner singularities.
- When numerical methods are used to approximate these problems, it is of great importance that the used meshes represent such features to achieve high accuracy at a low computational cost.



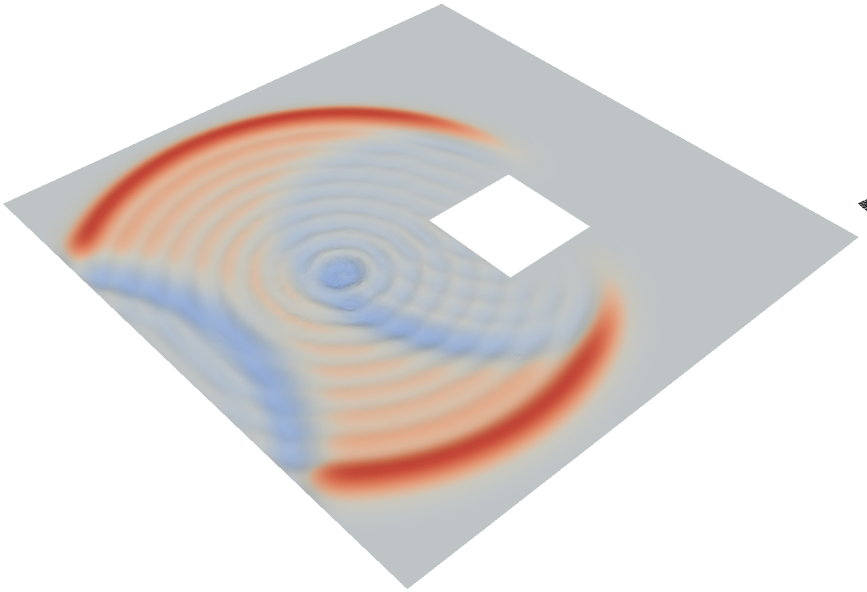


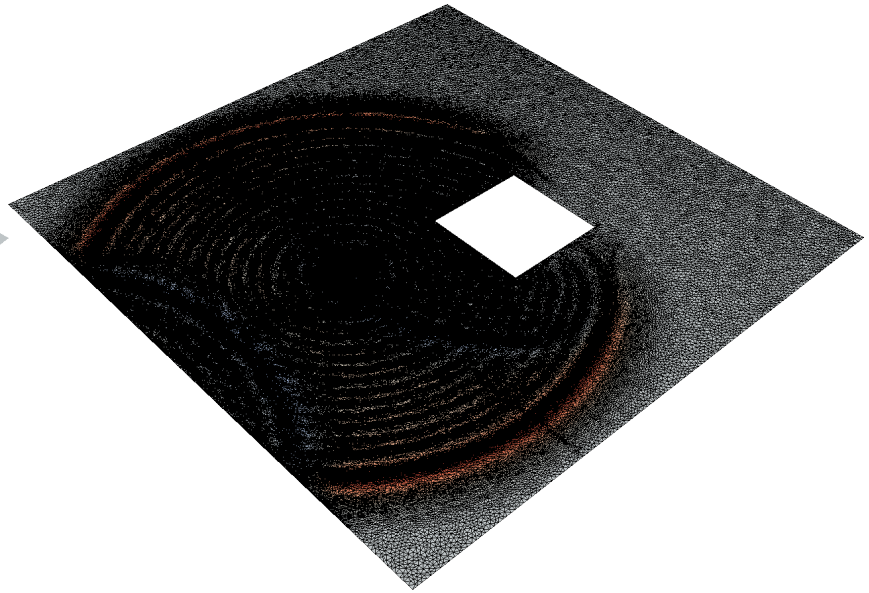
Images from Frey's IMR talk (2005)

Example: Adaptive FEM for the Wave Equation

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} - \mu \Delta u &= f \qquad \text{in } \Omega\,, \\ u &= 0 \qquad \text{in } \partial\Omega\,, \end{cases}$$

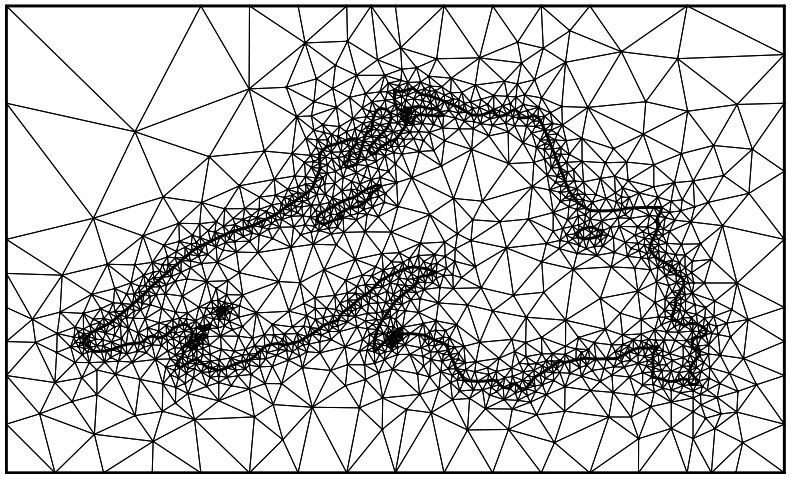
here $\mu = 1.$, f discrete Dirac function.



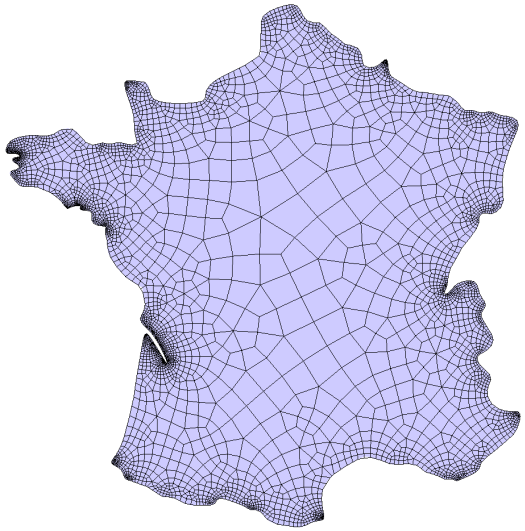


Mesh Generation in $\mathbb{R}^2$

- Well-studied, e.g., in graph theory, discrete and computational geometry.
- Efficient and optimal algorithms are available.
- Theoretical guarantees on good-quality, adapted, anisotropic meshes.
- Robust softwares, e.g. `Triangle` (UC Berkeley), `CGAL` (INRIA).



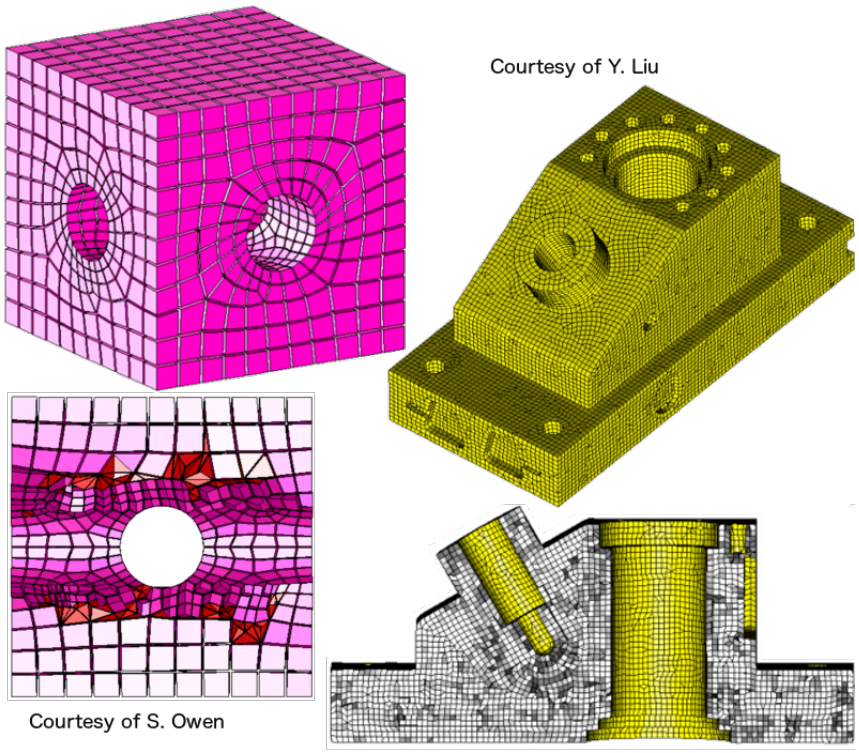
Ruppert 1995, Schewchuk 1996

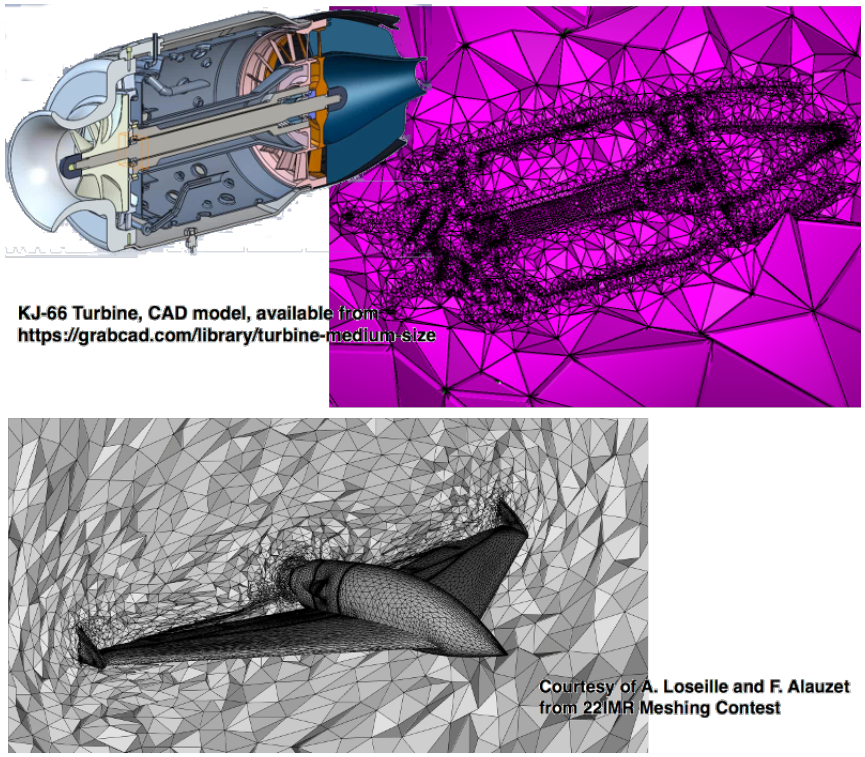


Pellenard & Alliez 2011

Mesh Generation in $\mathbb{R}^3$

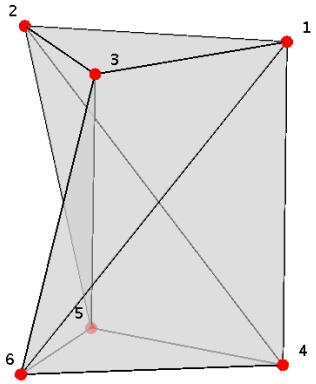
- It is still a very difficult problem, both in surface and volume meshes.
- Many open problems in the fields of discrete and differential geometry.
- Most of the computational problems have proven to be NP-hard.



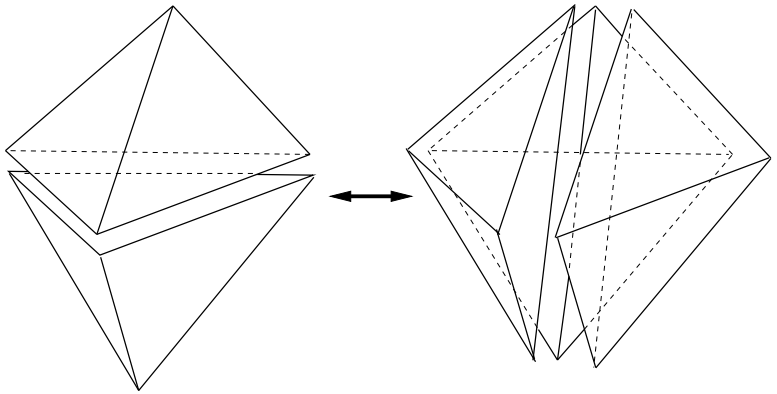


In 1998, **Tim Baker** (1948–2006) wrote:

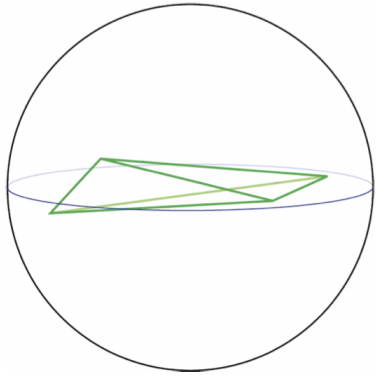
> *In three dimensions the theory is far from developed. The main difficulties are the following:* **(1)** *there exist configurations of boundary points and faces for which no conforming grid of tetrahedra exists unless extra points are inserted,* **(2)** *although 3D analogues of diagonal swapping exist, it does not appear possible to convert an arbitrary triangulation into the corresponding Delaunay triangulations,* **(3)** *the presence of slivers, formed by four coplanar points, can arise and indeed will often arise when efforts are made to create a constrained Delaunay triangulation that conforms with a prescribed boundary.*



The Schönhardt polyhedron

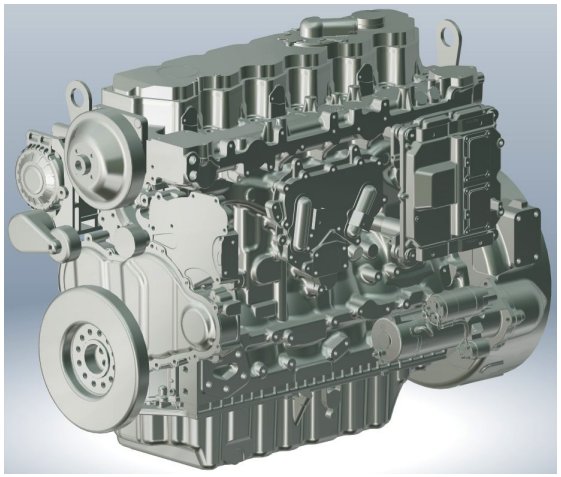


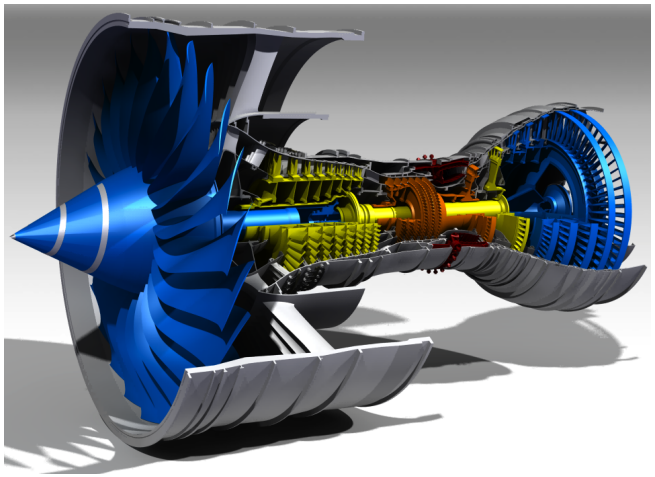
flip 2-to-3
3-to-2



Sliver

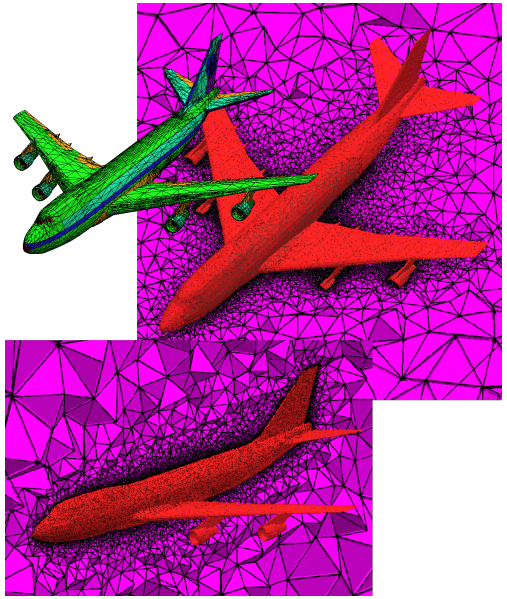
- 3D models can be arbitrarily complex, e.g., they may consist of different and connect parts, may contain special geometric features, like sharp features (small angles), thin layers, etc.
- The construction of efficient and particularly robust mesh generation techniques can only be achieved through algorithms with solid theoretical justification.



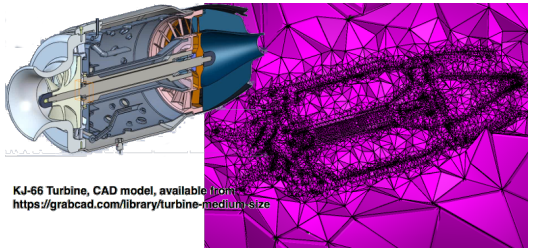


CAD models, freely available from `http://www.grabcad.com`.

The TetGen Project

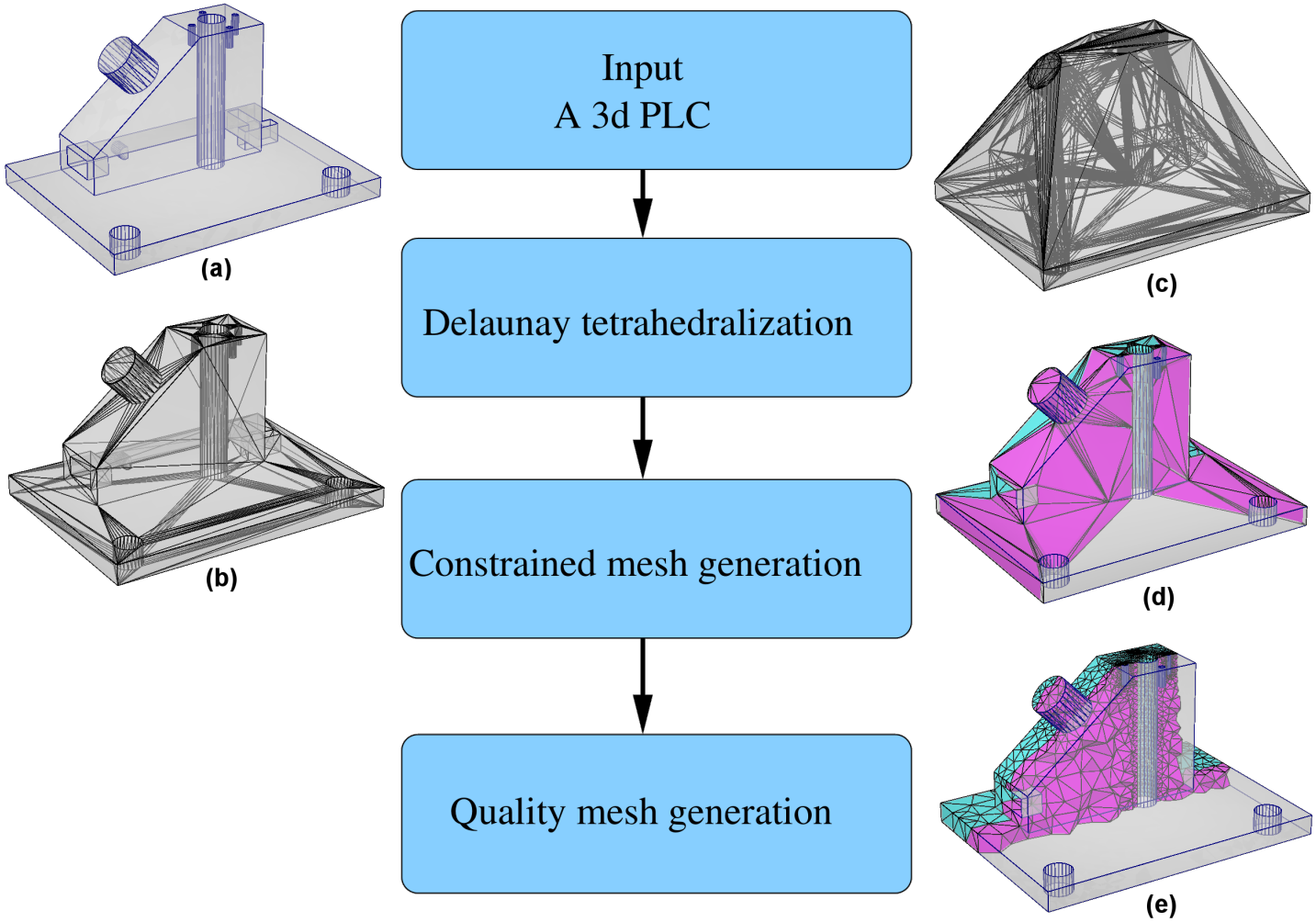


- A research project of WIAS since 2002.
- The goal is two-fold:
  - to study the underlying mathematical problems; and
  - to develop robust and efficient algorithms and softwares.
- It is freely available at `http://www.tetgen.org`.
  - latest version 1.5 (released in Nov. 2013).
  - about 10,000 downloads (Nov. 2013 - now).
  - about 20+ commercial licenses.



- H. Si, TetGen, a Delaunay-based Tetrahedral Mesh Generator, ACM Trans. Math. Softw., **41** (2):11:1–11:36, February **2015**.

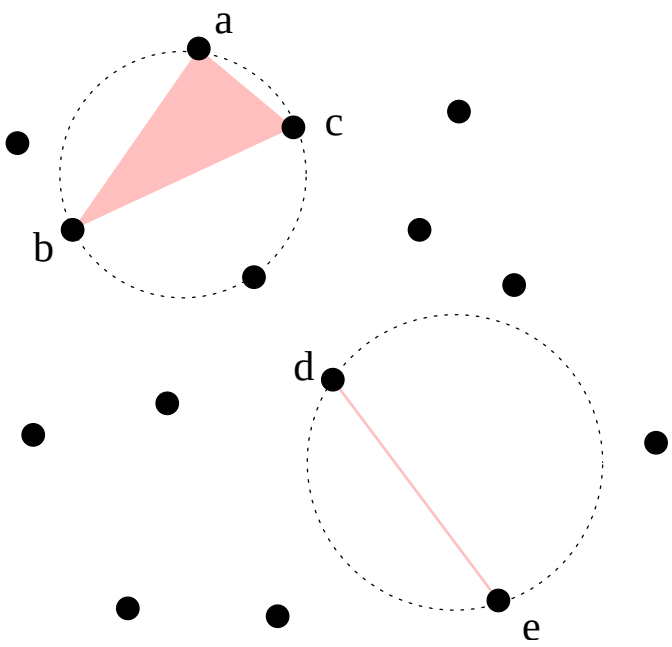
The Meshing Process



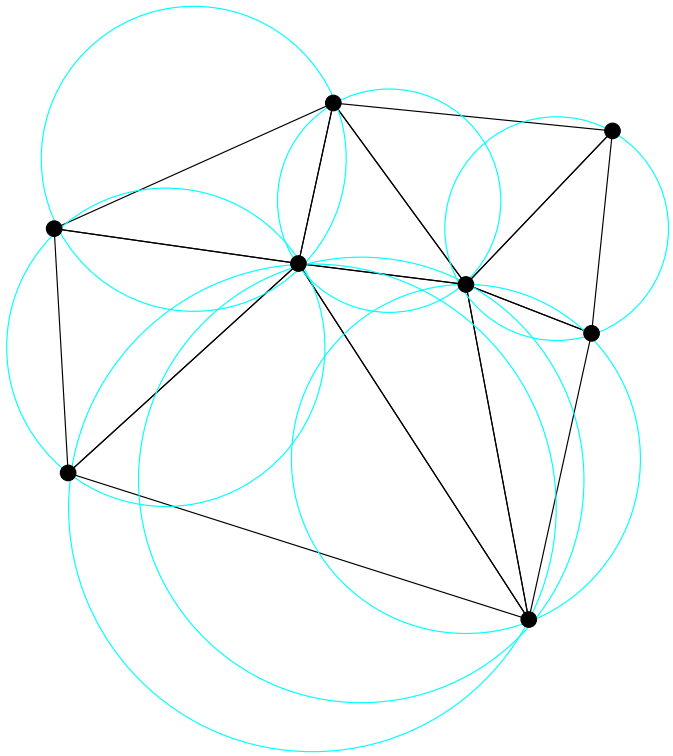
Delaunay Triangulations

The Empty Sphere Property

- A simplex σ whose vertices are in $S \subset \mathbb{R}^d$ is Delaunay if it has a circumscribed sphere such that it contains no other vertex of S in its inside.
- If S is in general position, all Delaunay simplices of S form the Delaunay triangulation of S [**Delaunay 1934**].



Delaunay simplices



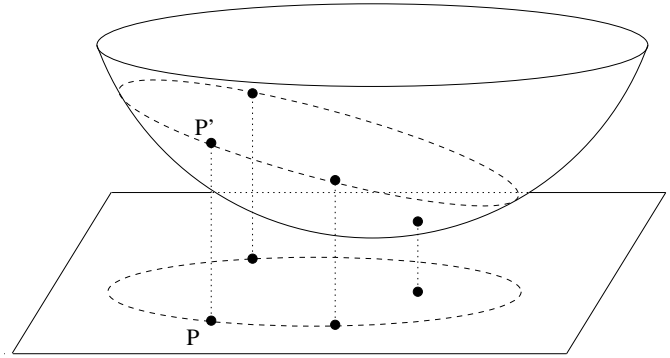
Delaunay triangulation

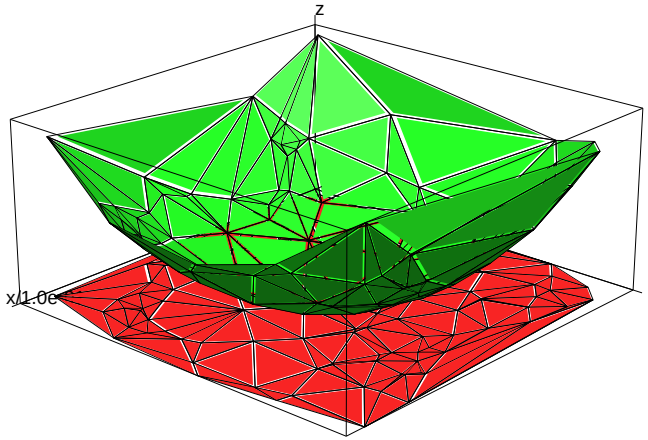
Lifting and Projection

- Delaunay triangulation of $S \subset \mathbb{R}^d$ can be obtained by first lifting every vertex $\mathbf{x} = (x_1, x_2, \cdots, x_d)$ in S into a vertex $\mathbf{x}' = (x_1, x_2, \cdots, x_d, x_{d+1})$ in $\mathbb{R}^{d+1}$ by letting the last coordinates (its "height") be

$$x_{d+1} := x_1^2 + x_2^2 + \cdots + x_d^2,$$

then taking the orthogonal projection of the convex hull of new point set $S' \subset \mathbb{R}^{d+1}$.



