Unstructured Mesh Generation and Adaptation

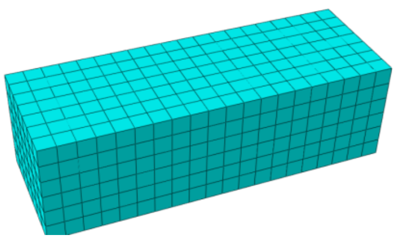
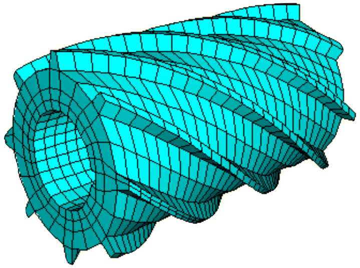
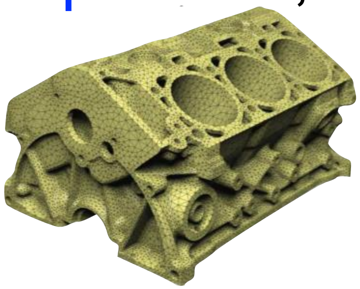
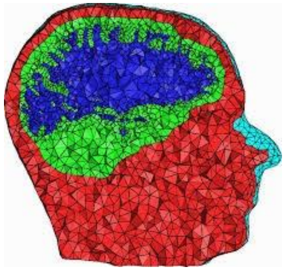
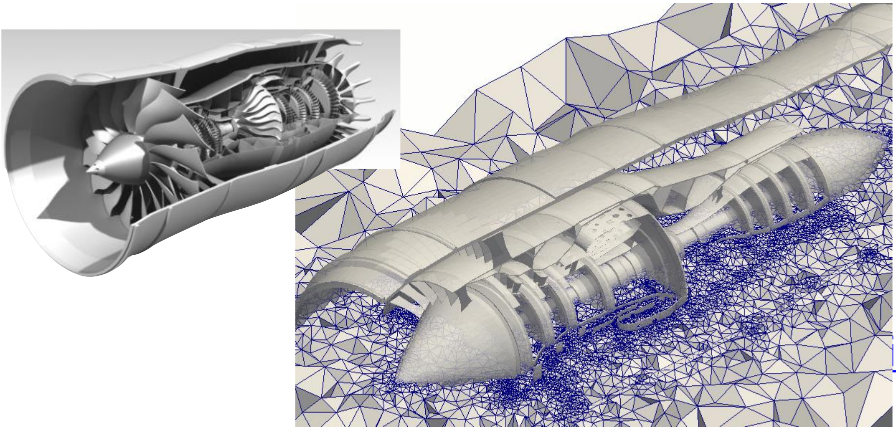
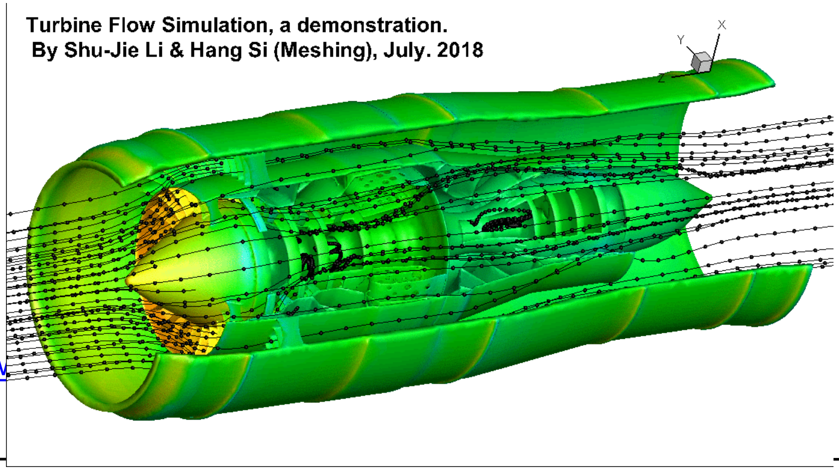
Hang Si
Weierstrass Institute for Applied Analysis and Stochastics (WIAS) Berlin