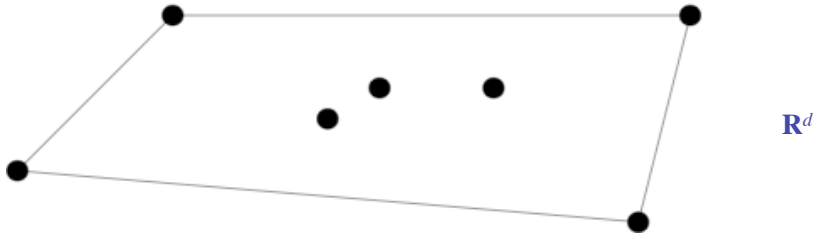


Regular Subdivisions

- Every piecewise linear function $f : Q \to \mathbb{R}$ over a polytope Q determines a polytope projection, by setting:

$$P := \mathrm{conv}\{\begin{pmatrix} \mathbf{x} \\ f(\mathbf{x}) \end{pmatrix} \; : \; \mathbf{x} \in Q\}.$$

The orthogonal projection of the lower envelope of P determines a regular subdivision of Q.

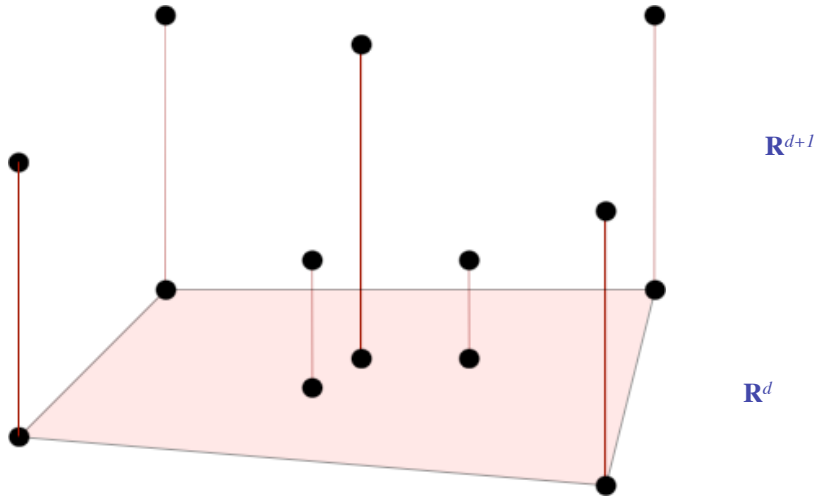


Regular Subdivisions

- Every piecewise linear function $f : Q \to \mathbb{R}$ over a polytope Q determines a polytope projection, by setting:

$$P := \mathrm{conv}\{\begin{pmatrix} \mathbf{x} \\ f(\mathbf{x}) \end{pmatrix} \; : \; \mathbf{x} \in Q\}.$$

The orthogonal projection of the lower envelope of P determines a regular subdivision of Q.

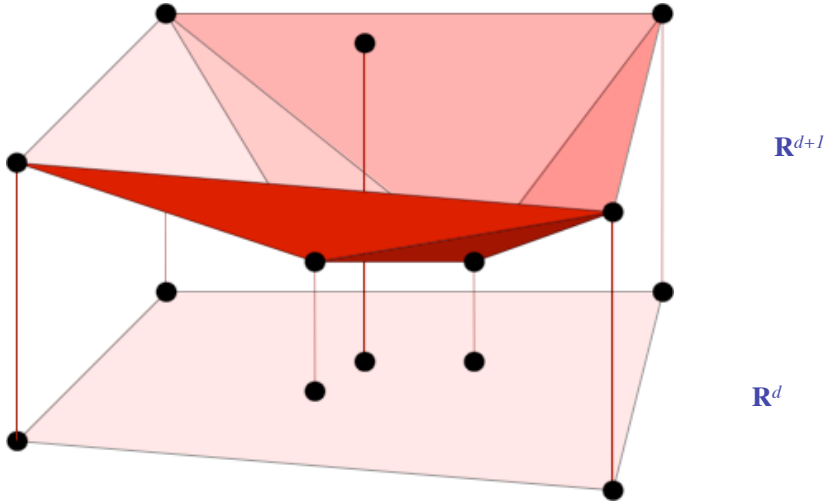


Regular Subdivisions

- Every piecewise linear function $f : Q \to \mathbb{R}$ over a polytope Q determines a polytope projection, by setting:

$$P := \mathrm{conv}\{\begin{pmatrix} \mathbf{x} \\ f(\mathbf{x}) \end{pmatrix} \; : \; \mathbf{x} \in Q\}.$$

The orthogonal projection of the lower envelope of P determines a regular subdivision of Q.

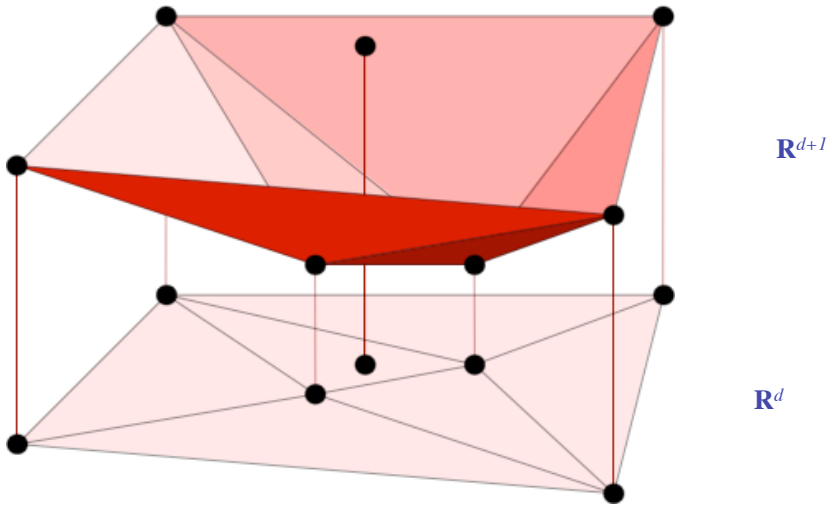


Regular Subdivisions

- Every piecewise linear function $f : Q \to \mathbb{R}$ over a polytope Q determines a polytope projection, by setting:

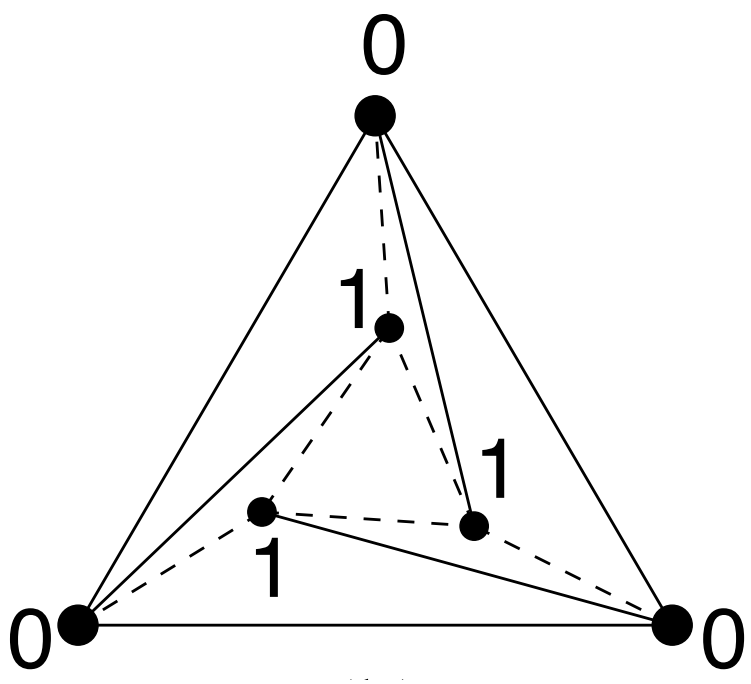
$$P := \mathrm{conv}\{\begin{pmatrix} \mathbf{x} \\ f(\mathbf{x}) \end{pmatrix} \; : \; \mathbf{x} \in Q\}.$$

The orthogonal projection of the lower envelope of P determines a regular subdivision of Q.



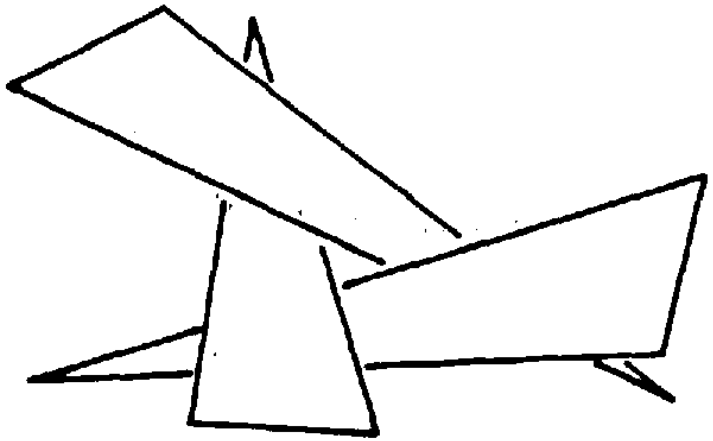
Non-regular subdivisions

- A subdivision of a point set S is *non-regular* if it is not a regular subdivision of S.
- **Remarks**: There are many non-regular triangulations. We only know few of them.



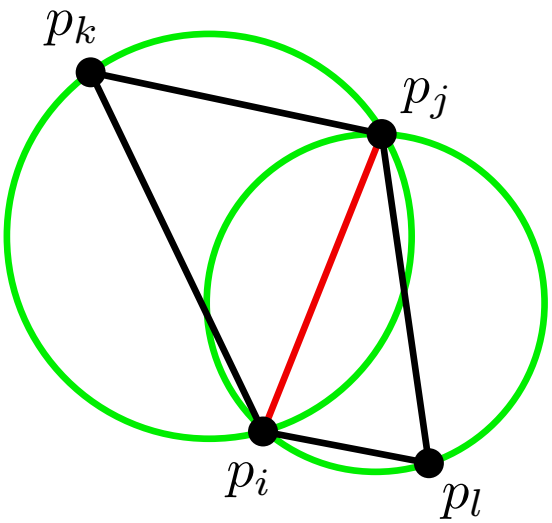
The Acyclic Theorem

- The in_front/behind relation: Let $\mathbf{x}$ be a point and P and Q be two disjoint convex objects in $\mathbb{R}^d$. We say that P *is in front of* Q *with respect to* $\mathbf{x}$ if there is a ray L starting at $\mathbf{x}$ that first passes through P and then through Q.
- **Theorem [Edelsbrunner 1990]**: The in_front/behind relation defined for the faces of any regular subdivision and for fixed viewpoint $\mathbf{x}$ in $\mathbb{R}^d$ is acyclic.

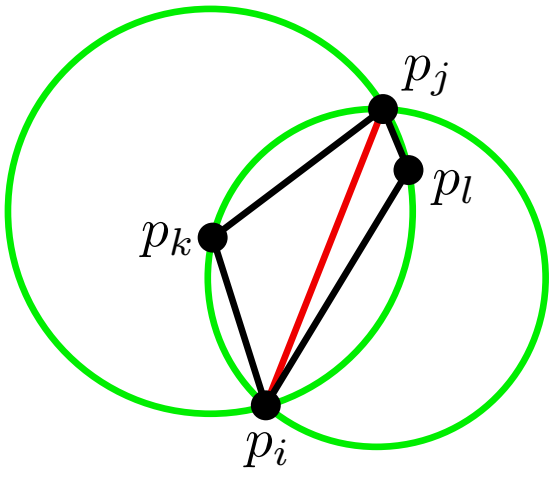


Locally Delaunay

- An edge is locally Delaunay if two triangle sharing it have no vertex inside each others' circumcirlces.



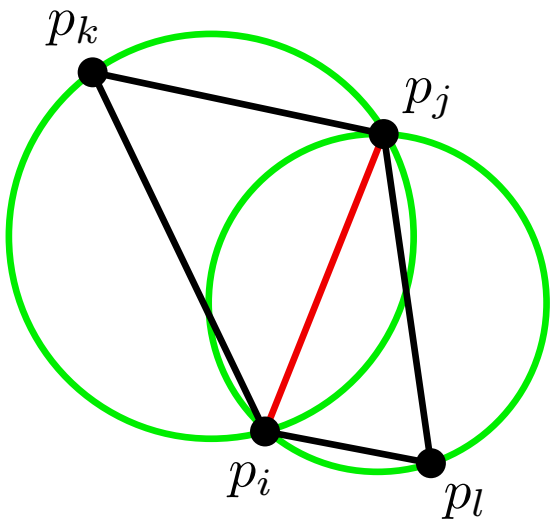
The edge $p_i p_j$ is locally Delaunay



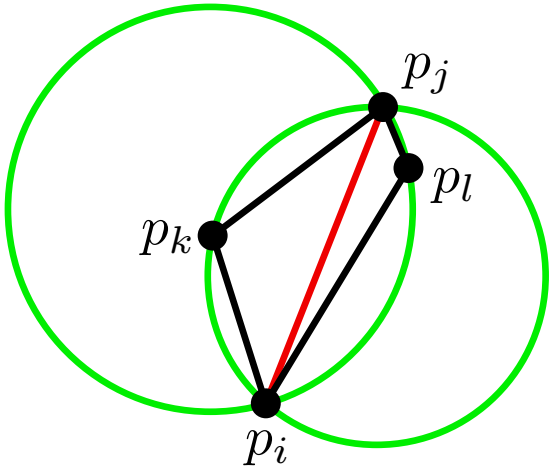
The edge $p_i p_j$ is not locally Delaunay

Locally Delaunay

- An edge is locally Delaunay if two triangle sharing it have no vertex inside each others' circumcirlces.
- **Delaunay Lemma**: If every edge of $\mathcal{K}$ is locally Delaunay, then $\mathcal{K}$ is a Delaunay triangulation of S.



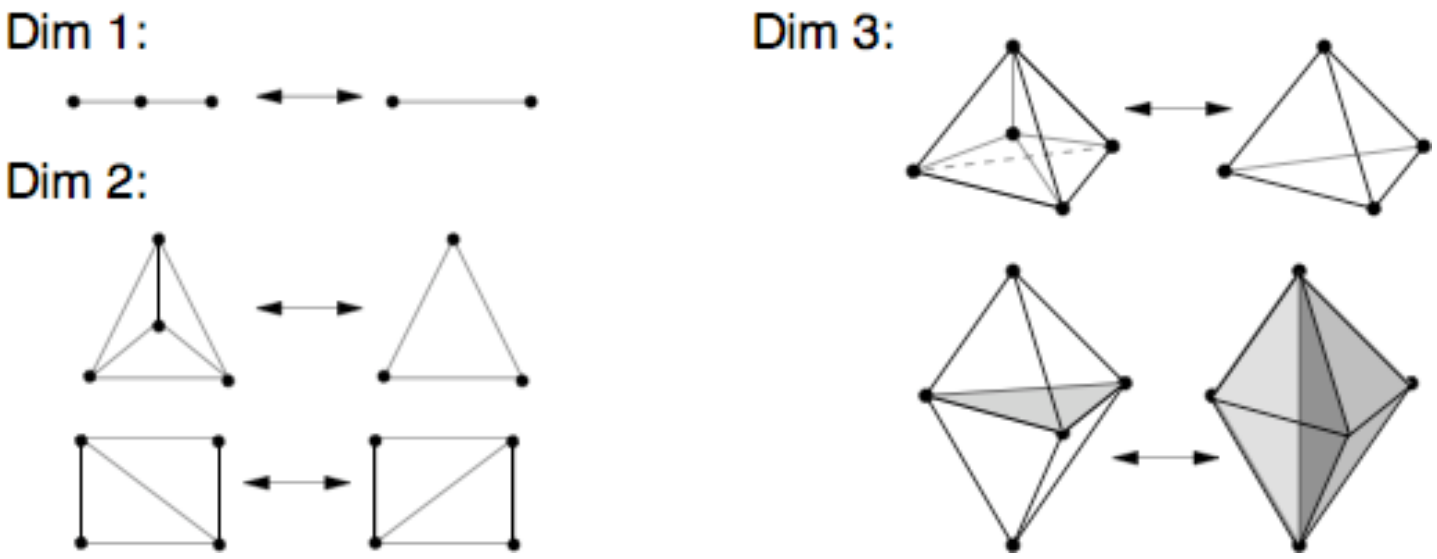
The edge $p_i p_j$ is locally Delaunay



The edge $p_i p_j$ is not locally Delaunay

Flips

- Flips (bistellar moves or Pachner moves) are essentially the exchange between the two possible triangulations on a subset of $d+2$ points.



Delaunay Flip

- **Theorem** If an edge is not locally Delaunay, and it can be flipped, then the resulting new edge is locally Delaunay.

$$\overline{ab} \text{ is not locally Delaunay} \Longleftrightarrow \overline{pq} \text{ is locally Delaunay}$$

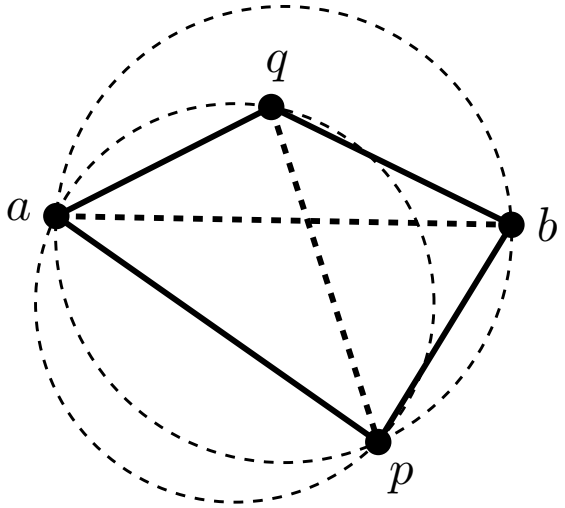
$\overline{ab}$ is not locally Delaunay

$\Longleftrightarrow q \in int(C_{abp})$

$\Longleftrightarrow \widehat{aqp} > \widehat{abp}$

$\Longleftrightarrow b \in ext(C_{apq})$

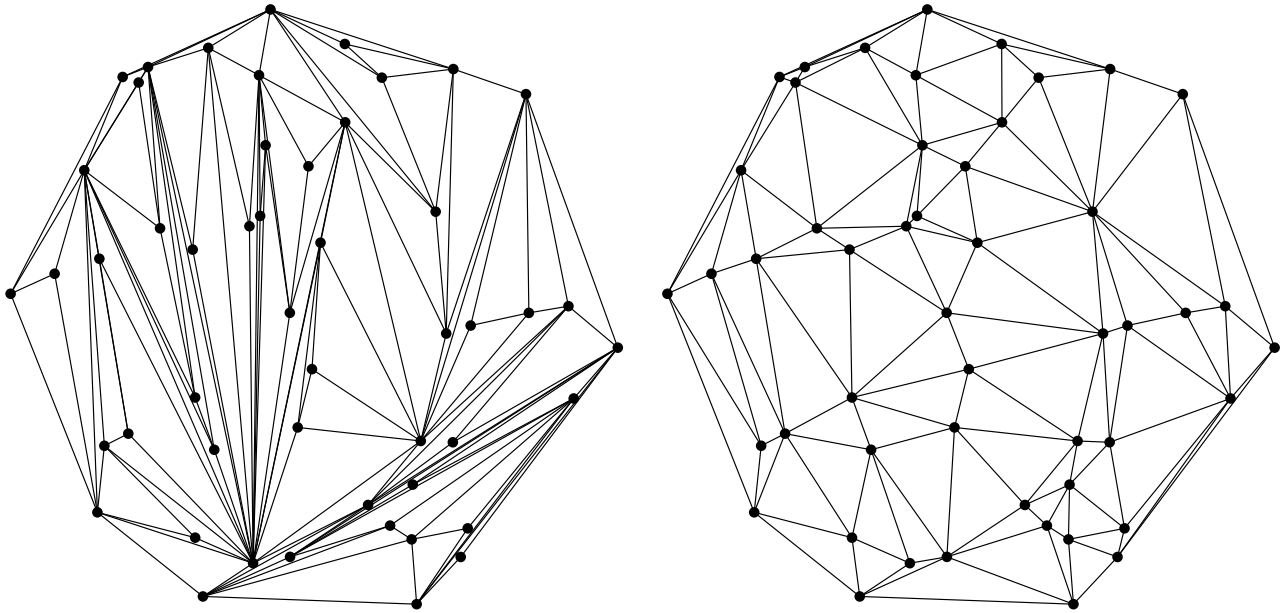
$\Longleftrightarrow \overline{pq}$ is locally Delaunay



Lawson Flip Algorithm [**Lawson 1971**]

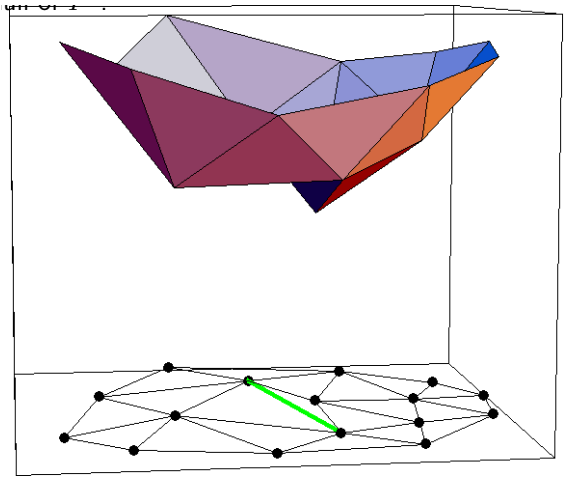
- Let $S = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ be a finite set of points in $\mathbb{R}^2$.
- Compute an initial triangulation $\mathcal{T}$ of a point set S.

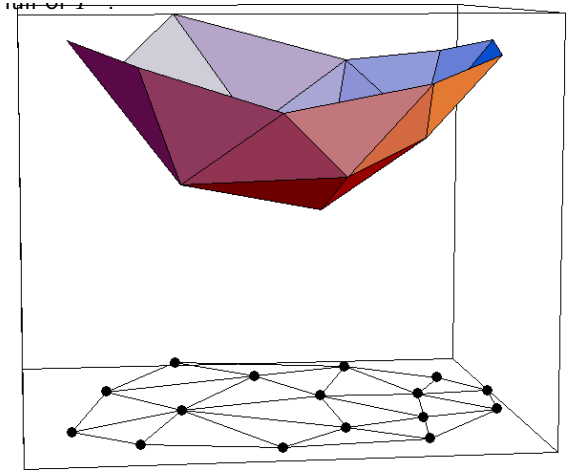
while $\exists$ a locally non-Delaunay edge $\mathbf{ab} \in \mathcal{T}$
  flip **ab**;
end while



Lawson Flip Algorithm [**Lawson 1971**]

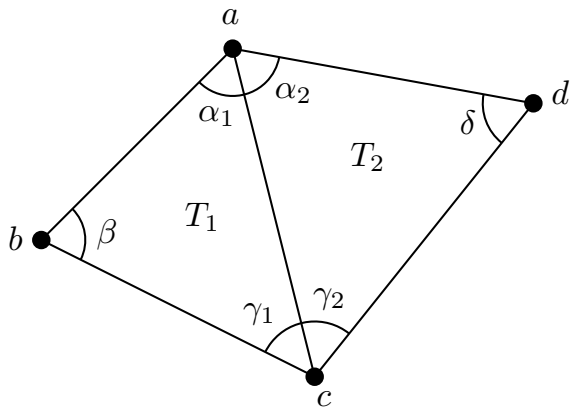
- **Correctness**:
  Flipping **ab** is like gluing the tetrahedron $\mathbf{a}'\mathbf{b}'\mathbf{c}'\mathbf{d}'$ below the lifted edge $\mathbf{a}'\mathbf{b}'$.
  Once **ab** is flipped, it will never be introduced again.
- **Runtime**:
  There are at most as many flips as there are edges connecting n points, i.e., $\binom{n}{2}$.
  The total running time is $O(n^2)$.

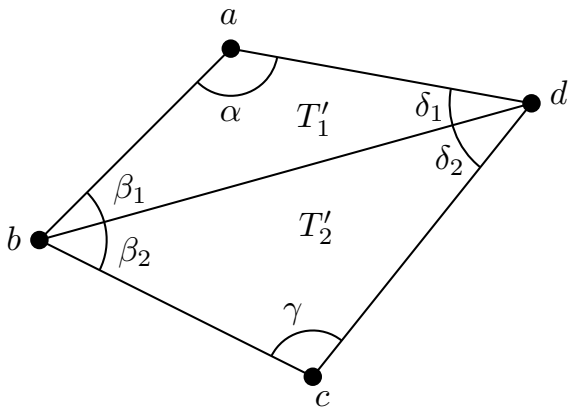




Lawson Flip Algorithm [**Lawson 1971**]

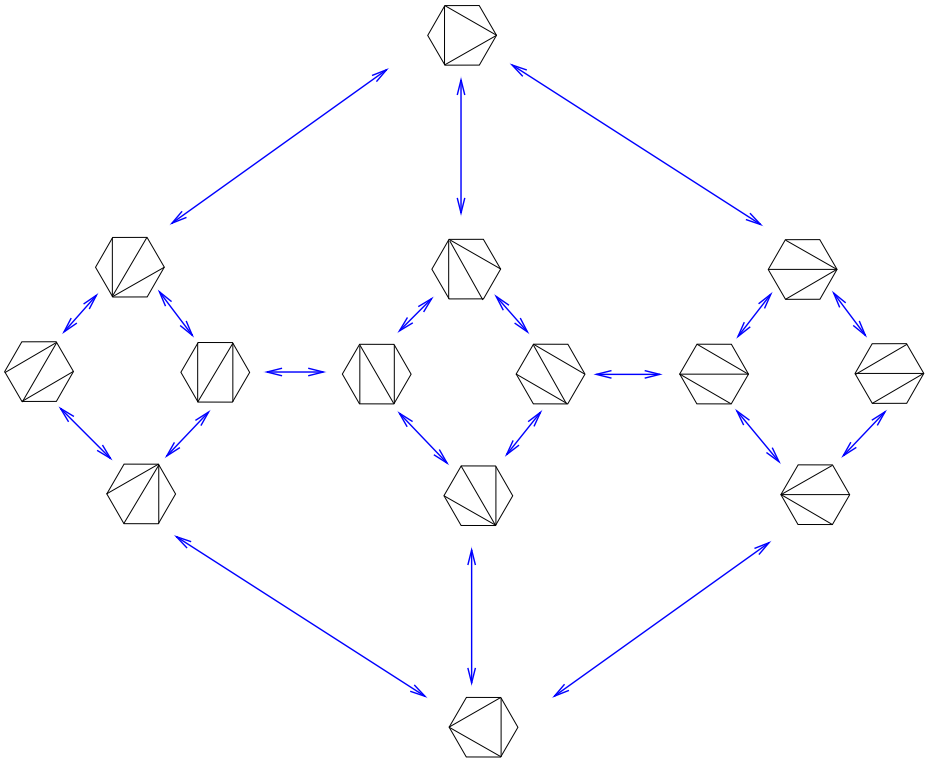
- **Maxmin Angle Property** Among all triangulation of a finite set $S \subset \mathbb{R}^2$, the Delaunay triangulation of S maximise the minimum angle.





Flip Graph

- The flip graph of a point set S: each vertex represents a triangulation of S, each edge represents a flip between two triangulations of S.
- The **Question**: is this graph connected?



The flip graph of the hexagon.

The Flip Graph Problem in $\mathbb{R}^2$

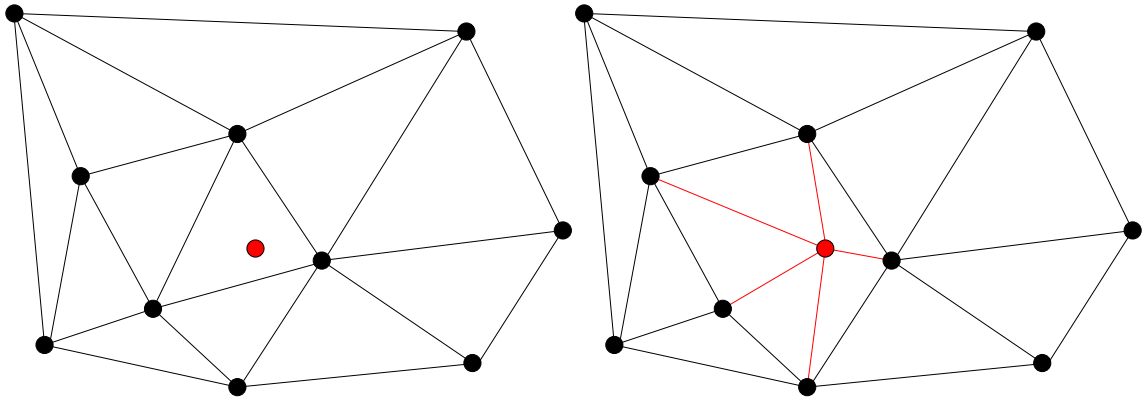
For every point set in $\mathbb{R}^2$:

- The flip graph of all triangulations is connected.
- You can always flip from one triangulation to another triangulation within $O(n^2)$ flips, e.g., using the Lawson's flip algorithm.
- The diameter of the graph is at most $4n$ (Lawson's algorithm is slow!)
- There is at least (expected) $n - 3$ edges are flippable.

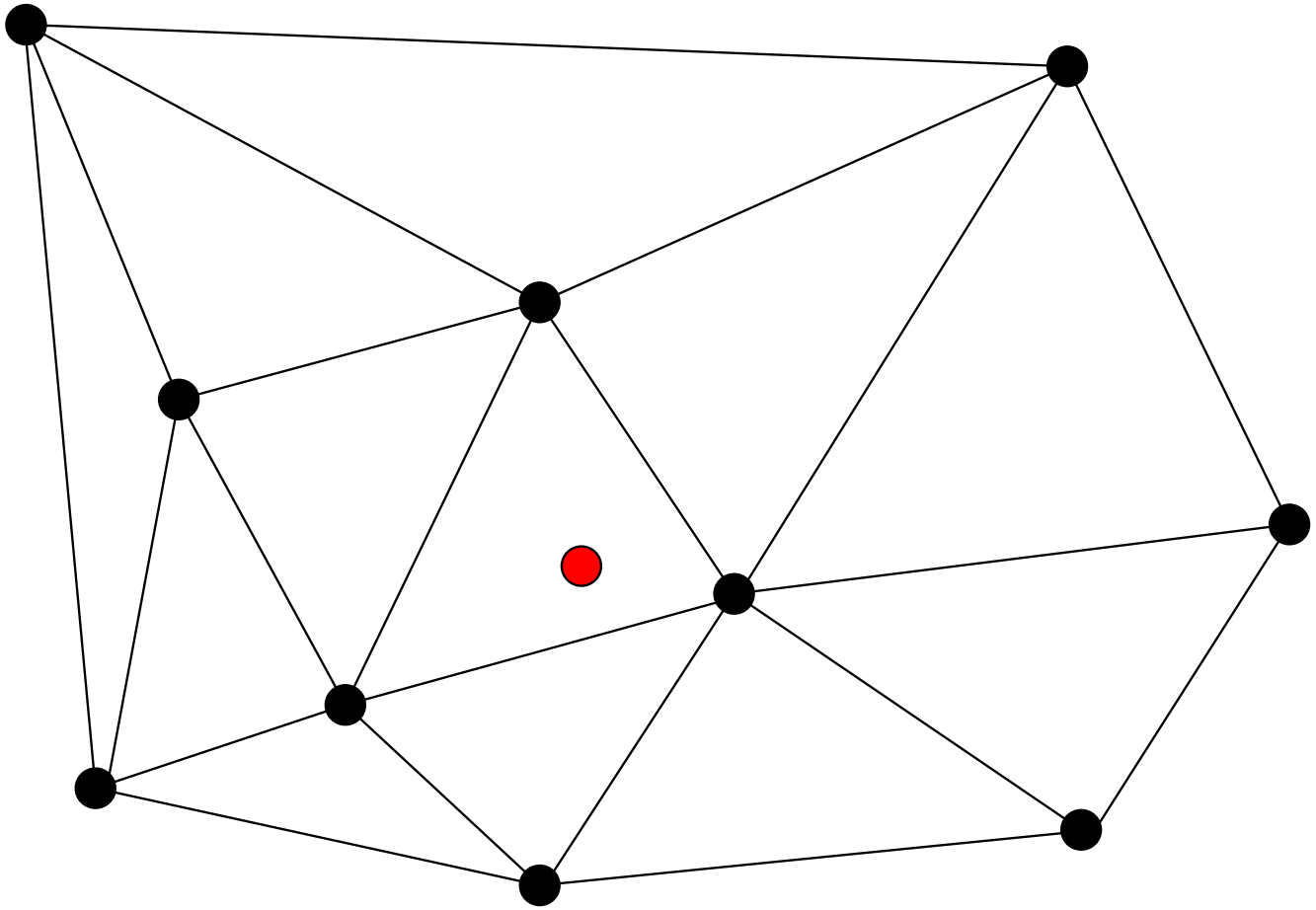
Incremental Flip Algorithm

- Let $S = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ be a finite set of points in $\mathbb{R}^3$.
- Let $[\mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}]$ be a sufficiently large tetrahedron that contains all points of S.

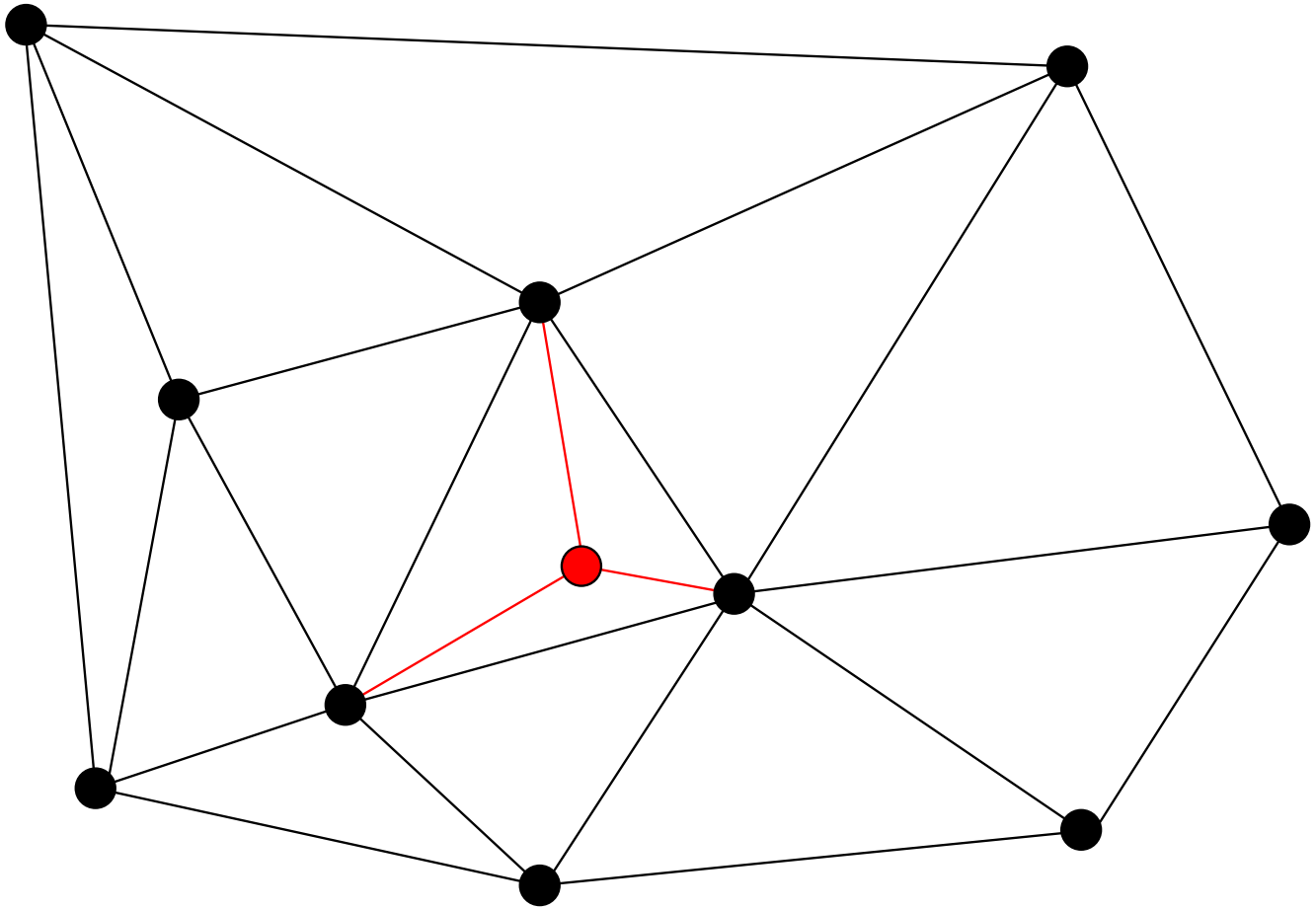
```
Let 𝒟_0 consists of only the tetrahedron [w, x, y, z];
for i = 1 to n do
    find [p, q, r, s] ∈ 𝒟_i that contains p_i;
    add p_i with a 1-to-4 flip;
    while ∃ triangle [a, b, c] not locally Delaunay;
        flip [a, b, c];
    endwhile
endfor
```



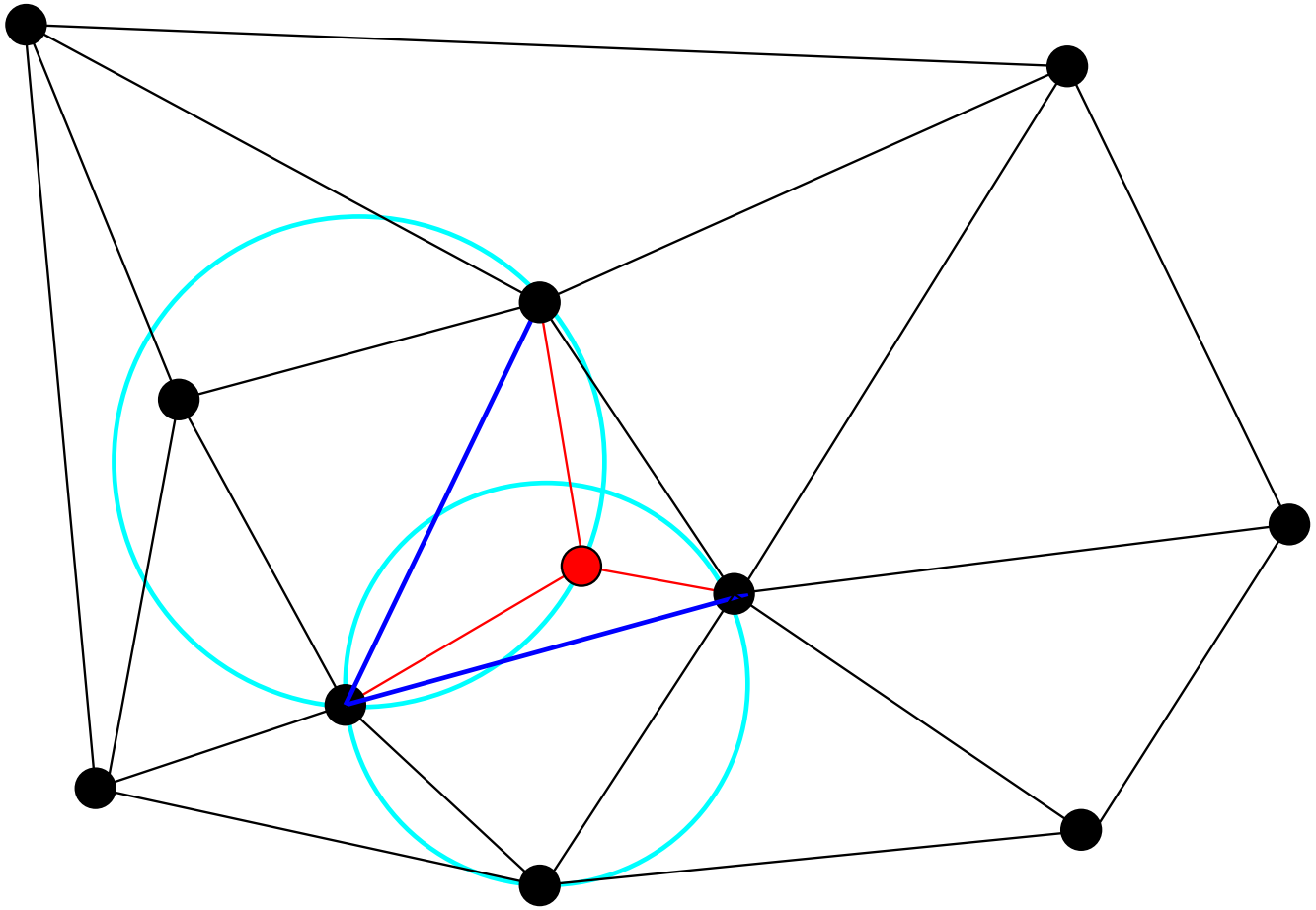
Incremental Flip Algorithm in $\mathbb{R}^2$



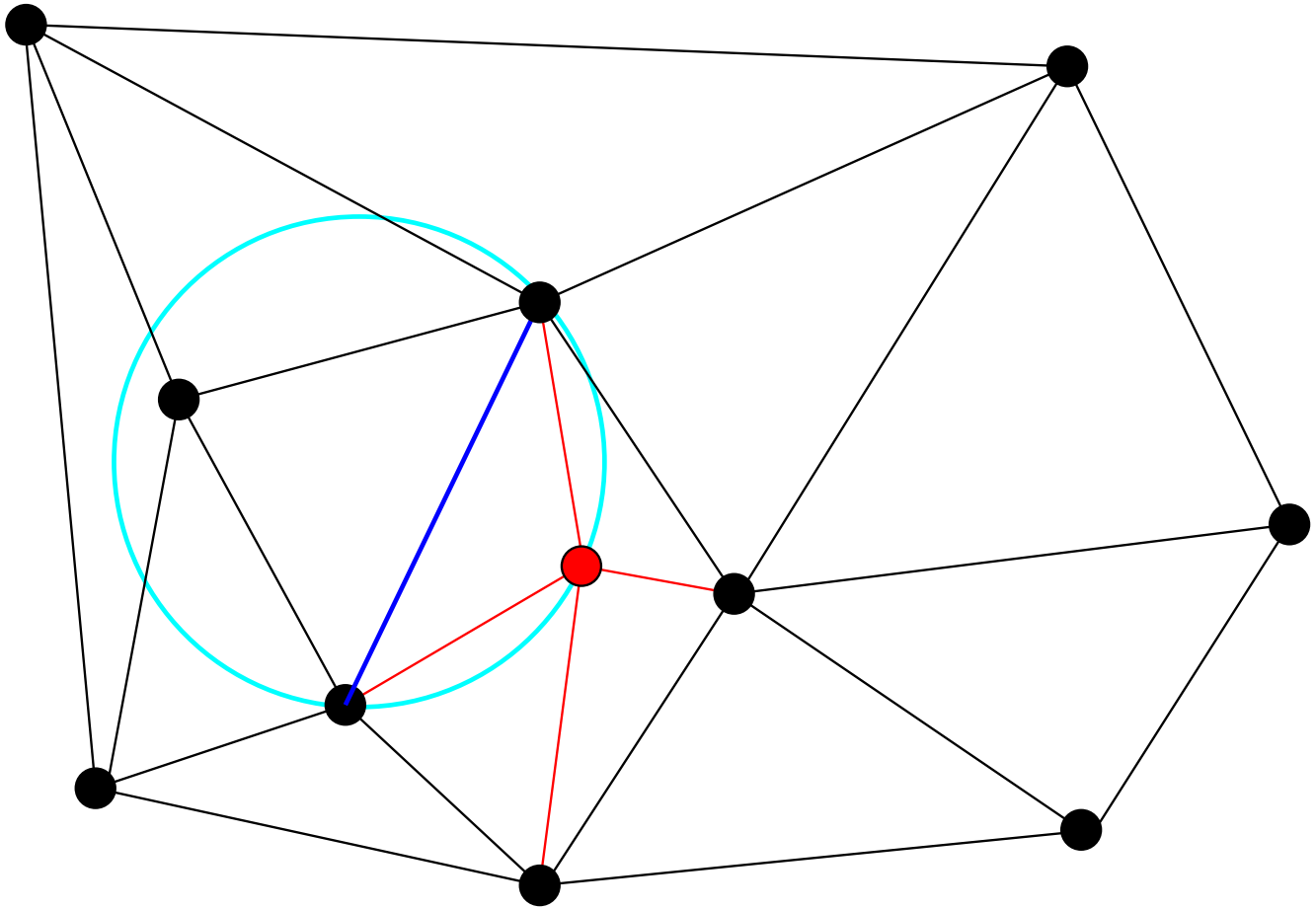
Incremental Flip Algorithm in $\mathbb{R}^2$



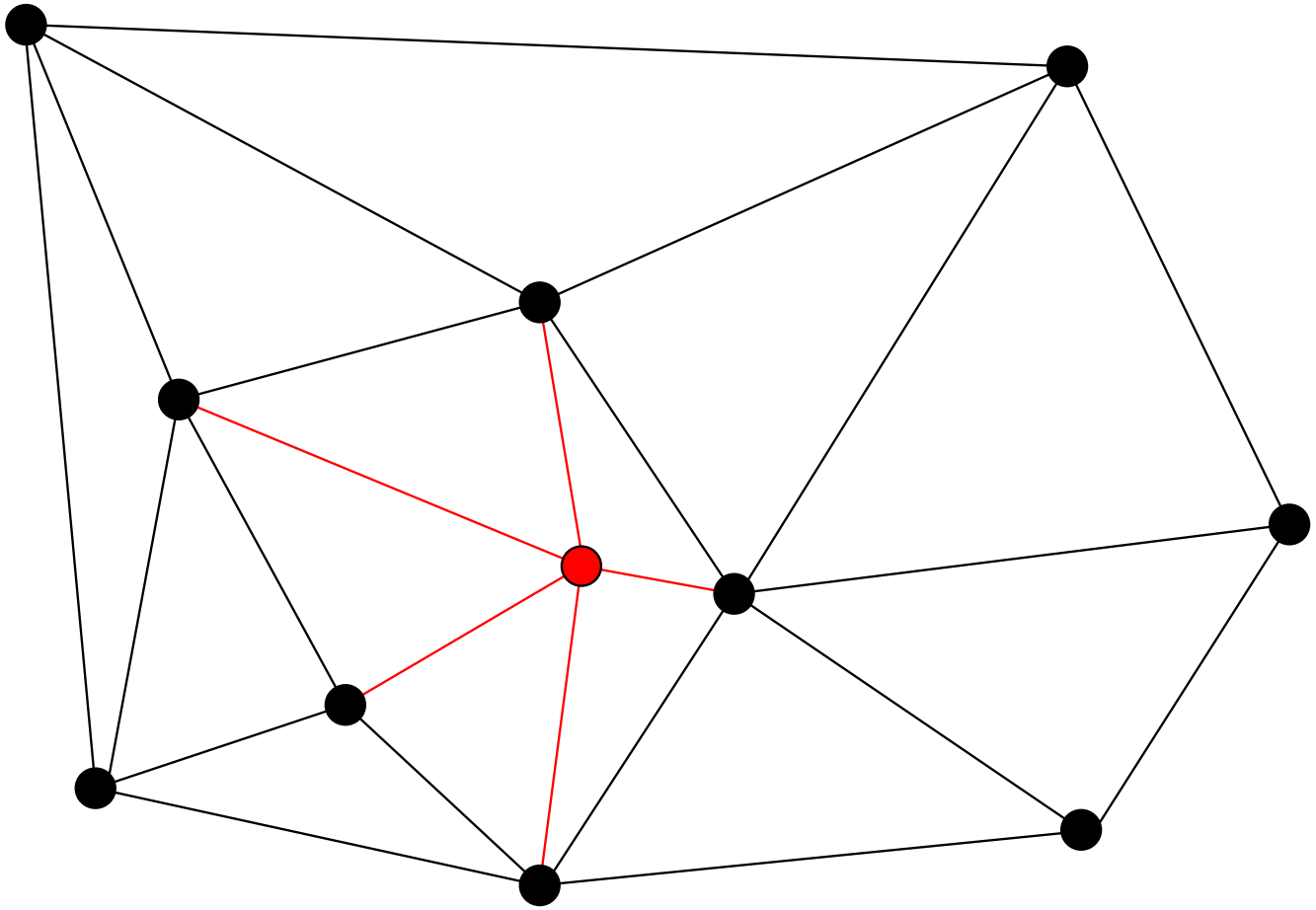
Incremental Flip Algorithm in $\mathbb{R}^2$



Incremental Flip Algorithm in $\mathbb{R}^2$

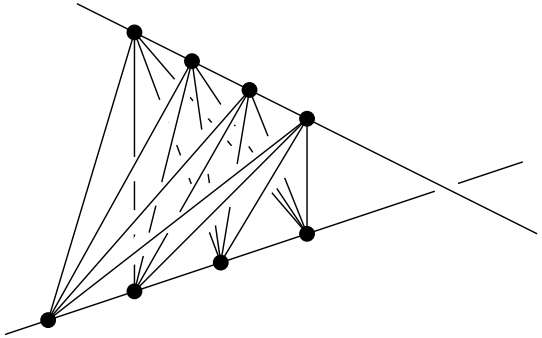


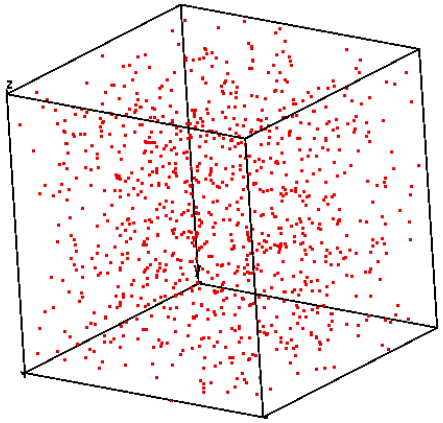
Incremental Flip Algorithm in $\mathbb{R}^2$



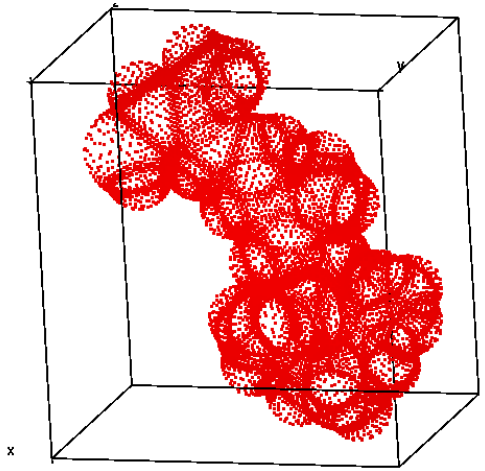
Incremental Flip Algorithm

- This algorithm is correct for creating Delaunay and regular triangulations in $\mathbb{R}^d$ [**Edelsbrunner & Shah 1996**].
- In $\mathbb{R}^2$, if points are inserted in a random order, the expected runtime of this algorithm is $O(n \log n)$ [**Clarkson & Shor 1986**].
- In $\mathbb{R}^3$, this worst runtime of this algorithm in $O(n^2)$ – The **Upper Bound Theorem** [**MacMullen 1970**] in convex polytopes.
- For point sets with a random or uniform distributions in $\mathbb{R}^3$, the expected runtime of the randomised incremental flip algorithm is $O(n \log n)$ [**Edelsbrunner & Shah 1996**].