2018-12-02
Winter School, NUMGRID 2018, Moscow

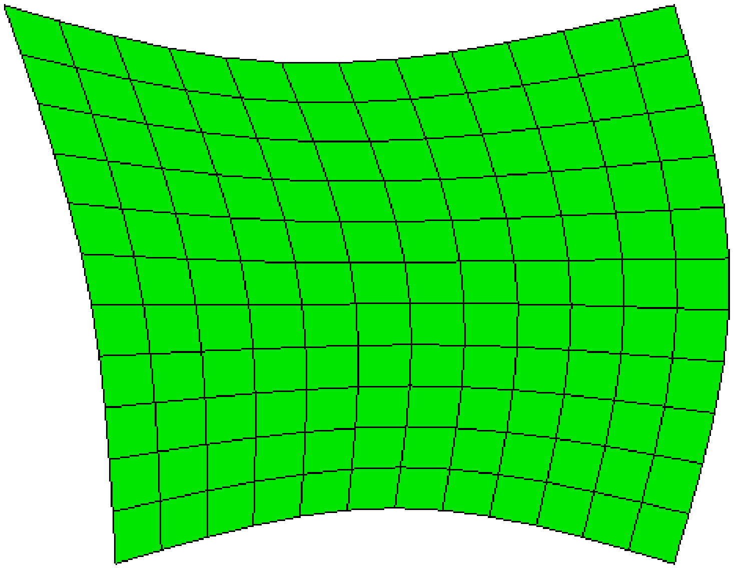
Leibniz Gemeinschaft

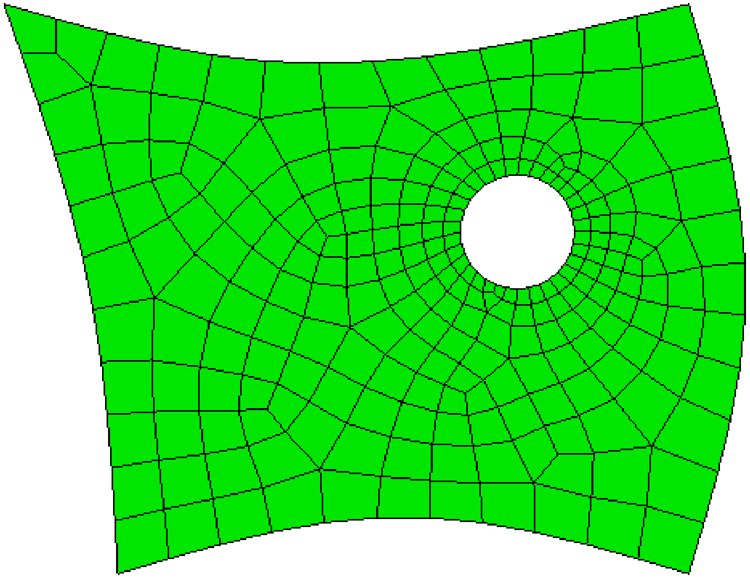
Motivation

1. Mesh generation is the process of partitioning a complex shape into a collection of simple shapes.
2. Mesh generation has many applications, in areas like geography, computer graphics, computer-aided design, and numerical solution of differential equations, …

Structured vs. Unstructured





Structured

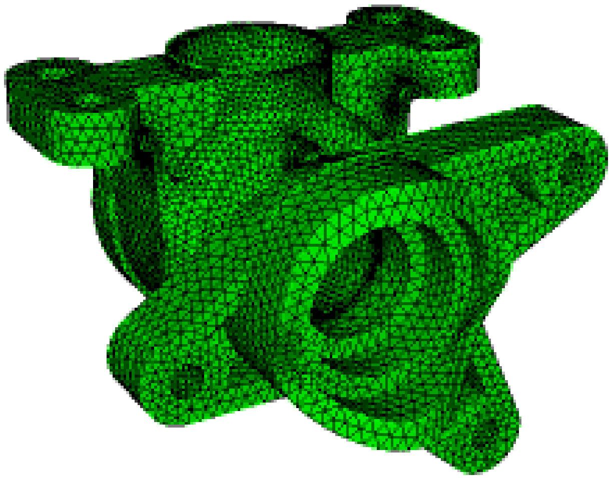
1. Interior node valence is constant.
   ie. number of elements at each interior node=4
2. Meshing algorithm relies on specific topology constraints.
   ie. number of sides=4