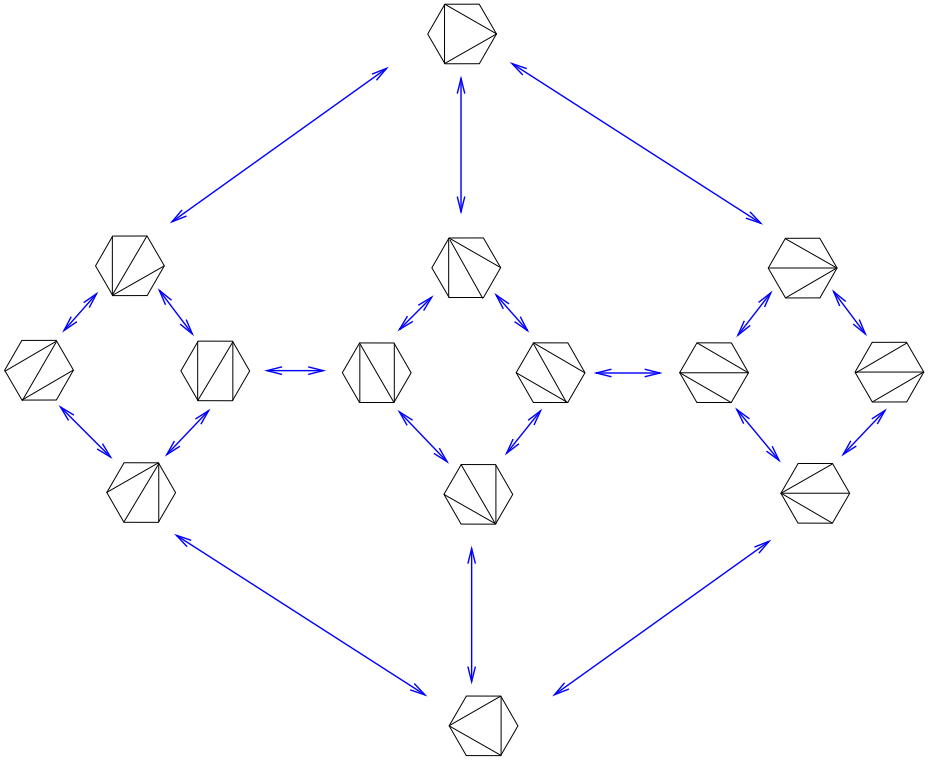


random



uniform

Flip Graph



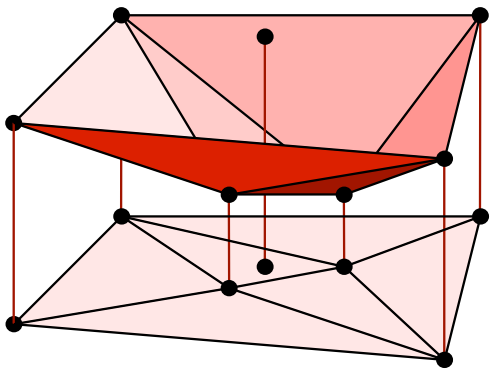
Flip Graph of Regular Triangulations

- **Theorem[Gelfand-Kapranov-Zelevinskii 1990]**: For every set A of n points in $\mathbb{R}^d$, there is a polytope $\Sigma(A)$ of dimension $n-d-1$ with the following correspondence:

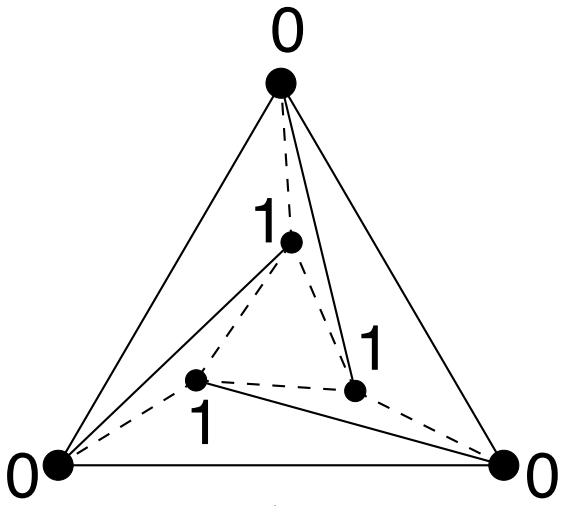
$$\begin{array}{rcl} \text{regular triangulation of A} & \longleftrightarrow & \text{vertices of } \Sigma(A) \\ \text{flips between them} & \longleftrightarrow & \text{edges of } \Sigma(A) \\ \text{poset of regular triangulation of A} & \longleftrightarrow & \text{poset of faces of } \Sigma(A) \end{array}$$

  This is called the **secondary polytope** of A.

- **Corollary**: The flip graph of regular triangulations is connected.



regular

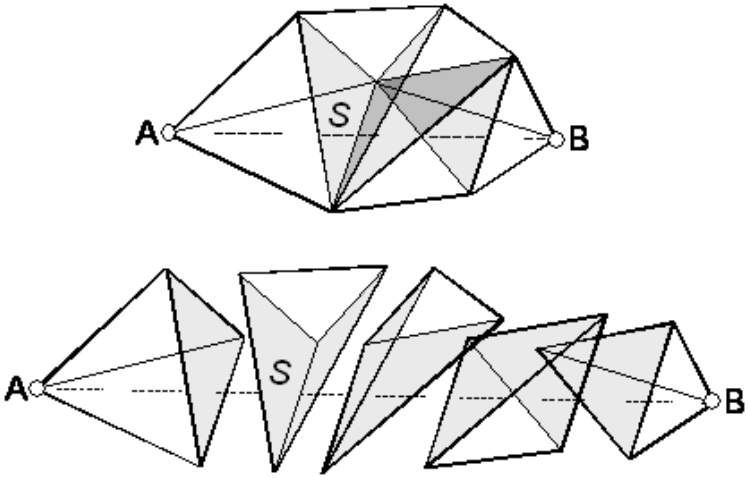


non-regular

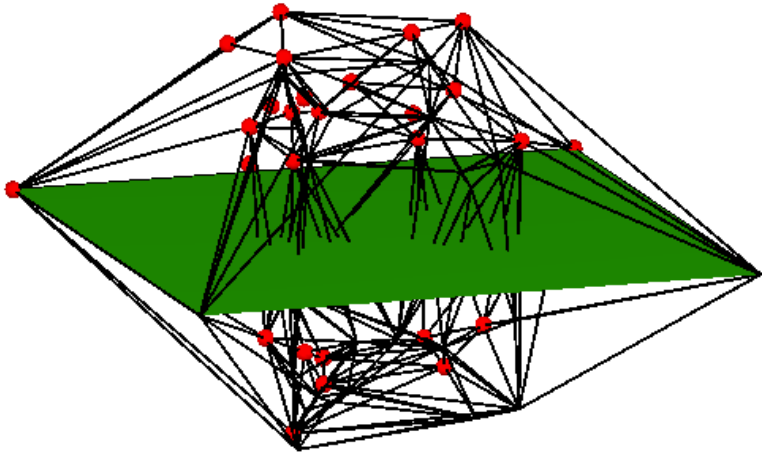
Constrained Mesh Generation

The Meshing Problem

- Given a set of constraints, edges and polygons, how to generate a triangulation that respects them?



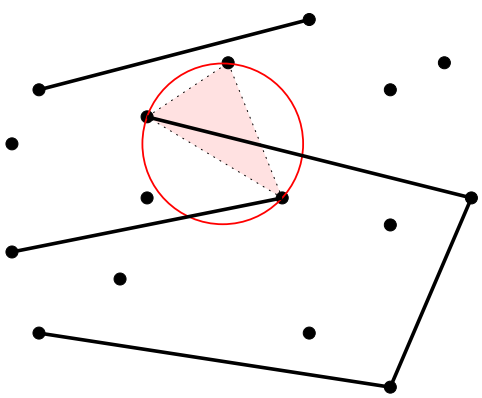
How to recover the edge AB?
Image from [**Owen 1999**]



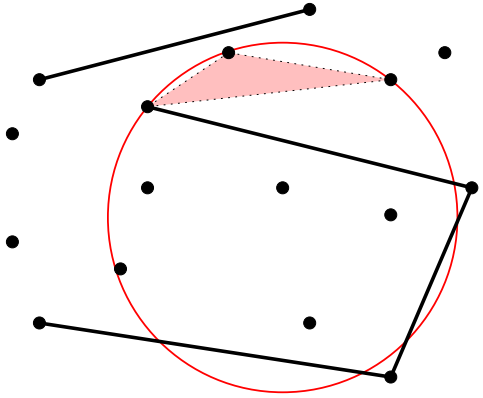
How to recover the rectangular face

Constrained Delaunay Triangulations

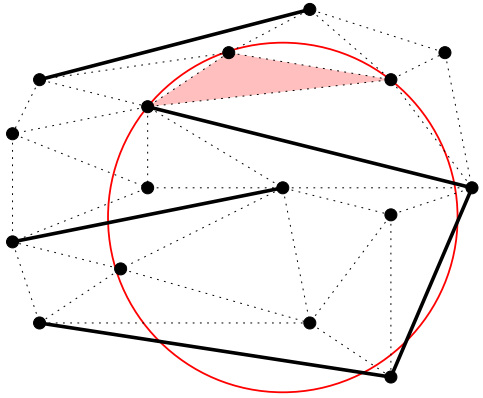
- A triangle is constrain Delaunay if (i) it does not intersect with an input segment, and (ii) its circumcircle encloses no vertex which is visible from its interior.
- A constrained Delaunay triangulation (CDT) is a triangulation whose triangles are all constrained Delaunay [**Lee & Lin 1986**].
- A CDT has many nice properties as close as to those of a Delaunay triangulation.
- Chew's algorithm constructs a CDT in $O(n \log n)$ time [**Chew 1989**].



a non-constrained triangle



a constrained Delaunay triangle



a constrained Delaunay triangulation

The edge flip algorithm

- Starting from an initial triangulation, recovering missing edges by flipping crossing edges. The algorithm halts in time $O(n^2)$ after fewer than $\binom{n}{2}$ flips.
- It takes at most $O(n^2)$ time to recover a set of edges with n vertices.

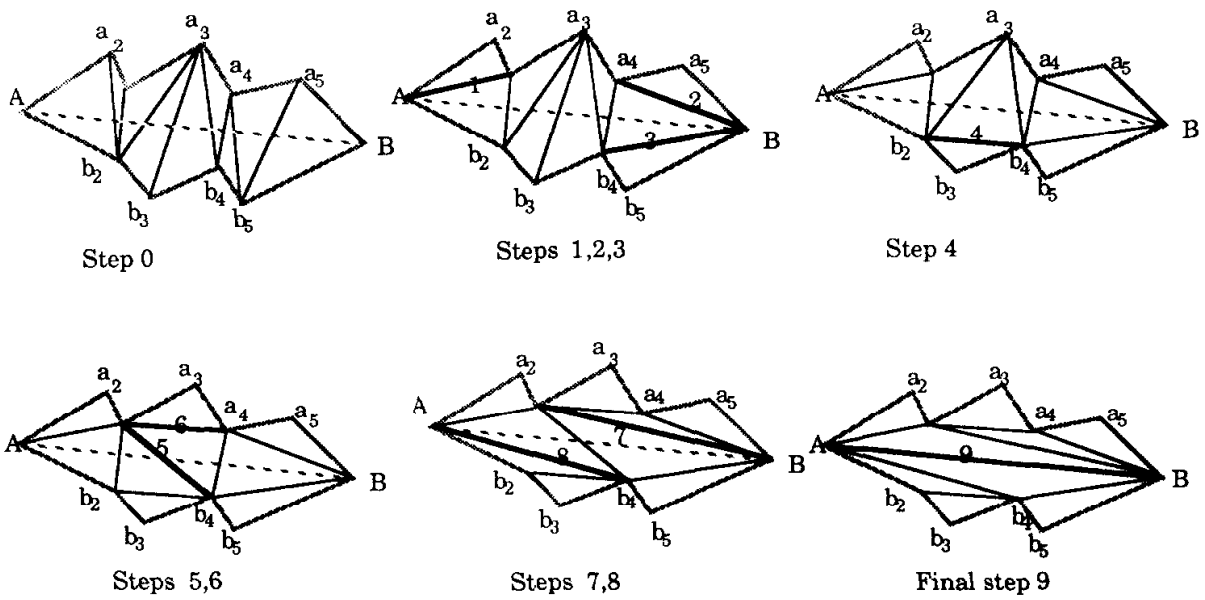
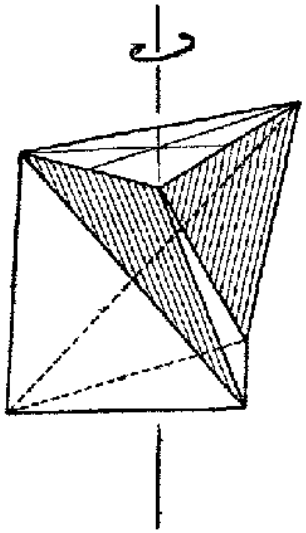


Fig. 11. Step 0, steps 1, 2, 3, step 4, steps 5, 6, steps 7, 8, and step 9.

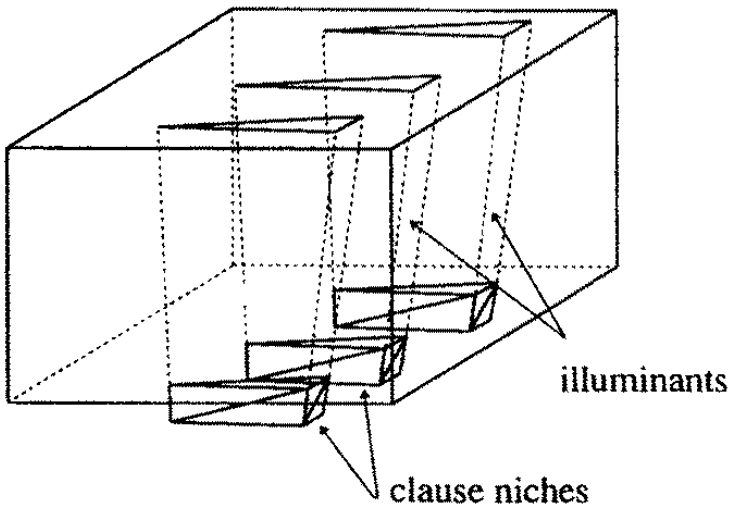
Recover an edge by a sequence of edge flips.
Images from [**George, Hecht, & Saltel 1991**].

3D Indecomposable Polyhedra

- A simple polyhedron P may not have a tetrahedralization without using additional points (Steiner points[(1)]) [**Lennes 1911, Schönhardt 1928**].
- The problem of deciding whether P can be tetrahedralized without Steiner points is NP-complete [**Rupper & Seidel 1992**].



The Schönhardt polyhedron
(a twisted prism)
Image from [**Schönhardt 1928**]

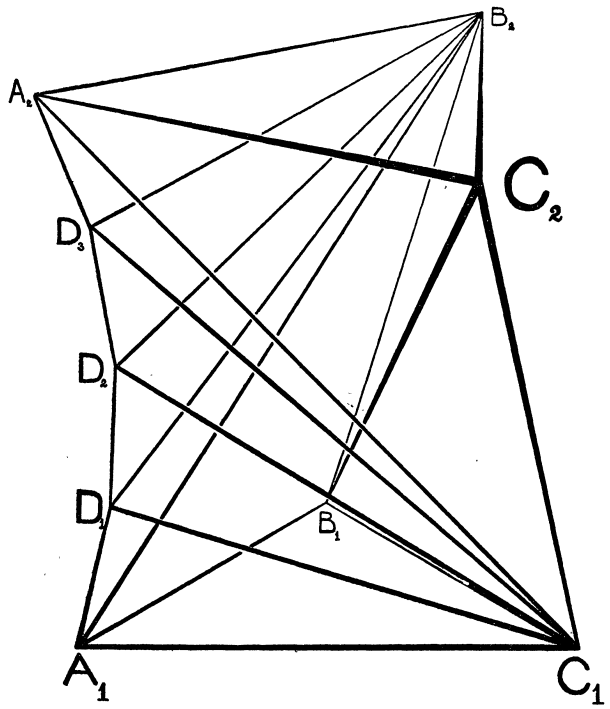


A prove of NP-completeness
The constructed polyhedron
Image from [**Rupper & Seidel 1992**]

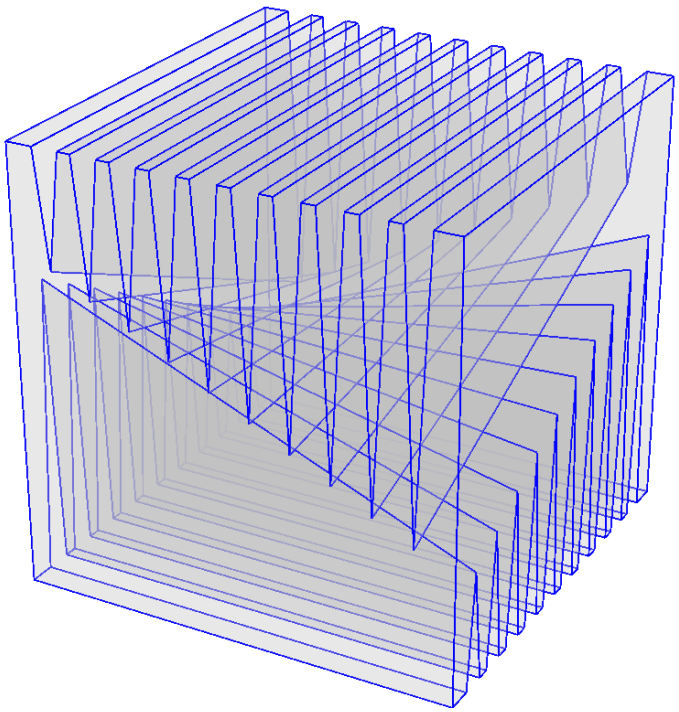
(1) Jakob Steiner (1796 – 1863), a Switzerland native and a geometer from Berlin.

The Number of Steiner points

- A constrained tetrahedral meshing algorithm should use a small number of Steiner points when it is possible.



A Bagemihl's Polyhedron Π_9
(Π_6 = Schönhardt's polyhedron)
[Bagemihl 1948]

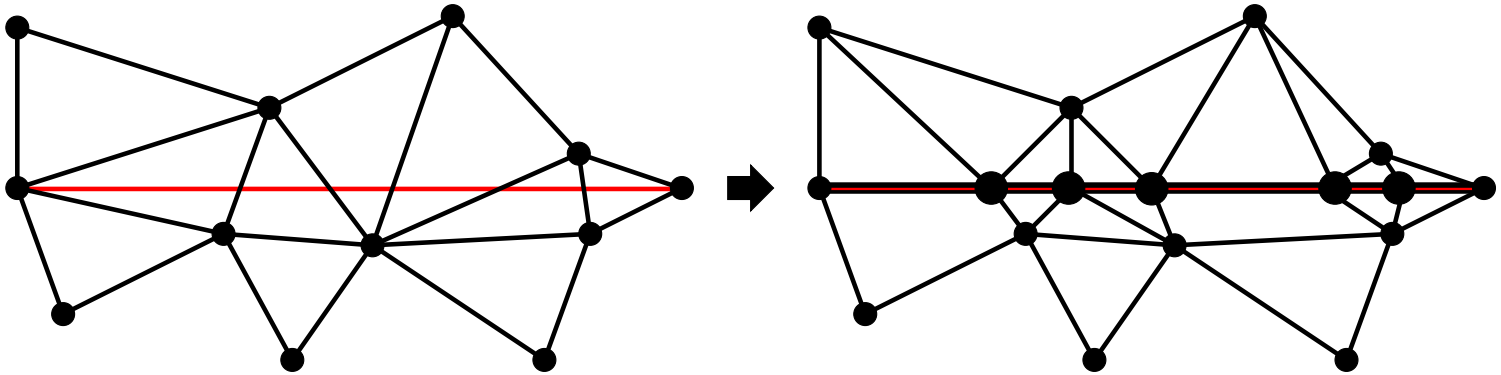


A Chazelle's polyhedron

[Chazelle 1984]

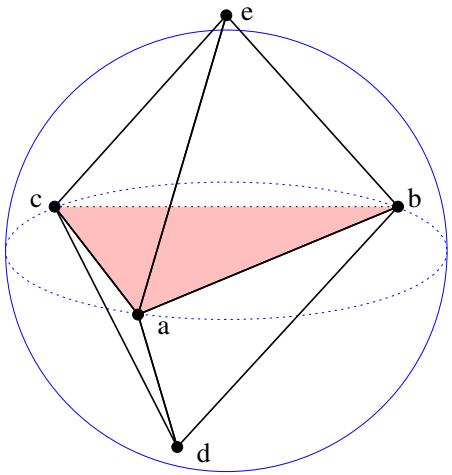
Conforming Delaunay Tetrahedralizations

- Algorithms: [**Murphy, Mount, & Gable 2000**], [**Cohen-Steiner, Colin de Verdière, & Yvinec 2002**]
- A conforming Delaunay tetrahedralization may include a lot of Steiner points. An $O(n^3)$ upper bound of Steiner points for 2d conforming Delaunay triangulation is proven [**Edelsbrunner & Tan 1993**]. (A recent improvement of this result by Bishop is $O(n^{2.5})$). The 3d case is still open.

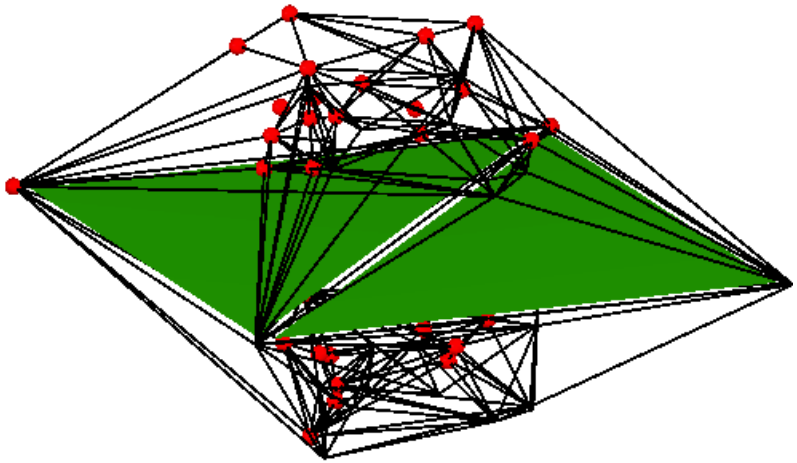


Constrained Delaunay Tetrahedralizations

- CDTs have similar properties as those of DTs [**Shewchuk 2008**].
- A triangular face is locally Delaunay if two tetrahedra sharing it have no vertex inside each others' circumspheres.
- The **Locally Delaunay Lemma** (for CDT): If every triangle in $\mathcal{T}$ not in a polygon of $\mathcal{X}$ is locally Delaunay, then $\mathcal{T}$ is a CDT of $\mathcal{X}$ [**Shewchuk 1998**].



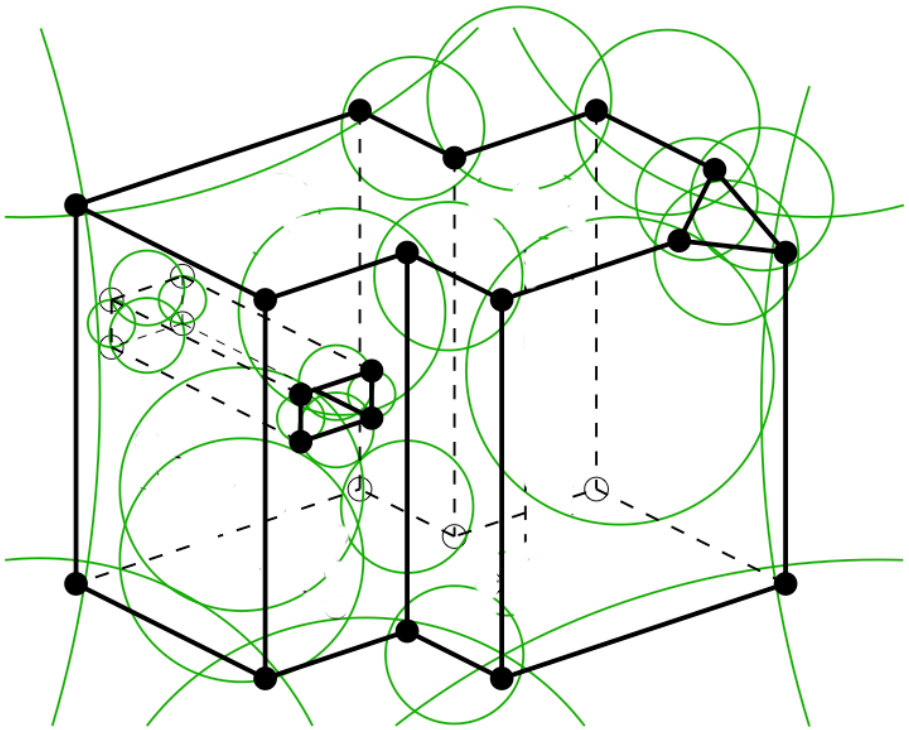
a locally Delaunay face (shaded)



a constrained Delaunay tetrahedralization (CDT)

The Existence of CDTs

- An edge e in a PLC $\mathcal{X}$ is *strongly Delaunay* if there exists a circumball of e such that no other vertex of $\mathcal{X}$ lies inside or on the boundary of the ball.
- **Theorem** [**Shewchuk 1998**]. If every edge of the PLC is strongly Delaunay, then it has a CDT.
- A *Steiner CDT* of $\mathcal{X}$ is a CDT of $\mathcal{X} \cup \mathcal{S}$, where $\mathcal{S}$ is a set of *Steiner points*.



Courtesy of J. Shewchuk

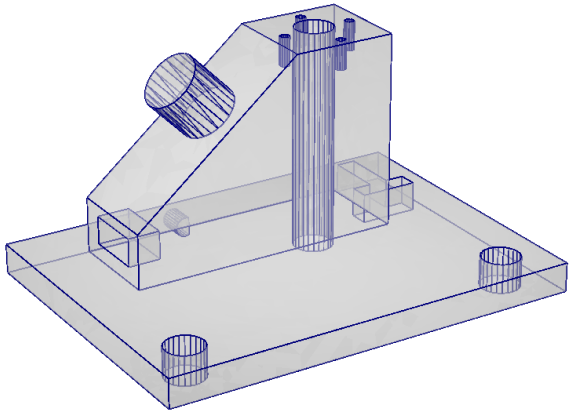
CDT Algorithms

Given a 3D PLC $\mathcal{X}$, a Steiner CDT of $\mathcal{X}$ is generated in three steps:

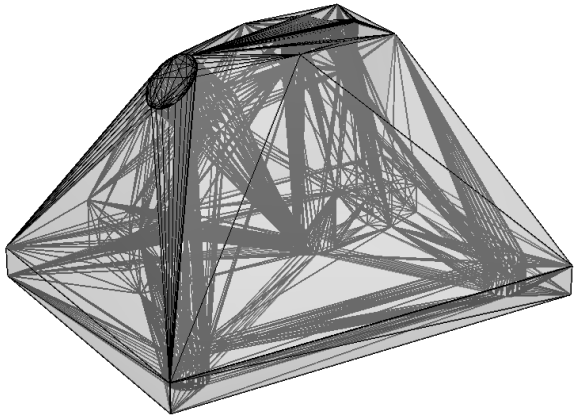
(1) **Initialization**: Creating a Delaunay tetrahedralization of the vertices of $\mathcal{X}$;

(2) **Segment insertion**: Splitting all non-Delaunay segments of $\mathcal{X}$ by inserting Steiner points, until all subsegments are Delaunay;

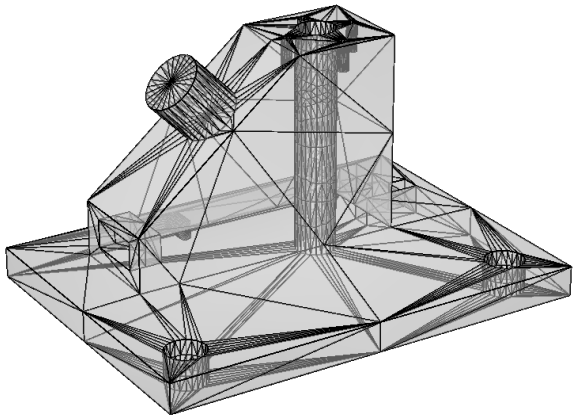
(3) **Polygon insertion**: Generating the Steiner CDT of $\mathcal{X}$.



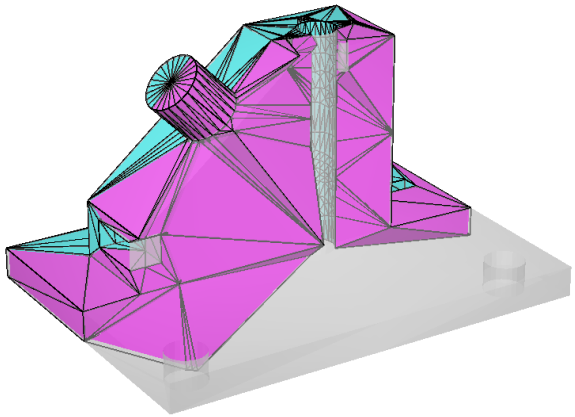
An input PLC $\mathcal{X}$



(1) Initialization



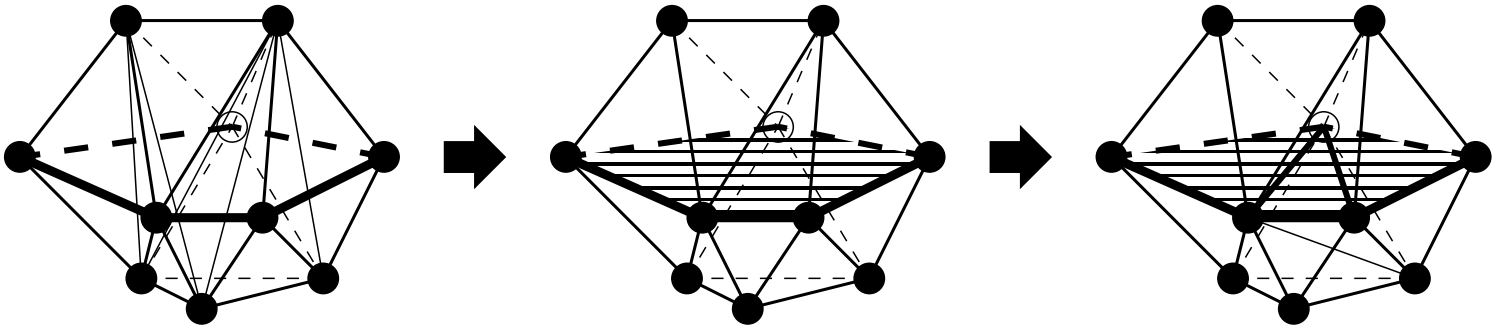
(2) Segment insertion



(3) Polygon insertion

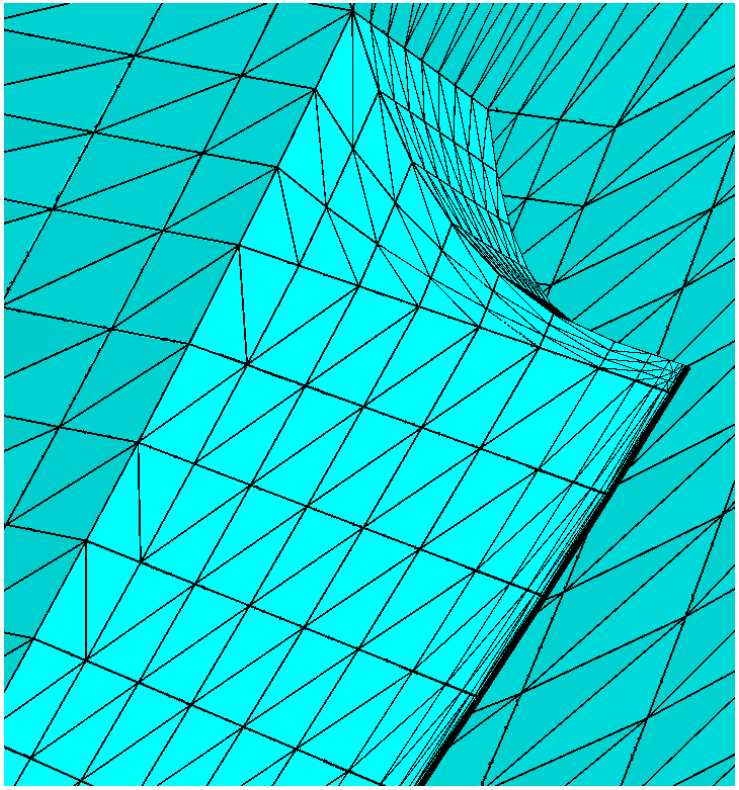
CDT Algorithms (Cont'd)

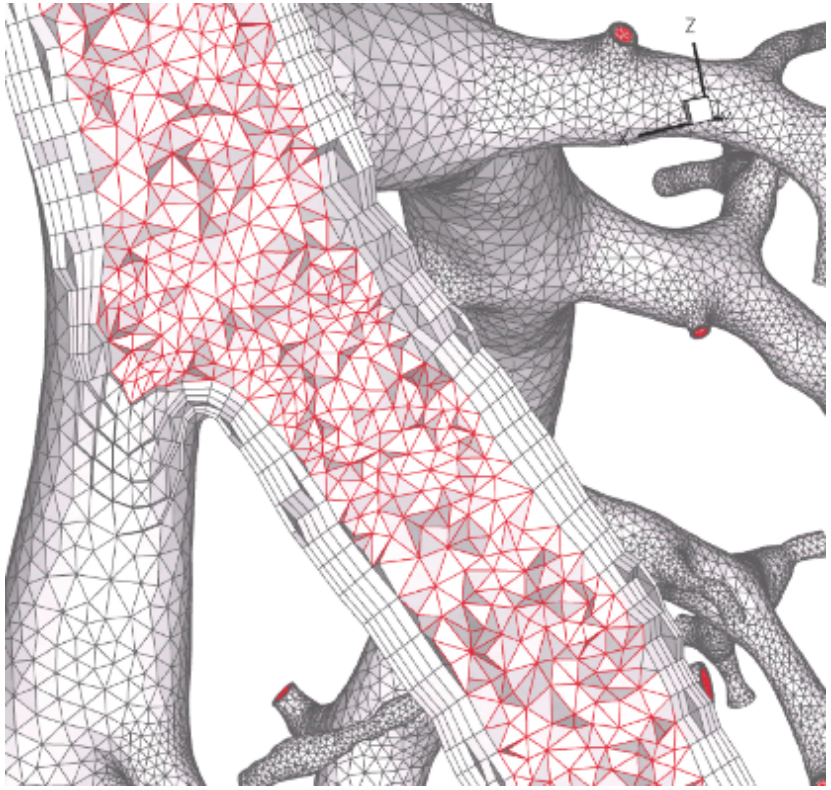
- Segment splitting is easy - incremental DT construction.
  - Choices of Steiner points: [**Shewchuk 2002**] and [**Si & Gärtner 2005**].
- Polygon insertion is the key step in creating CDTs.
  - A flip-based algorithm [**Shewchuk 2003**].
  - A cavity re-tetrahedralization algorithm [**Si & Gärtner 2005, 2011**].
  - Handling non-coplanar polygons [**Si & Shewchuk 2012**].



Insert a polygon into a CDT.

Preserving Constraints





Boundary Recovery Methods

(1) Use edge/face swaps together with interior Steiner points insertion [**George, Hecht, & Saltel 1991**] (in `TetMesh-GHS3D`).

(2) Insert Steiner points at where the boundaries and $\mathcal{T}$ intersect, delete vertices or relocate them from the boundaries afterwards [**Weatherill & Hassan 1994**].

(3) Combine methods *(1)* and *(2)* [**George, Borouchaki, & Saltel 2003**] (in `TetMesh-GHS3D`).

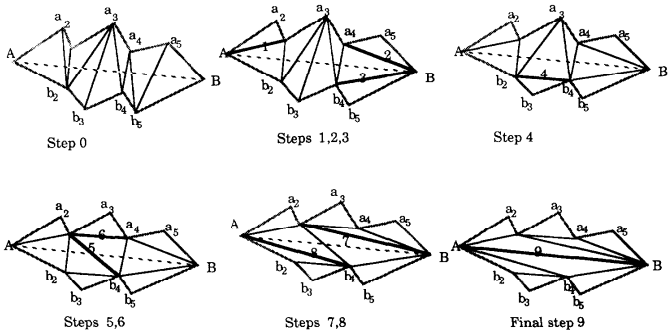


Fig. 11. Step 0, steps 1, 2, 3, step 4, steps 5, 6, steps 7, 8, and step 9.

(1) [**George, Hecht, and Saltel 1991**]

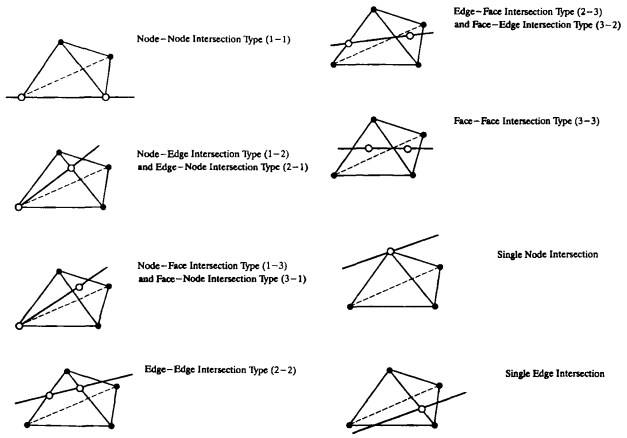
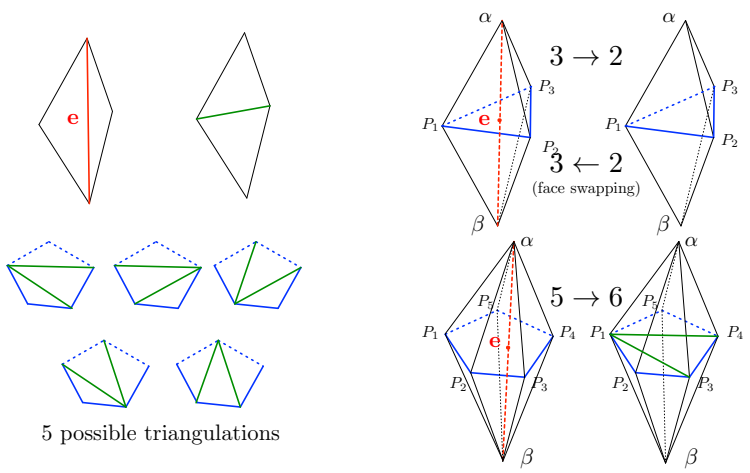


Figure 5. Types of edge–tetrahedron intersections for missing surface edges

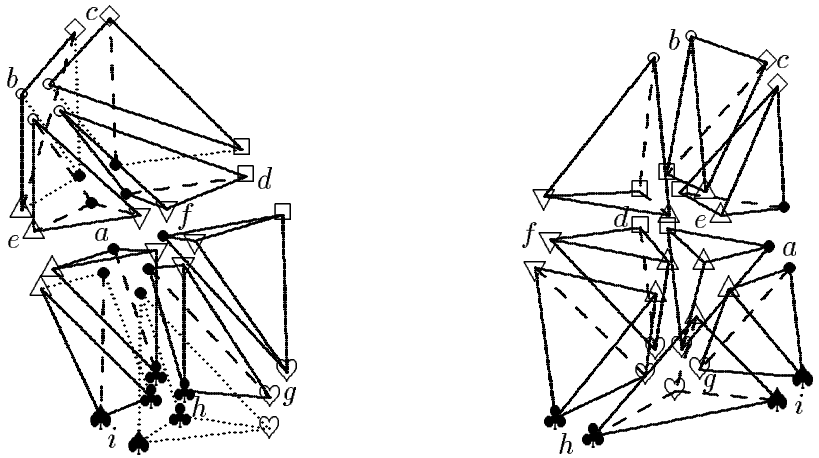
(2) [**Weatherill and Hassan 1994**]

Combination of Flips

- The *n*-to-*m* operation has been widely used in mesh generation and mesh optimization, e.g., [**Marcum & Weatherill 1995**; **Joe 1995**; **Freitag & Ollivier-Gooch 1997**; **George & Borouchaki 2003**; **Klinger & Shewchuk 2007**; **Alauzet 2012**], etc.
- Some other types of combinations of flips were reported. In particular, several operations described in [**Joe1995**] can be viewed as a combination of two *n*-to-*m* flips with the limitation that *n* is either 3 or 4.



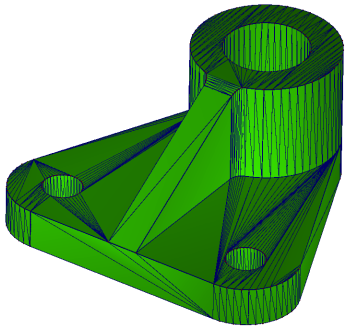
Images from [**Alauzet 2012**]



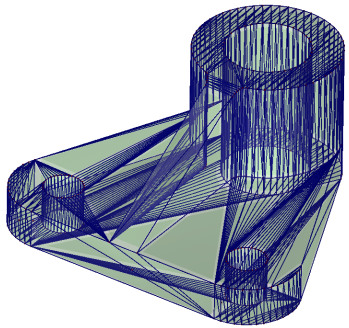
Images from [**Joe 1995**]

A Revised Boundary Recovery Method [**Si 2013**]

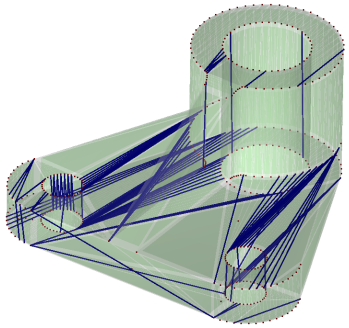
- This method uses the same procedures as those in [**George et al. 1991**; **Weatherill and Hassan 1994**; **George et al. 2003**].
- A new edge removal algorithm based on flips [**Si 2013**] is employed in recovering edges and faces.
- This algorithm is implemented in `TetGen`.



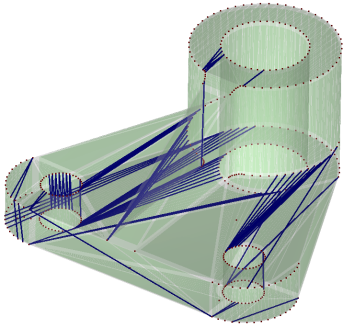
(a) `tdpg192.mesh`



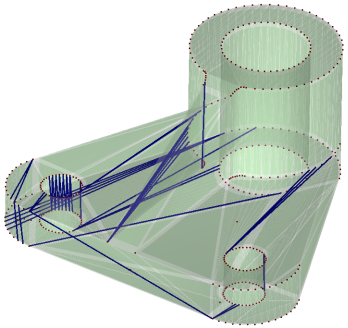
(b) Begining (1308 edges)



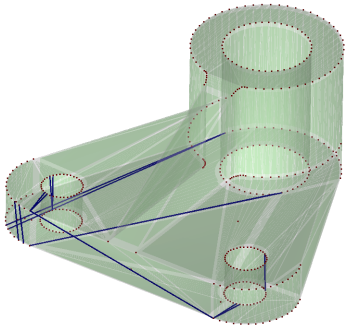
(c) level = 2 (114 edges)



(c) level = 4 (88 edges)



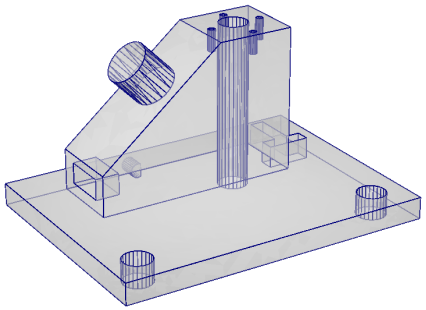
(b) level = 7 (46 edges)



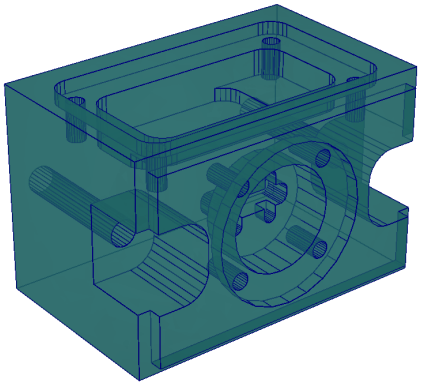
(c) level = ∞ (14 edges)

Compared with TetMesh-GHS3D

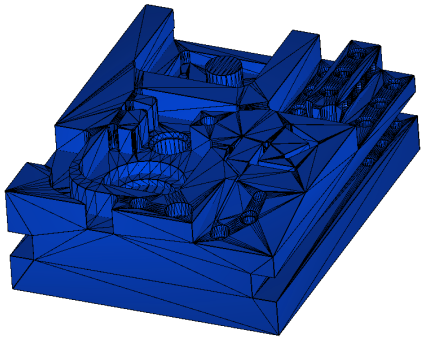
- Input meshes are available from INRIA's mesh repository.
- TetMesh-GHS3D, version 4.2, 2010.



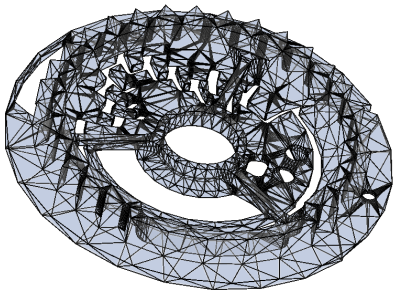
cami1a



thepart



mohne

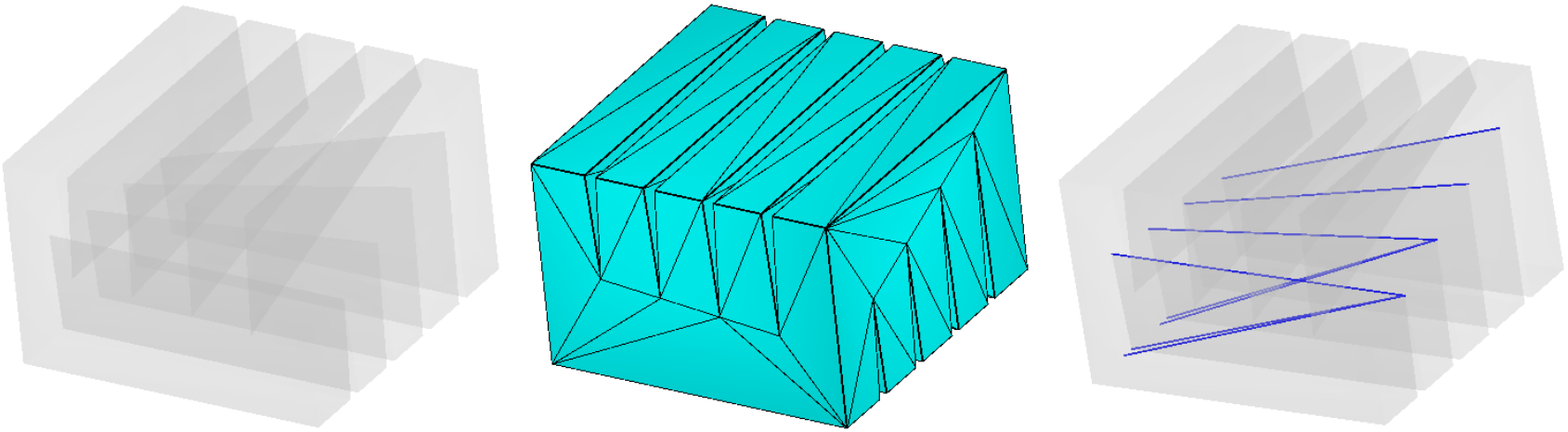


thru-maze

	cami1a	thepart	mohne	thru-maze
(1) [**George, Hecht & Saltel 1991**]	4	15	34	18
(2) [**George, Borouchaki & Saltel 2003**]	12	Failed	94	16
TetGen's approach	2	0	10	8

Reports of the number of (interior) Steiner points.

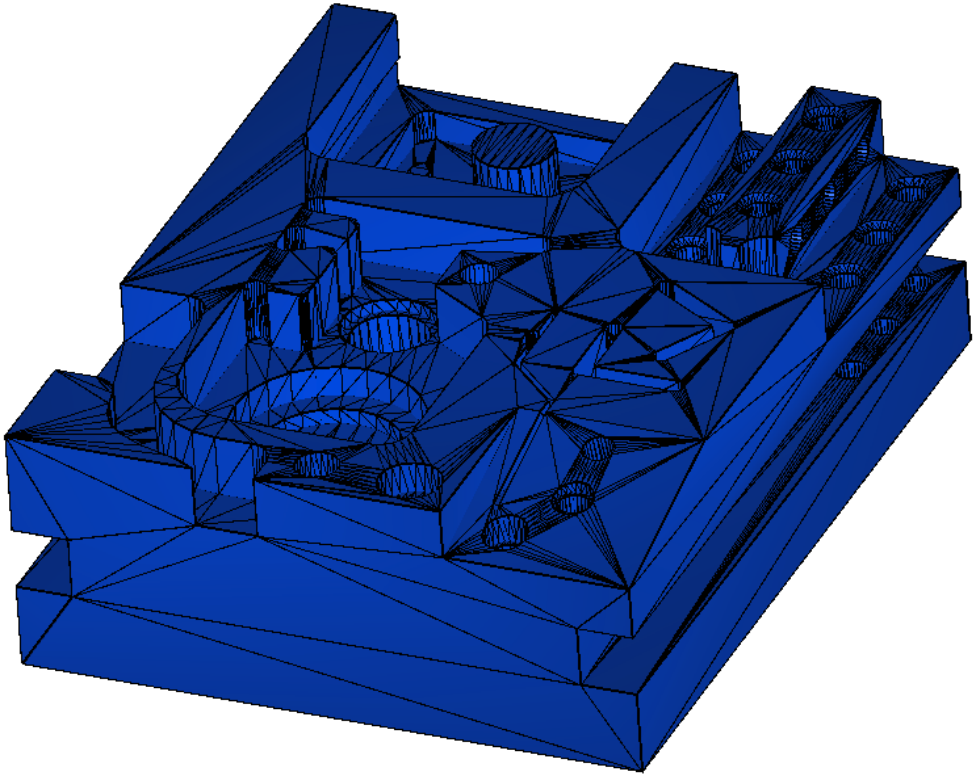
Compared with TetMesh-GHS3D (Cont'd)



Example, CHAZEL.mesh.
Left and Middle: the input PLC, 56 points, 108 triangles.
Right: the 8 missing segments after flips.