Unstructured

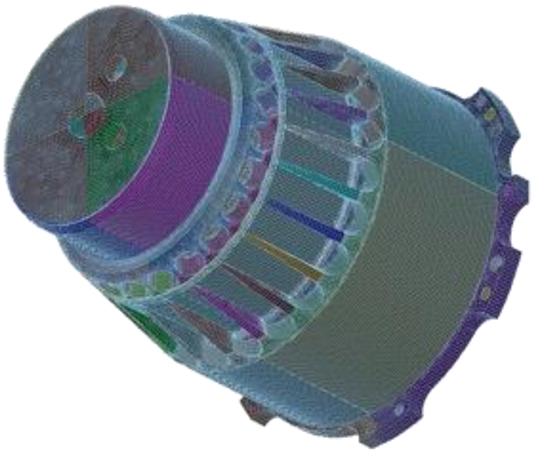
1. Interior node valence varies.
   ie. number of elements at each node=3,4,5…
2. Meshing algorithm applies to arbitrary topology
   ie. number of sides is arbitrary

Tet Meshing Vs. Hex Meshing

Tet Meshing
1. Fully Automated, mostly push-button
2. Generate millions of elements in minutes/seconds
3. User time generally minutes/hours
4. Can require 4-10X number of elements to achieve same accuracy as all-hex mesh
5. Tet-Locking phenomenon for linear tet results in stiffer physics

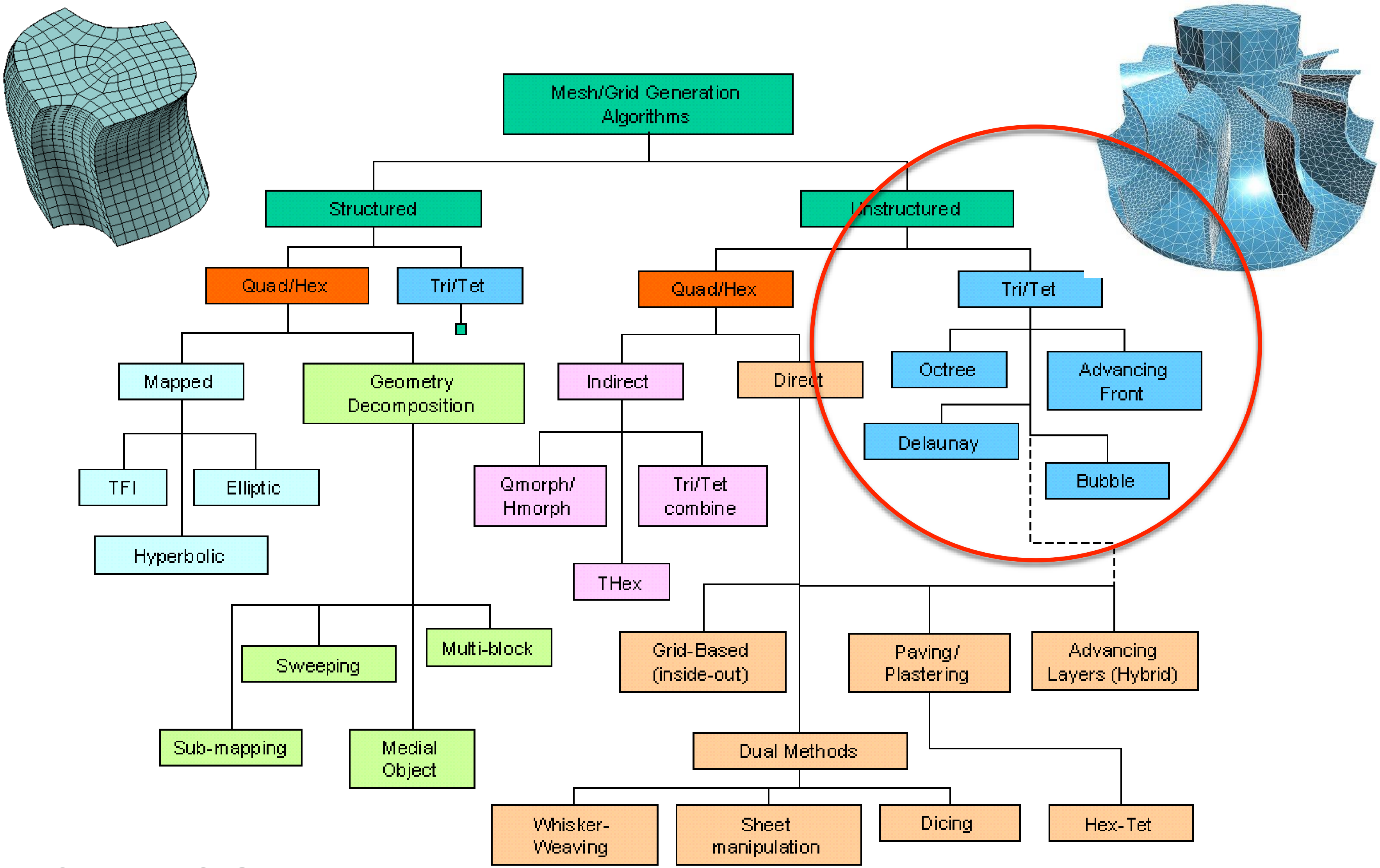


Hex Meshing
1. Partially Automated, some Manual
2. Can require major user effort/expertise to prepare geometry to accept a hex mesh
3. User time to generate mesh may be typically days/weeks/months
4. Computational methods may prefer or require hex element
5. Preferred by most analysts for solution accuracy