	CHAZEL
(1) [**George, Hecht & Saltel 1991**]	Failed
(2) [**George, Borouchaki & Saltel 2003**]	16
TetGen's approach	14

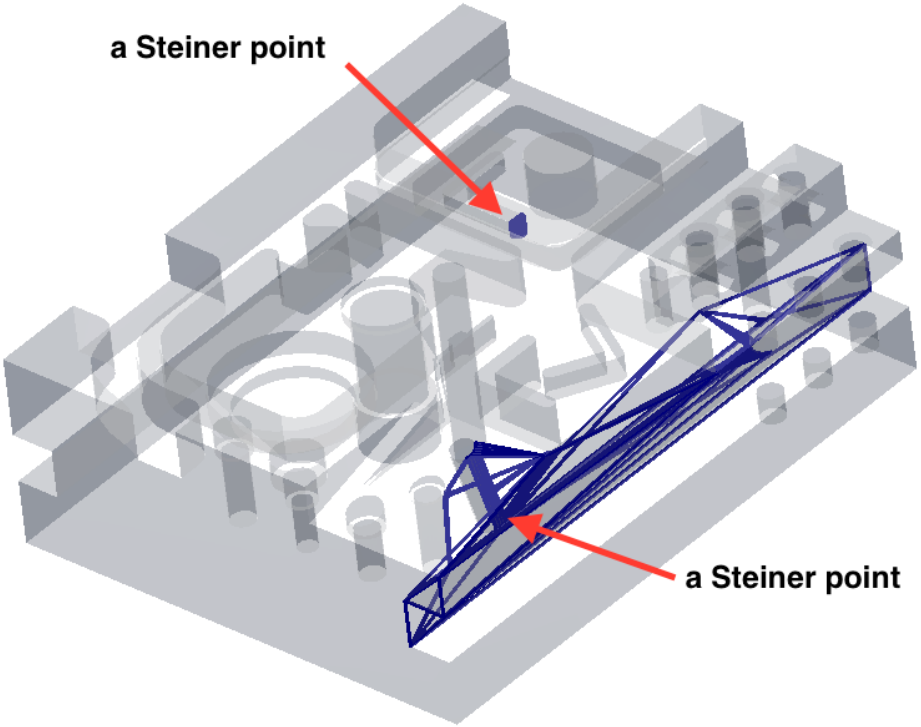
Experiment 1 (TetGen v1.5)



Example: `mohne` (from INRIA Mesh Repository)

Input: 2760 points, 5560 triangles
Output: added 2 Steiner points

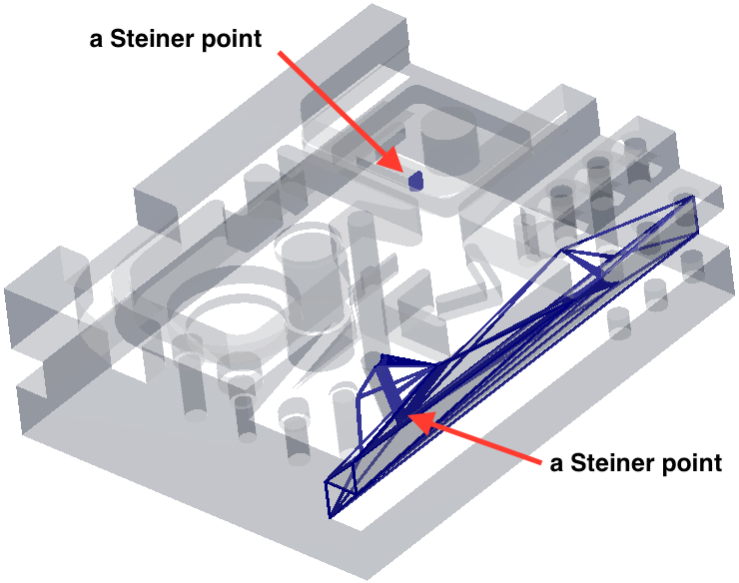
Experiment 1 (TetGen v1.5)

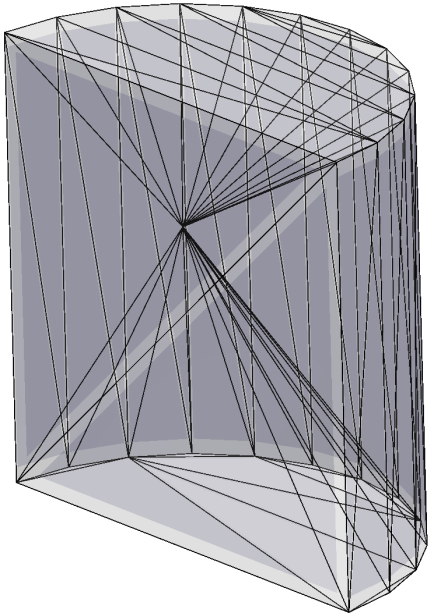


Example: `mohne` (from INRIA Mesh Repository)

Input: 2760 points, 5560 triangles
Output: added 2 Steiner points

Experiment 1 (TetGen v1.5)

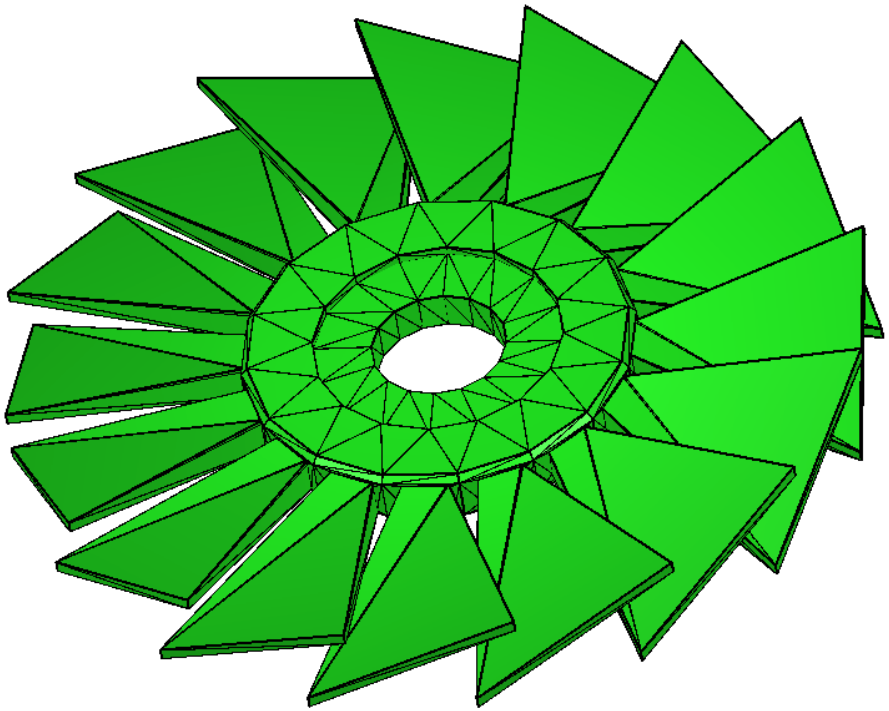




Example: `mohne` (from INRIA Mesh Repository)

Input: 2760 points, 5560 triangles
Output: added 2 Steiner points

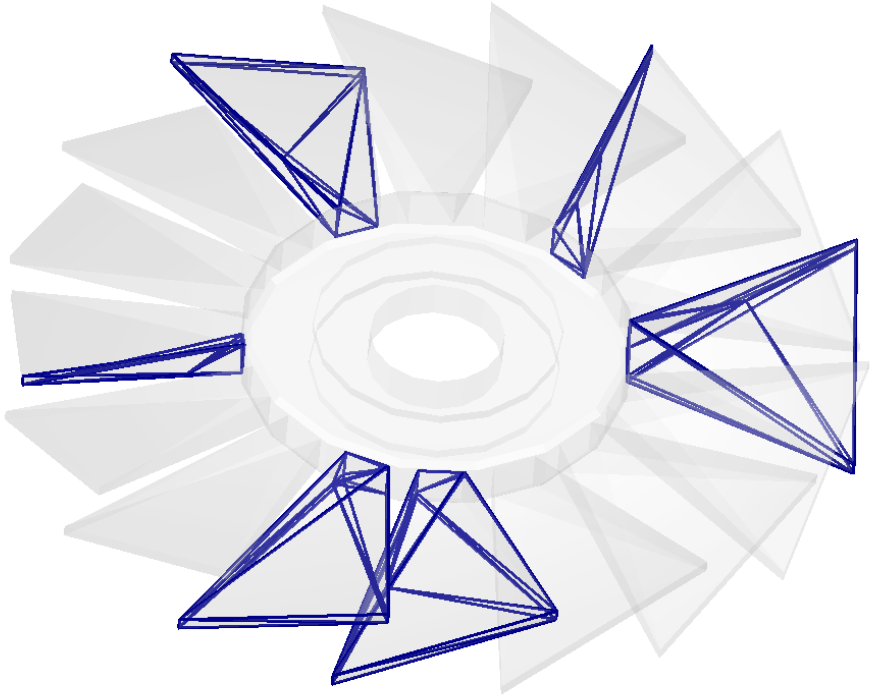
Experiment 2 (TetGen v1.5)



Example: 03-machinery-part_cut (from INRIA Mesh Repository)

Input: 448 points, 1120 triangles
Output: added 8 Steiner points

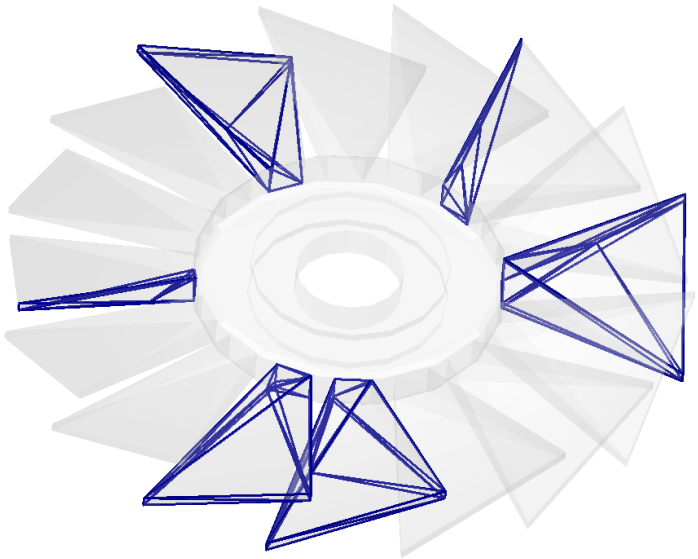
Experiment 2 (TetGen v1.5)

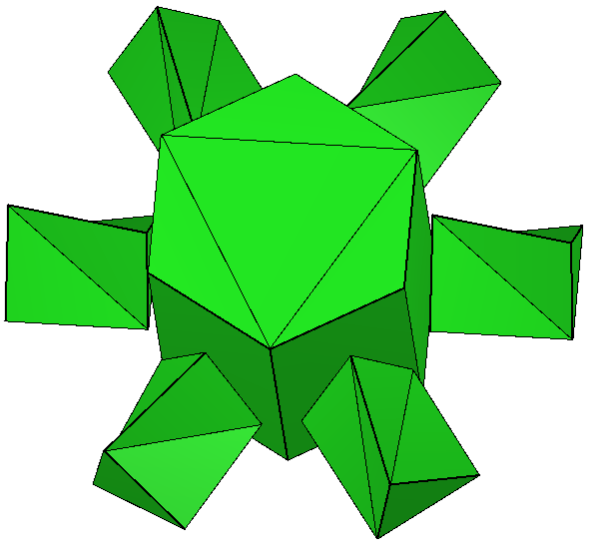


Example: 03-machinery-part_cut (from INRIA Mesh Repository)

Input: 448 points, 1120 triangles
Output: added 8 Steiner points

Experiment 2 (TetGen v1.5)



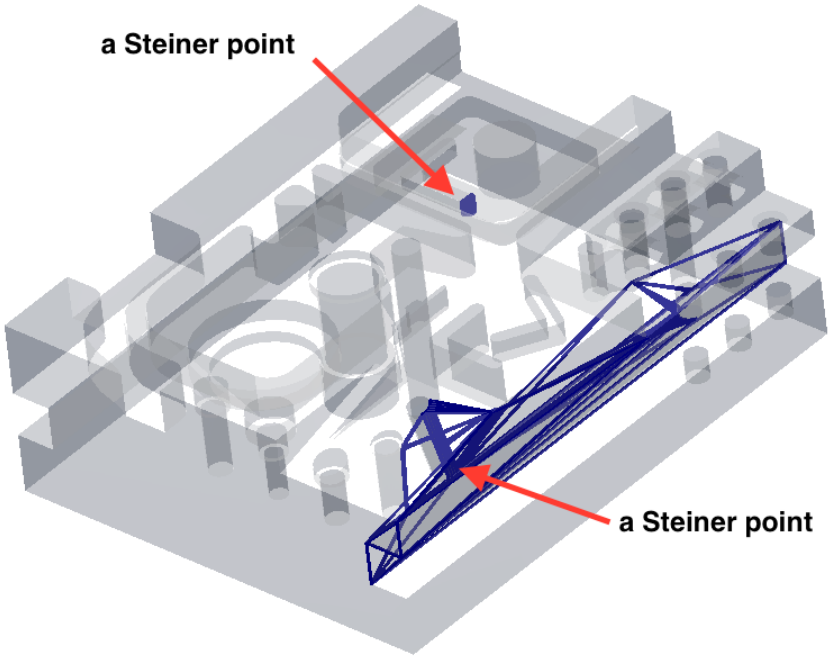


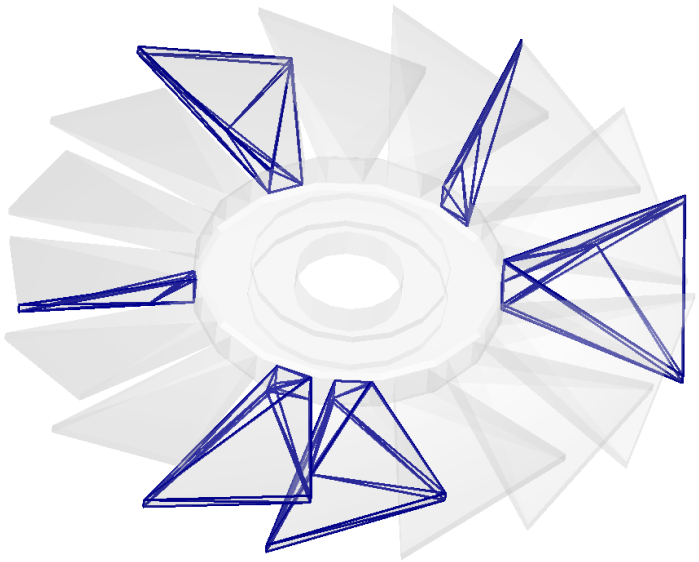
Example: 03-machinery-part_cut (from INRIA Mesh Repository)

Input: 448 points, 1120 triangles
Output: added 8 Steiner points

Irreducible Polyhedra

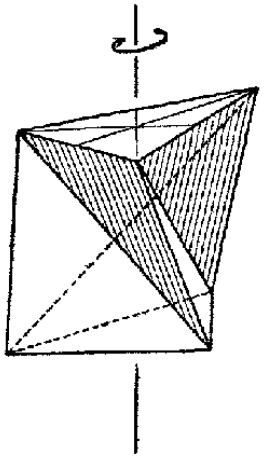
A polyhedron is irreducible if it cannot be cut into smaller parts without using additional vertices.



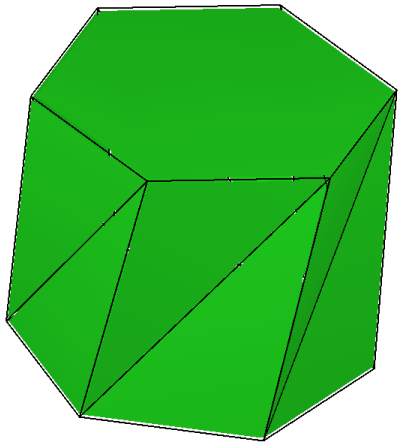


Examples of Irreducible Polyhedra

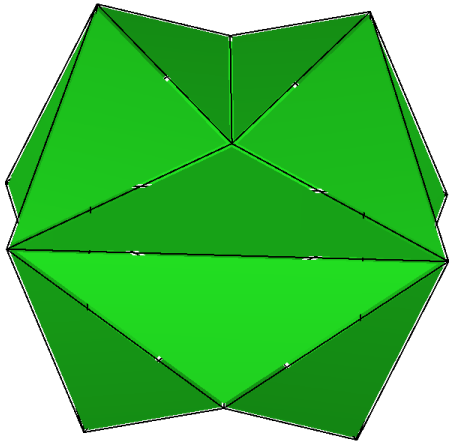
Schönhardt polyhedron
1928



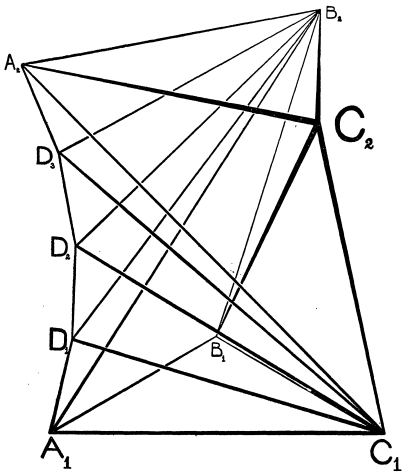
Rambau's generalisation
2005



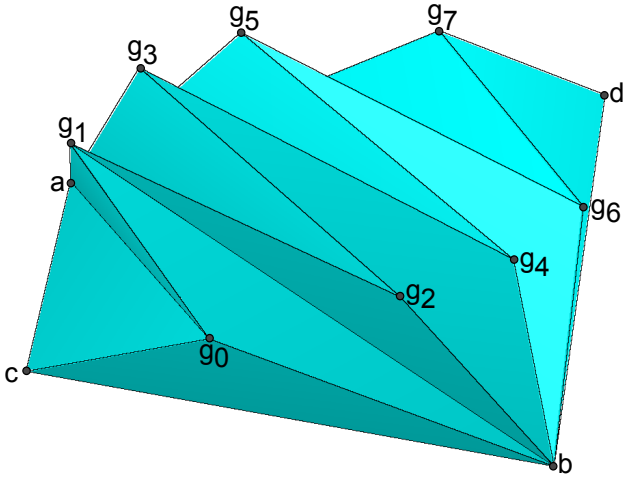
Jesson's polyhedron
1967



Bagemihl polyhedron
1948



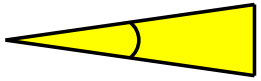
Goerigk and Si's generalisation
2015



Mesh Adaptation

Mesh Quality

Skinny elements cause problems.

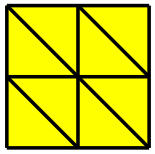


Small angles cause poor conditioning.

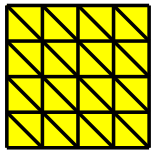
Large angles cause discretization error
& big errors in interpolated derivatives.

For tetrahedra, this applies to the dihedral angles.
(Not the plane angles!)

The number of elements matters.



Fewer elements ➡ faster solution.

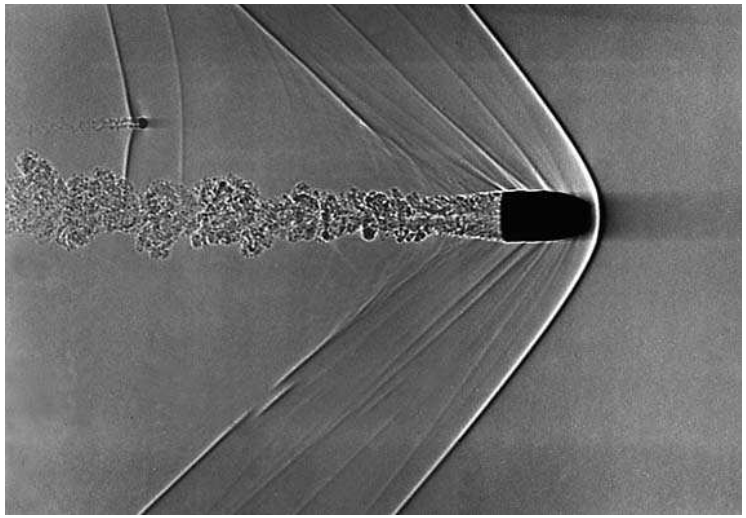


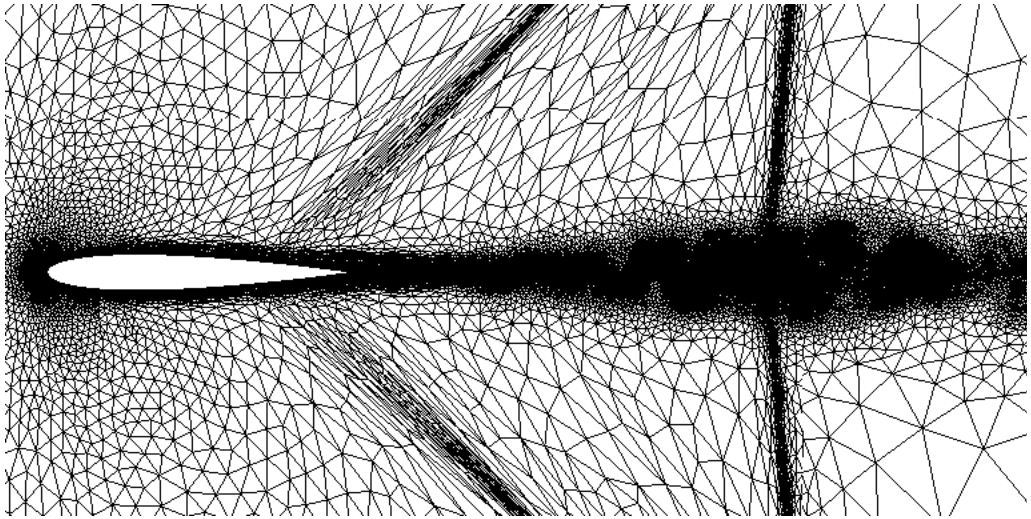
More elements ➡ more accurate solution.

A slide from Shewchuk's IMR course (2005)

Mesh Quality

- The mesh quality strongly relates to the physics and the used numerical method (stability, convergence and order of convergence, error estimate) and to practical issues (accuracy, reliability, and efficiency). These criteria are often antagonists and incompatible with another.
- Studying mesh quality is an active research topic, see [**Knupp 2000**, **Shewchuk 2002**, **Huang 2005**].

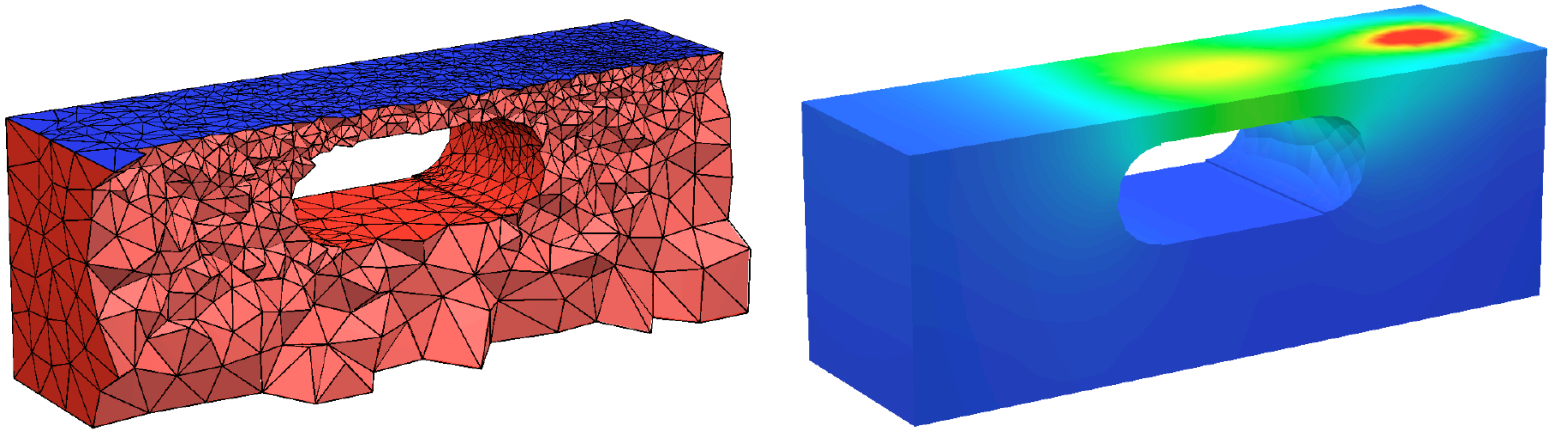




Images from Frey's IMR course (2005)

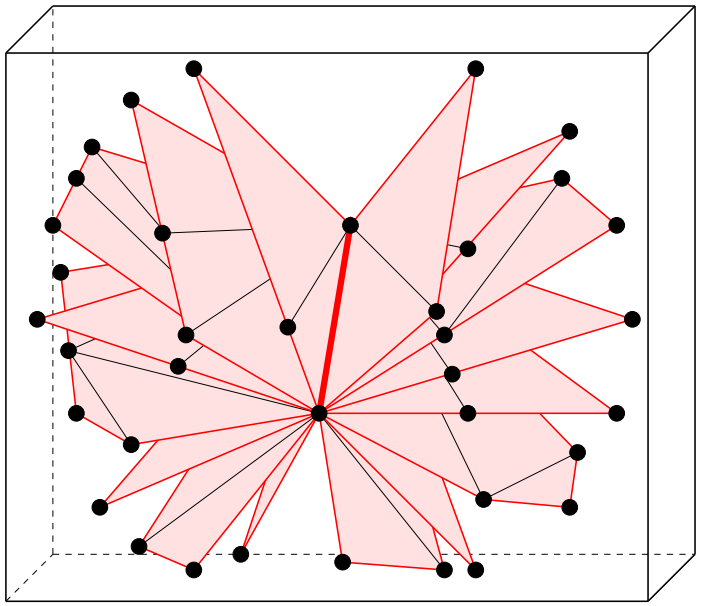
Basic Requirement of Quality Meshing Algorithms

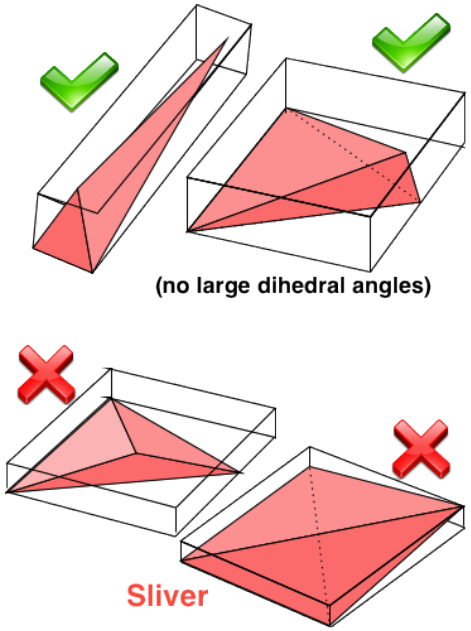
- A good mesh should then try to obtain a good compromise between different criteria, preserving the accuracy of the solution at a reasonable computational effort.
- The basic requirement of quality meshing algorithms are:
  - (Boundary preservation) ability to preserve domain boundaries.
  - (Mesh quality) ability to generate nicely-shaped tetrahedra.
  - (Mesh size) ability to generate a small number of tetrahedra.
  - (Mesh adaptivity) ability to generate more elements in local regions (where higher accuracy is needed).