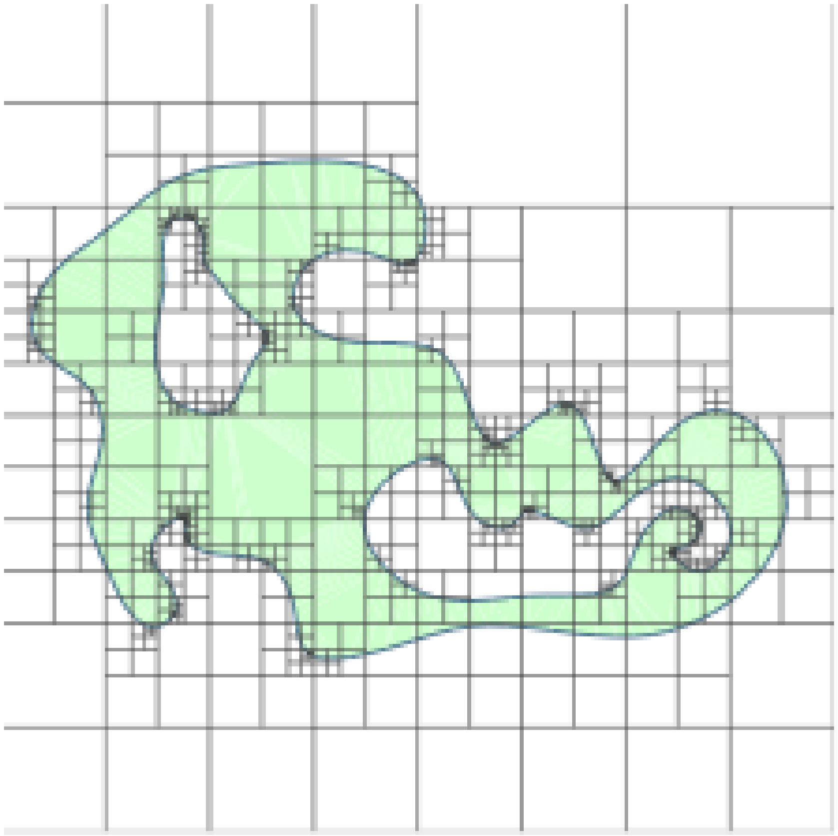
Courtesy S. Owen, Short Course 23rd IMR

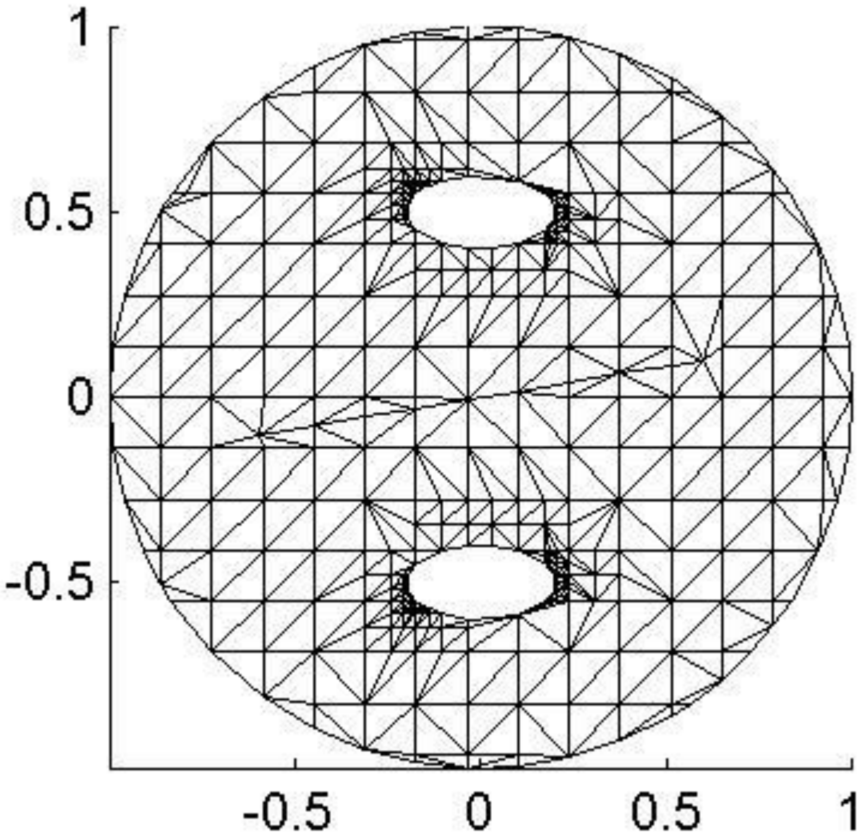
Mesh generation methods



Courtesy S. Owen

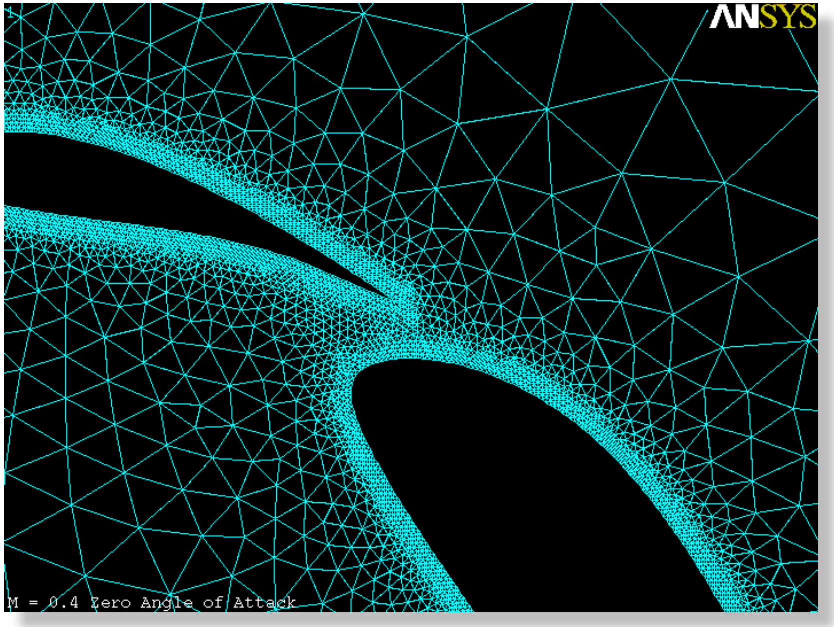