Small Angles

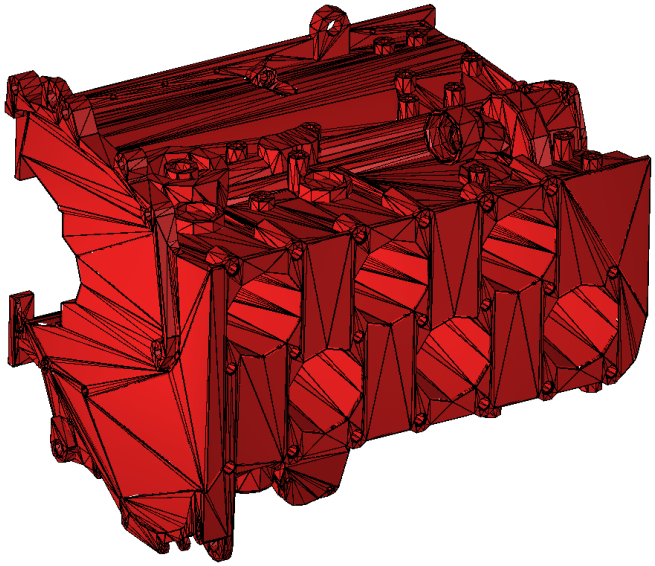
- Small angles are acute angles or dihedral angles formed by constraints. They need to be preserved or correctly represented.
- Small angles can not be meshed by nicely-shaped tetrahedra.
- The challenge is how to use tetrahedra with no large dihedral angles (no slivers) in meshing small angles.



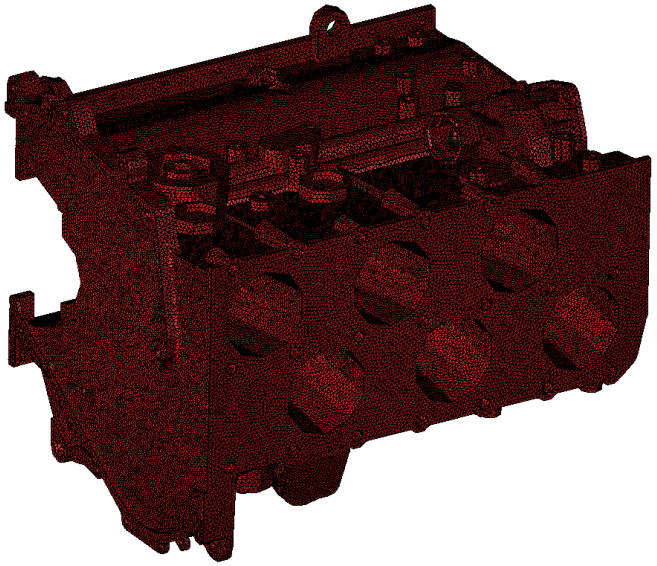


Surface Remeshing

- When the input surface mesh have poorly-shaped elements, it is necessary to re-mesh it by nicely-shaped elements.
- When domain boundary are formed curved surfaces, it is necessary to re-mesh the input surface mesh according to the geometry approximation and mesh quality of the curved surfaces.
- The challenge is how to efficiently combine quality surface re-meshing methods with the quality tetrahedral meshing algorithms.



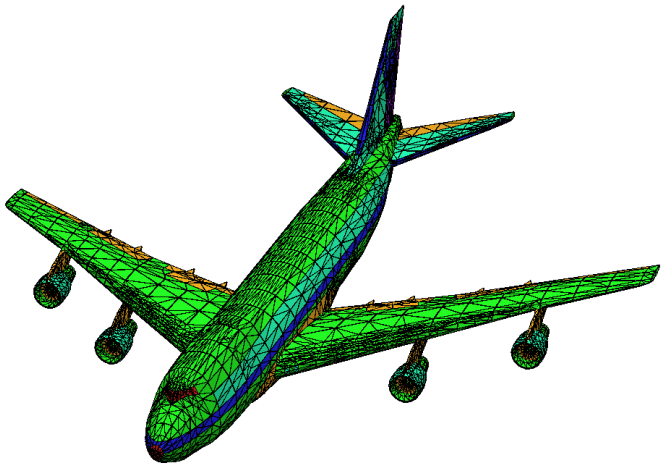
An input surface mesh

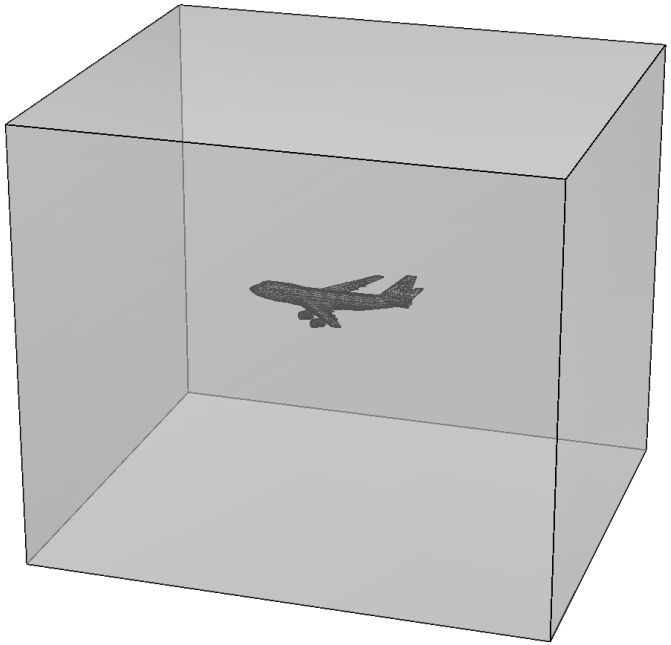


A remesh of the input

Steiner Points Generation

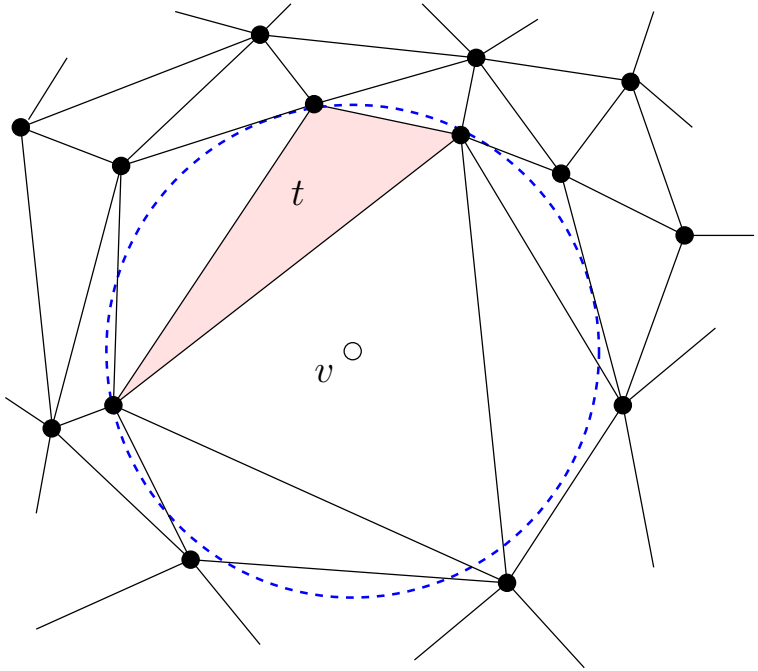
- Advancing-front: [Lo 1991, **Löhner 1996**, **Marcum & Weatherill 1995**];
- Sphere packing: [**Shimada & Gossard 1995**, **Miller et al 1996**];
- Octree-based: [**Mitchell & Vavasis 2000**];
- Longest edge subdivision: [**Rivara 1997**];
- Delaunay Refinement: [**Chew 1989**, **Ruppert 1995**, **Shewchuk 1998**];

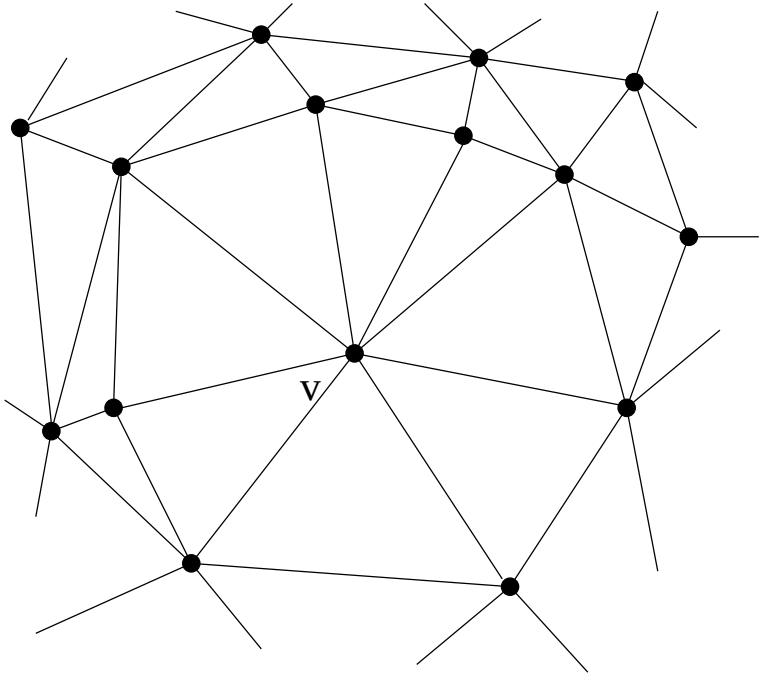




Delaunay Refinement [**Chew 1989, Ruppert 1995**]

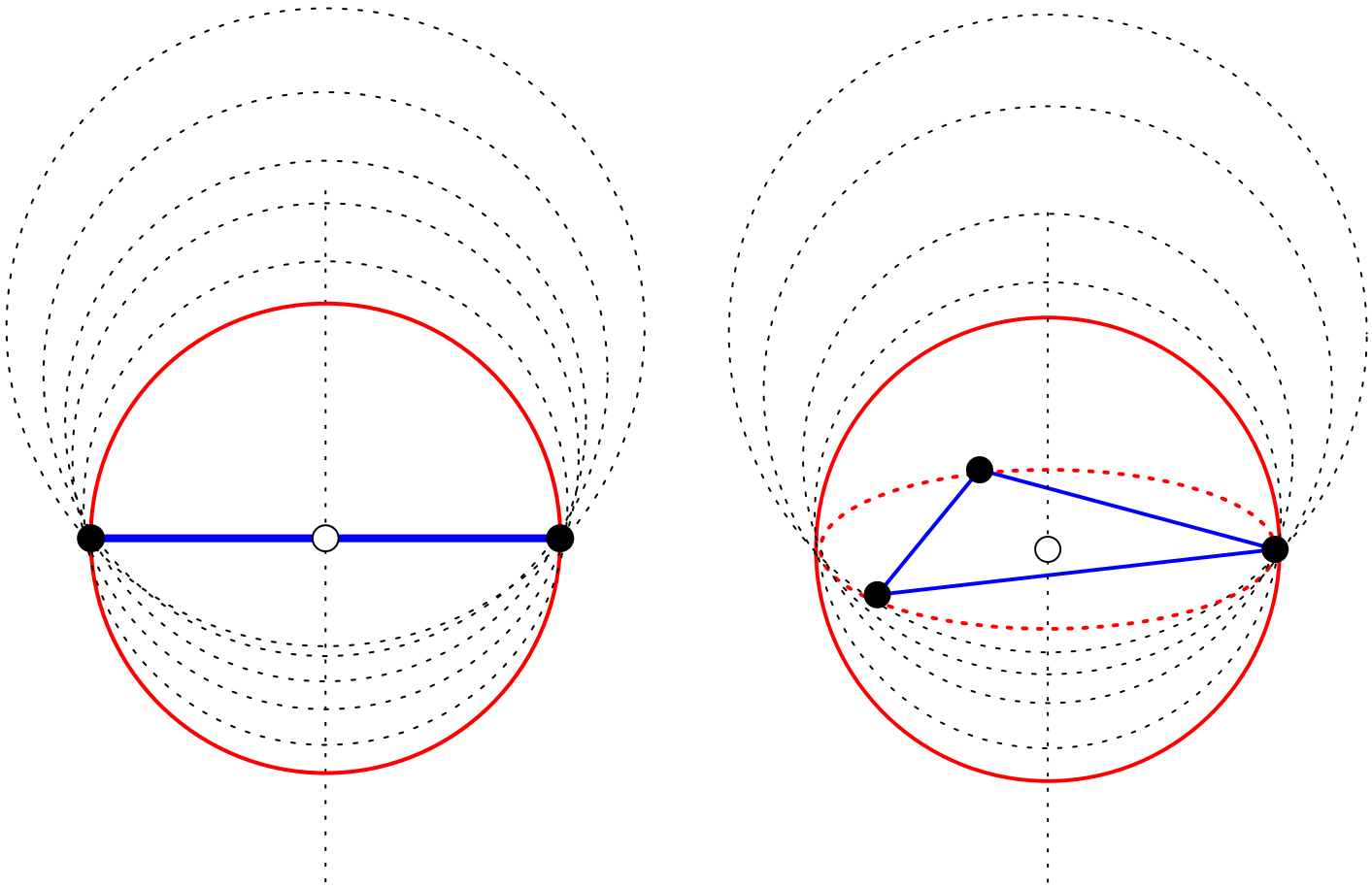
- Kill bad elements by insertion of their circumcenter.
- Bad elements: badly-shaped, oversized, etc.





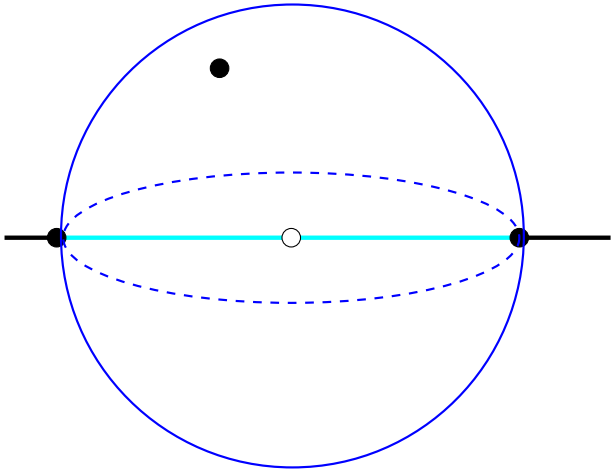
Diametrical Spheres

- A segment or a subface in $\mathbb{R}^3$ has infinitely many circumscribed spheres. The smallest one is called its diametral sphere.
- A segment or a subface is said to be encroached if there is a vertex inside its diametral sphere.

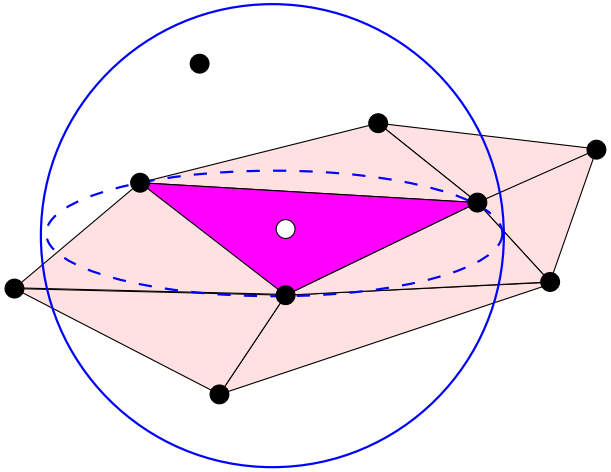


Point Insertion Rules

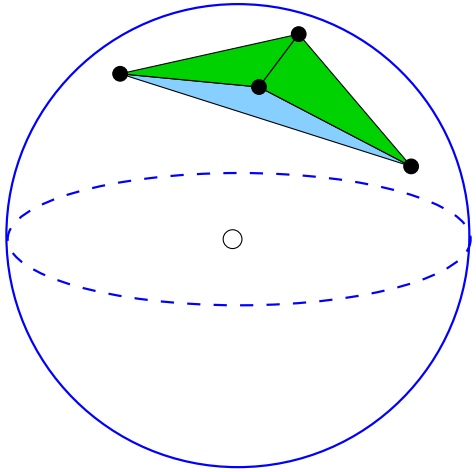
- **Rule 1**: Split a segment if it is encroached.
- **Rule 2**: Split a subface if it is encroached. However, if the new vertex would encroaches upon a segment, reject the vertex. Split the encroached segment(s) instead.
- **Rule 3**: Split a badly-shaped tetrahedron. However, if the new vertex would encroached upon a subface or a segment, reject the vertex. Split the encroached subface(s) or segment(s) instead.



Rule 1



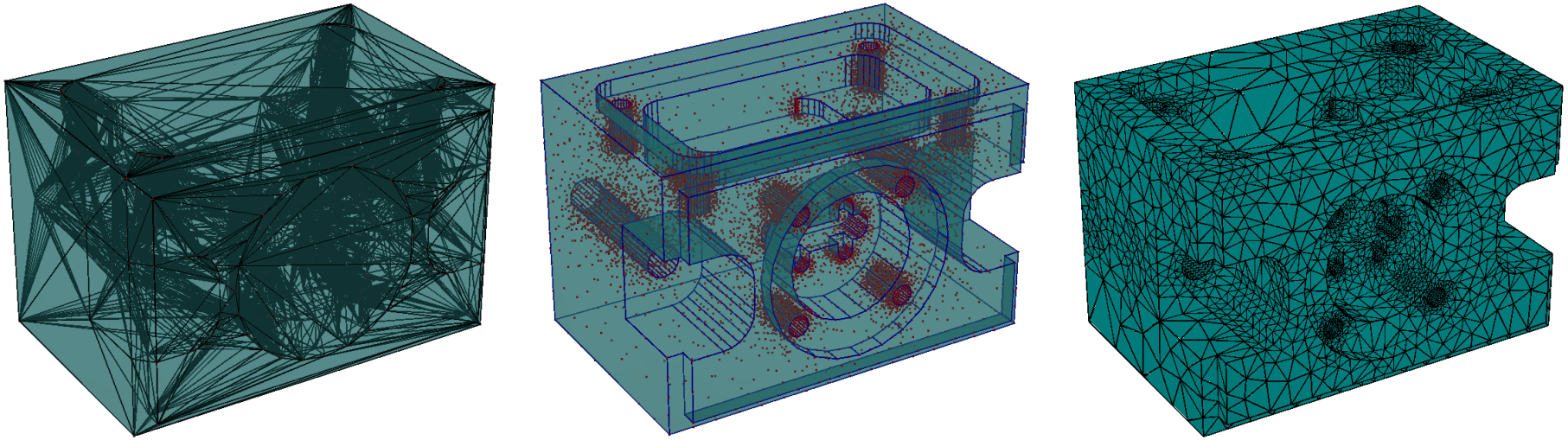
Rule 2



Rule 3

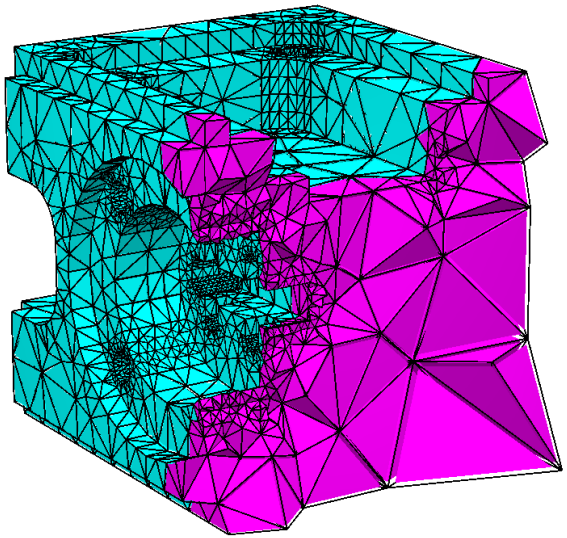
Rupper and Shewhcuk's Algorithm [**Ruppert 1995, Shewchuk 1998**]

DelaunayRefinement ($\mathcal{X}$, ρ_0)
// $\mathcal{X}$ is a PLC; ρ_0 is a radius-edge ratio bound.
Initialize a set V of the vertices of $\mathcal{X}$;
Initialize a Delaunay tetrahedralization $\mathcal{D}$ of V;
repeat:
    Create a new point by rule i, $i \in \{1, 2, 3\}$;
    Add $\mathbf{v}$ to V, update $\mathcal{D}$ of V;
until {no new point can be generated};
return $\mathcal{D}$ of V;

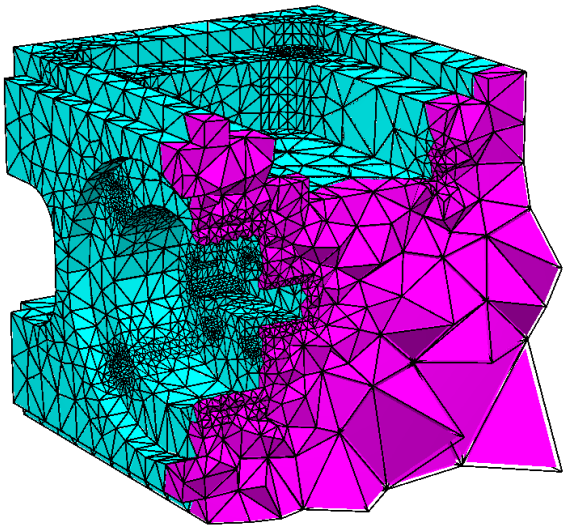


Guarantees on Mesh Quality and Mesh Size

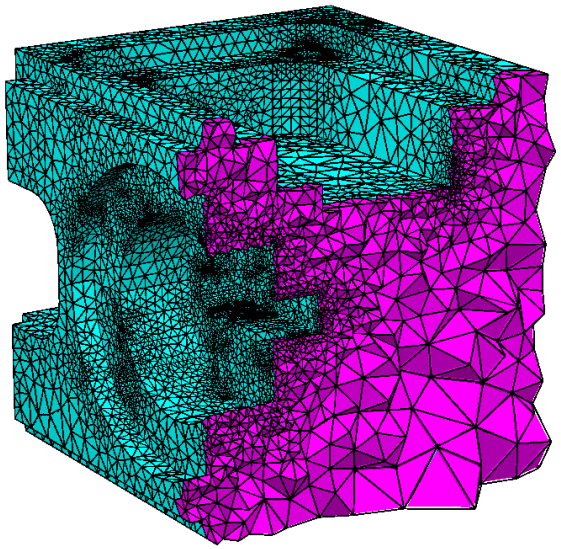
- (Mesh quality) Well-shaped tetrahedra, $\rho(t) \leq \rho_0, \forall t \in \mathcal{T}$.
- (Mesh size) Well-graded mesh, $\|\mathbf{v} - \mathbf{w}\| \geq \frac{\mathrm{lfs}(\mathbf{v})}{D+1}$, $D = \frac{(3+\sqrt{2})\rho_0}{\rho_0 - 2}$.
- (Mesh property) It is a conforming Delaunay tetrahedral mesh.



min. angle $= 14.5^o$
$7,862$ nodes, 1.5 sec.
1.5 sec.

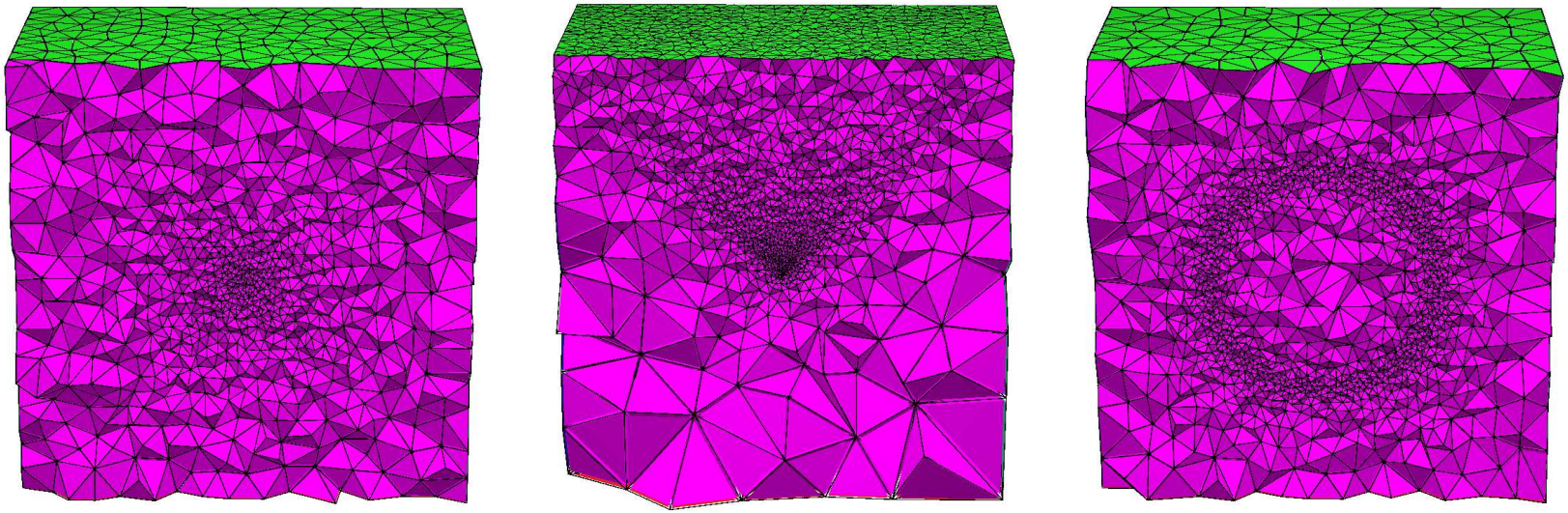


min. angle $= 20.7^o$
$14,653$ nodes, 2.5 sec.
2.5 sec.



min. angle $= 27^o$
$54,560$ nodes, 8.7 sec.
8.7 sec.

Mesh Adaptation

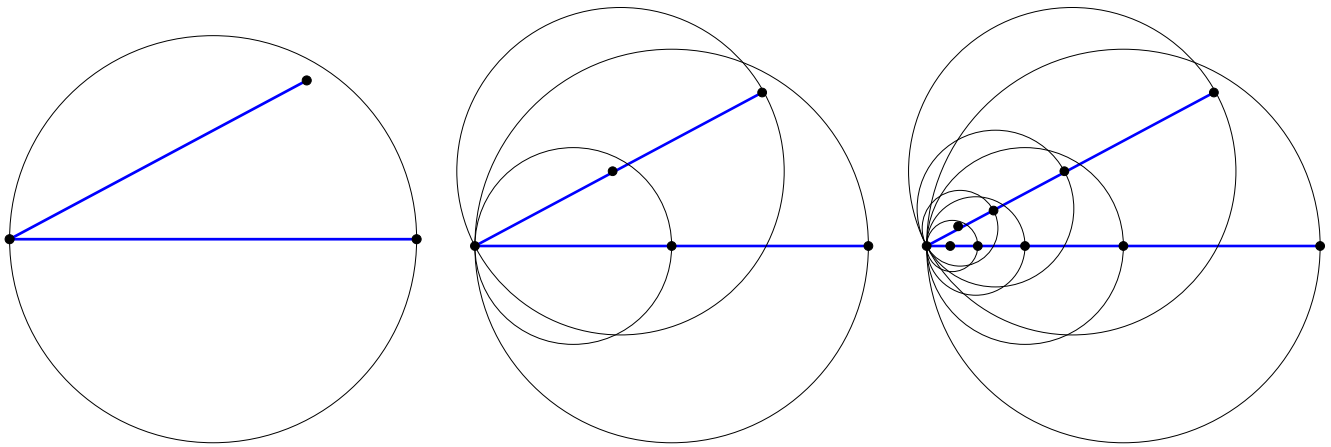


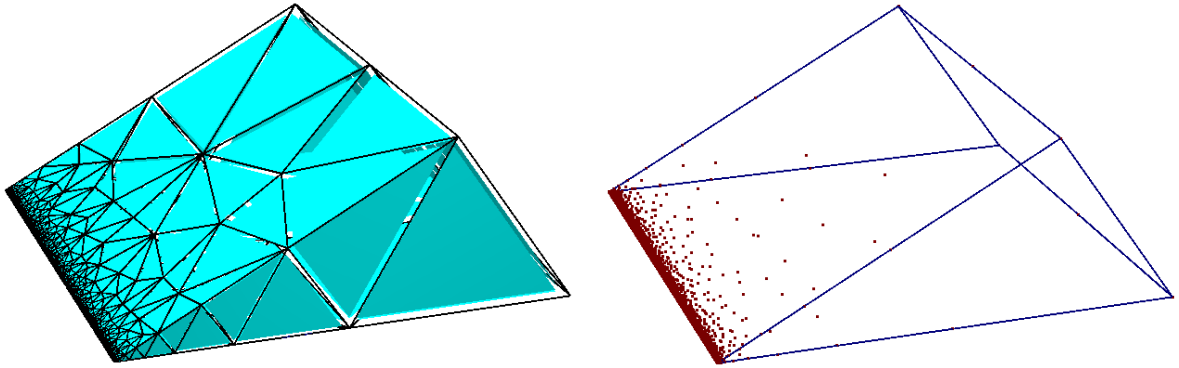
			S_v	L_v	S_v	L_v	S_v	L_v
	$<$	0.5	0	0	0	0	0	0
0.5	$-$	$1/\sqrt{2}$	58	0	0	0	0	0
$1/\sqrt{2}$	$-$	1	3221	1	283	0	0	0
1	$-$	$\sqrt{2}$	15062	113	10778	14	1927	49
$\sqrt{2}$	$-$	2	4246	3867	1187	1044	94186	12594
2	$-$	$2\sqrt{2}$	0	18606	0	11190	12276	95746
	$>$	$2\sqrt{2}$	0	0	0	0	0	0

Mesh adaptation examples from [**Si 2008**]

Small Angles

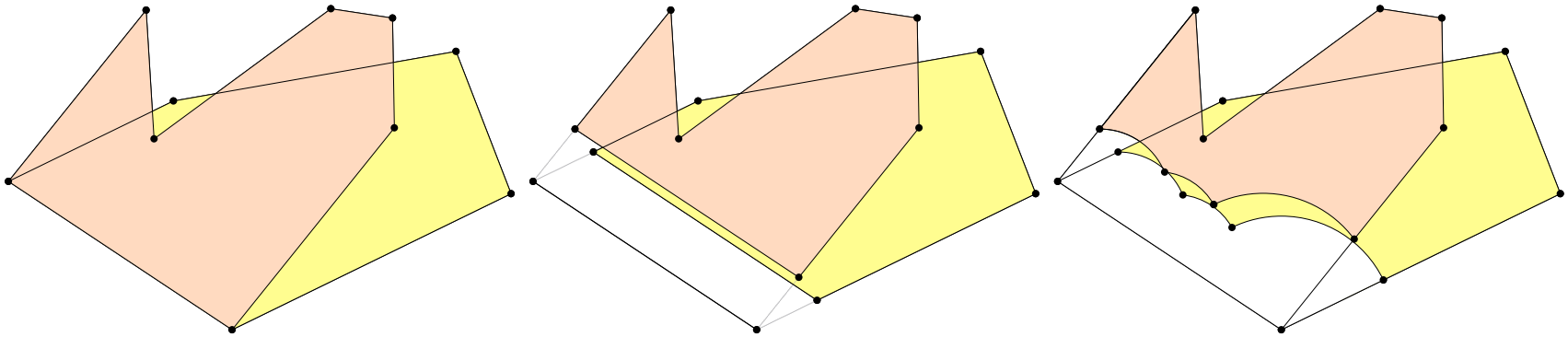
- No guarantee on termination when two boundary constraints meet at a small angle.
- Small angles pose a fundamental challenge to Delaunay refinement. Without special consideration, termination can not be guaranteed.





Handling Small Angles

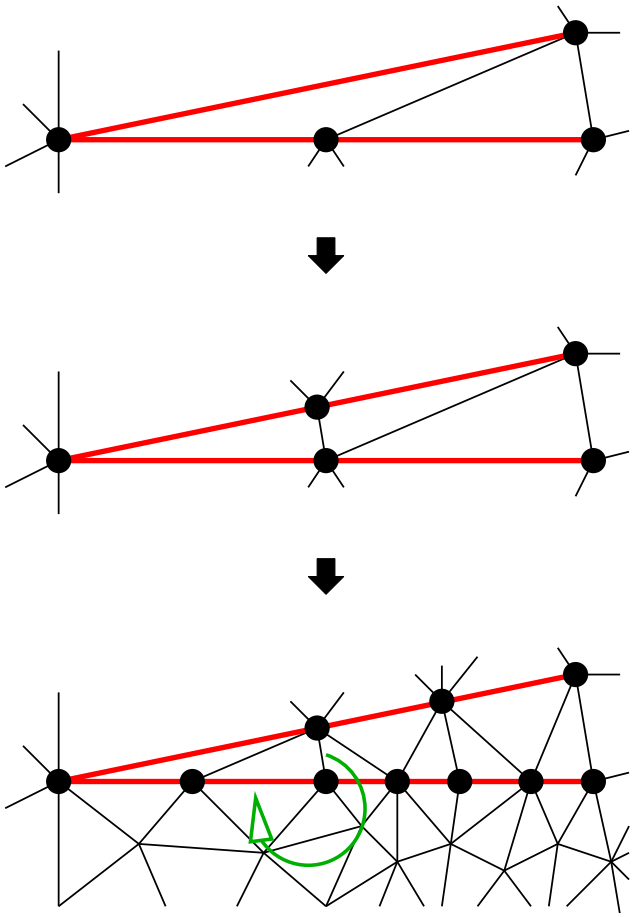
- The **basic idea**: Create protect regions to separate small input angles, e.g., [**Murphy, el al 2000**, **Cohen-Steiner, et al 2002**, **Cheng & Poon2003, Cheng & Dey 2004, 2005, Pav & Walkington 2004**].



Protect regions ([**Pav & Walkington 2004**]).

Constrained Delaunay Refinement [**Shewchuk & Si 2014**]

- Observation: small angles are "edge length reducers".



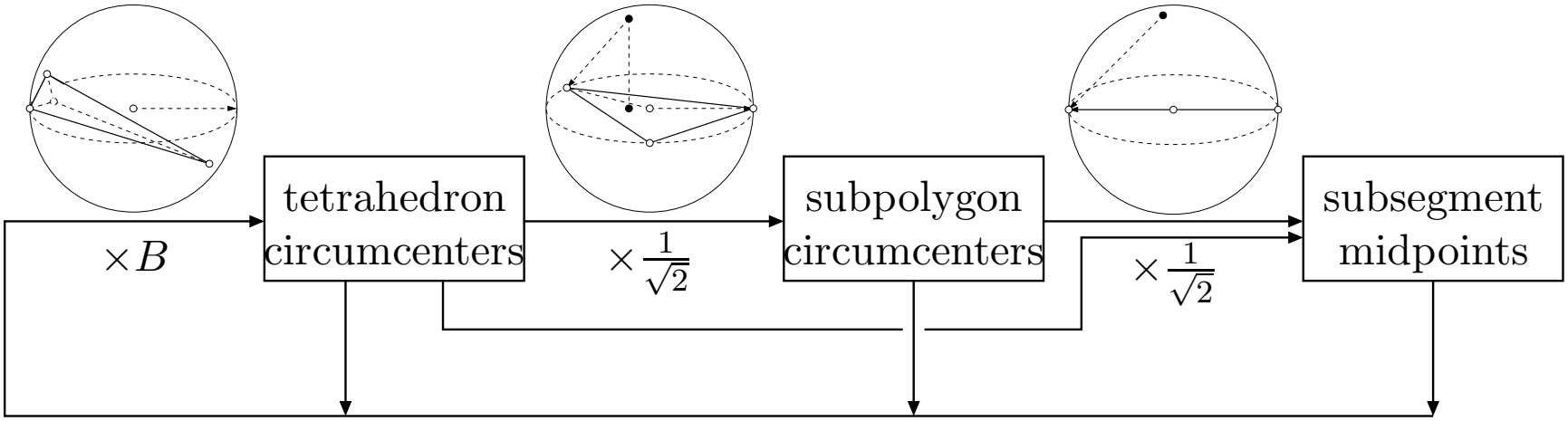
A subsegment is split.
New vertex encroaches upon another subsegment.

Another vertex is inserted, creating a very short edge.
Oops!

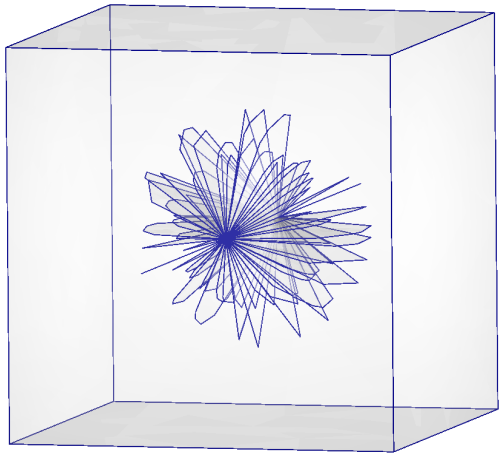
Skinny tetrahedra get split.
Small edge lengths propagate.
Subsegment split again!

Constrained Delaunay Refinement [**Shewchuk & Si 2014**]

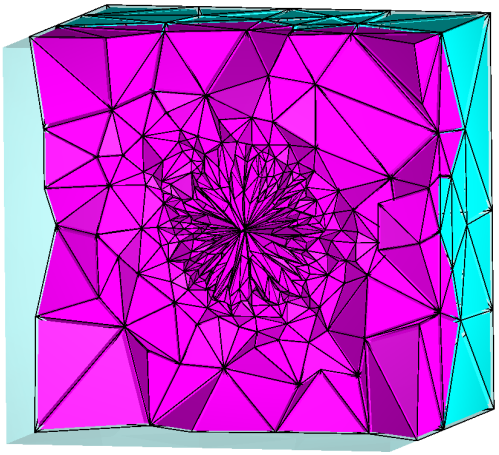
- The idea: Fake the edge lengths at small angles.
  - Store for each vertex a *relaxed insertion radius* which satisfies the **Lemma** [Ruppert 1995, Shewchuk 1998] of insertion radius.
  - If a new edge is "unreasonably short", i.e., it violates the Lemma, store the insertion radius of its parents for the new vertex.
  - Split a badly-shaped tetrahedron only if its relaxed radius-edge ratio ≥ 2.
- Now the flow chart applies to the relaxed insertion radii – The algorithm terminates.



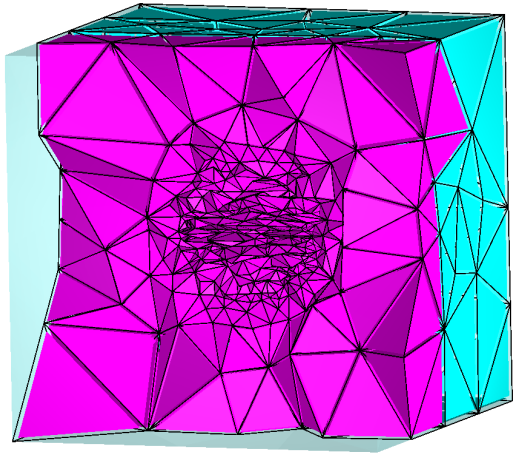
Constrained Delaunay Refinement [**Shewchuk & Si 2014**]



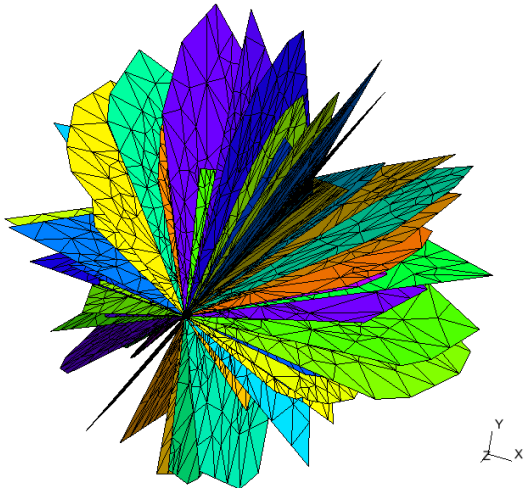
Input `test-64-6`
161 vertices, 70 polygons



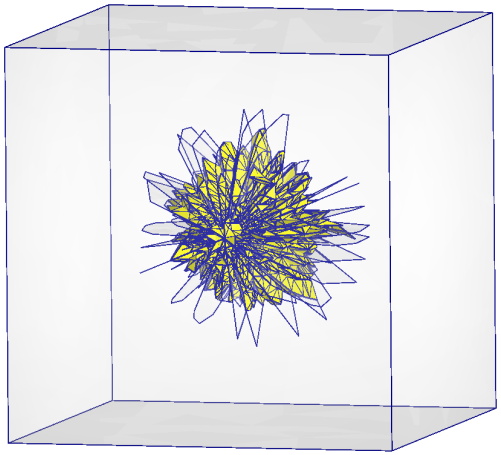
Tet mesh, $3,733$ vertices
(cut along the Z-axis)



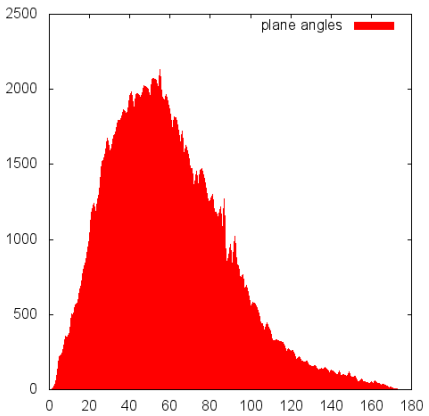
Tet mesh, $23,727$ tets
(cut along the Y-axis)



refined "fan blades"



remaining skinny tetrahedra
(radius-edge ratios > 2)



plane angles

Further Topics

Mesh Improvement

- Mesh improvement (mesh optimization) is a very important post-process in generating quality tetrahedral meshes.
- Typical methods and techniques for mesh improvement combine vertex smoothing, mesh reconnection, and vertex insertion/deletion, see [**Freitag & Olliver-Gooch 1997, Klingner & Shewchuk 2008**].
- The convergence of the typical "hill climbing" mesh improvement process is very hard to achieve.
- New techniques for mesh improvement needs to be developed.

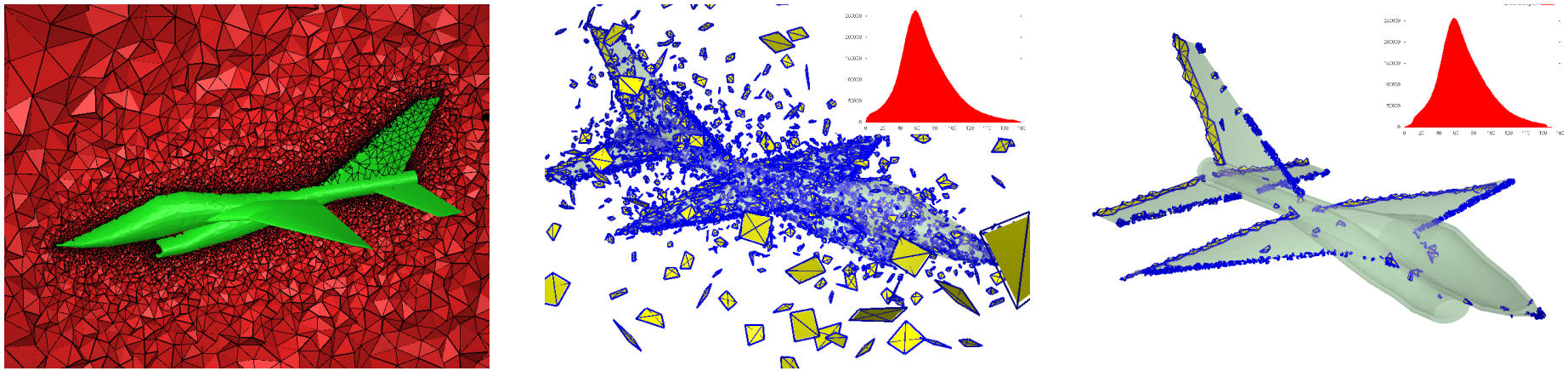
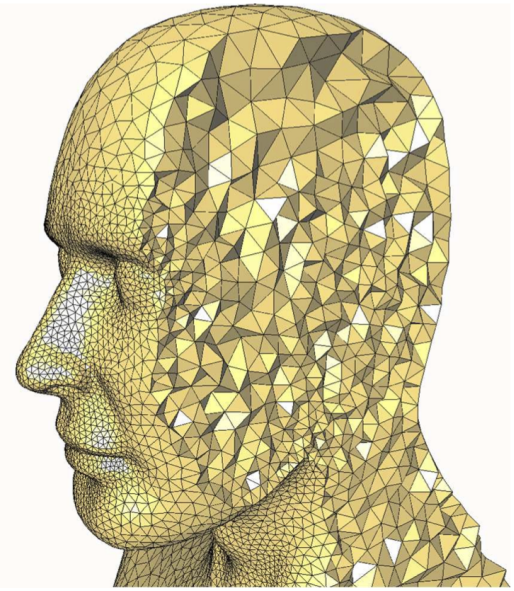


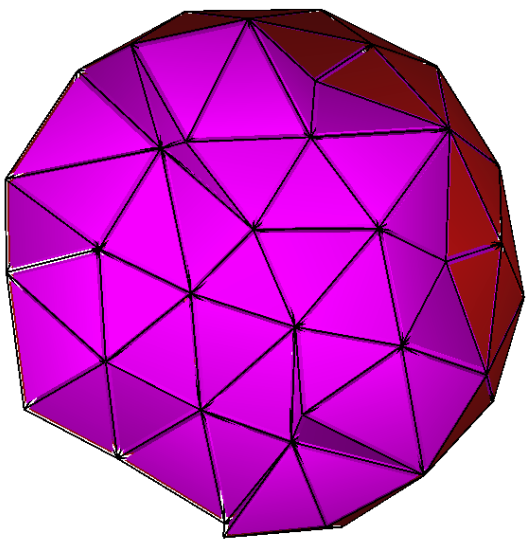
Figure: Left: A tet mesh from constrained Delaunay refinement. Middle: A highlight of the bad quality tets from the left mesh. Right: A highlight of the bad quality tets after mesh improvement.

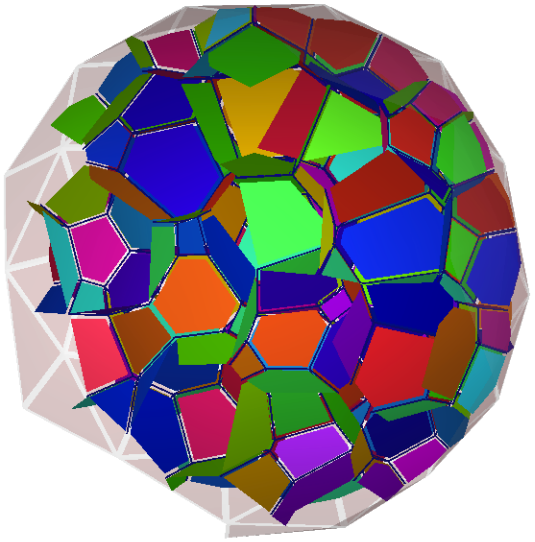
Variational Mesh Generation Methods

- Variational Voronoi-Delaunay methods aim at generating and improving good quality tetrahedral mesh simultaneously through mesh optimization.
- The key idea is to utilize a convex mesh-dependent energy function whose local or global minimum exists. Two such functionals have been proposed in [**Du & Wang 2003**] and [**Chen & Xu 2004**]. The minimum of these two functionals are optimal Delaunay triangulation (ODT) and centroidal Voronoi tessellation (CVT).
- However: both methods suffer the same issues (slivers and small angles) as those of Delaunay refinement.
- The efficiency of these methods needs to be improved.



ODT [**Alliez et al, 2005**]

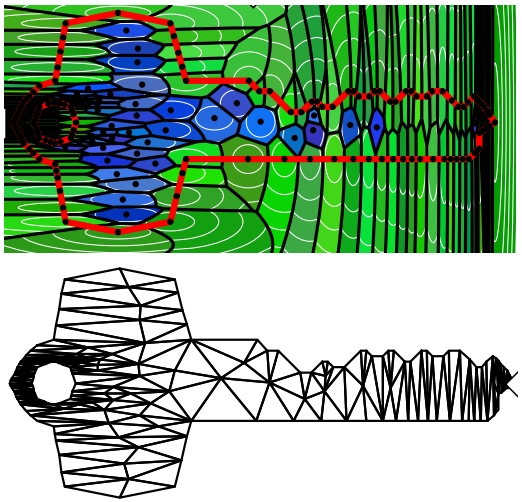




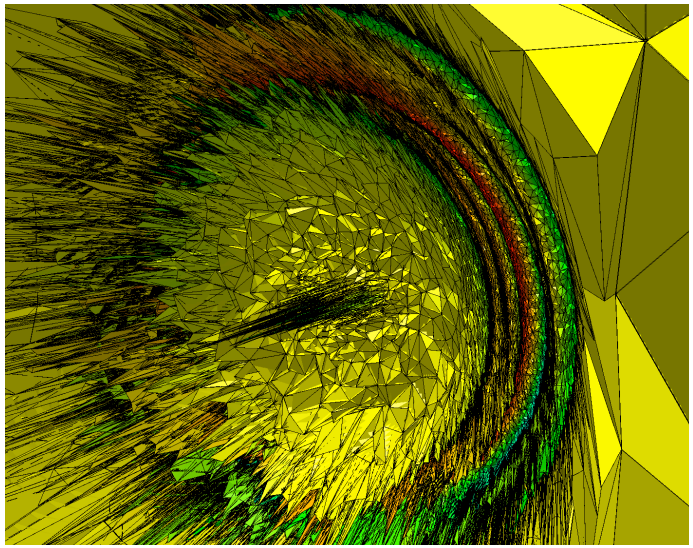
CVT [**Levy & Liu 2010**]

Anisotropic mesh generation

- Ellipsoidal packing: [**Li, Teng, & Üngör 1999**, **Yamakawa & Shimada 2003**]
- Anisotropic mesh optimzation: [**Bossen & Heckbert 1996**]
- Anisotropic Voronoi diagrams: [**Labelle & Shewchuk 2003**]
- Anisotropic Delaunay refinement: [**Borouchaki & George et al 1997, Frey & Alauzat 2004**, **Dobrzynski & Frey 2008**, **Boissonnat et al 2008**]



Anisotropic Voronoi diagram. Image from [**Labelle & Shewchuk 2003**]



Courtesy of F. Alauzet