Quadtree-Octree methods

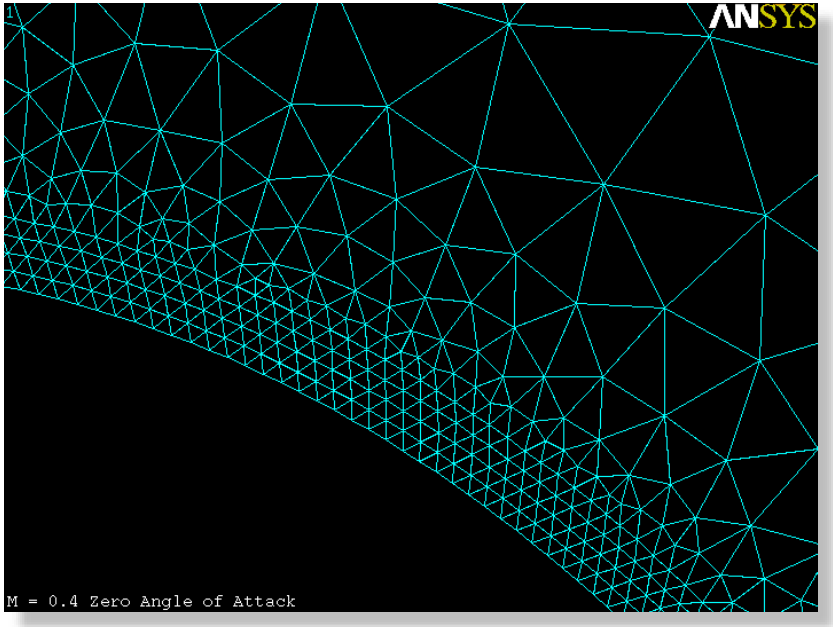


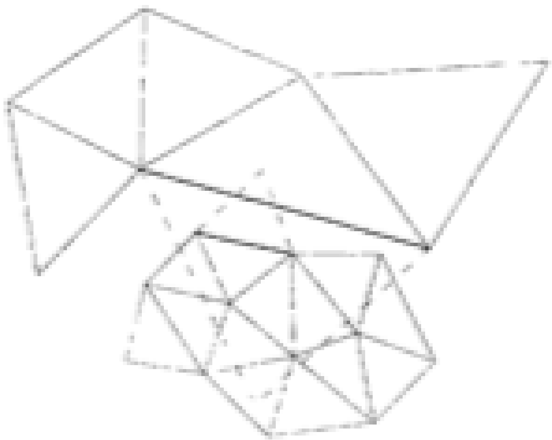


QMG

Advancing front







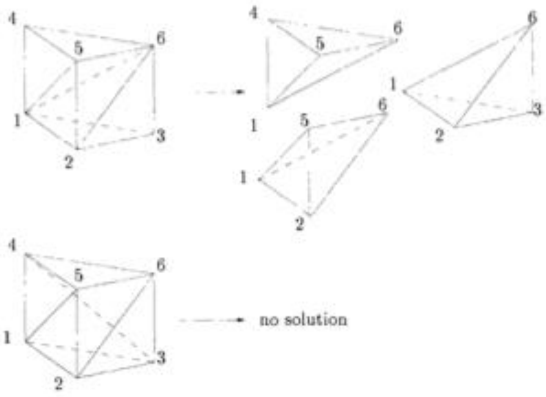
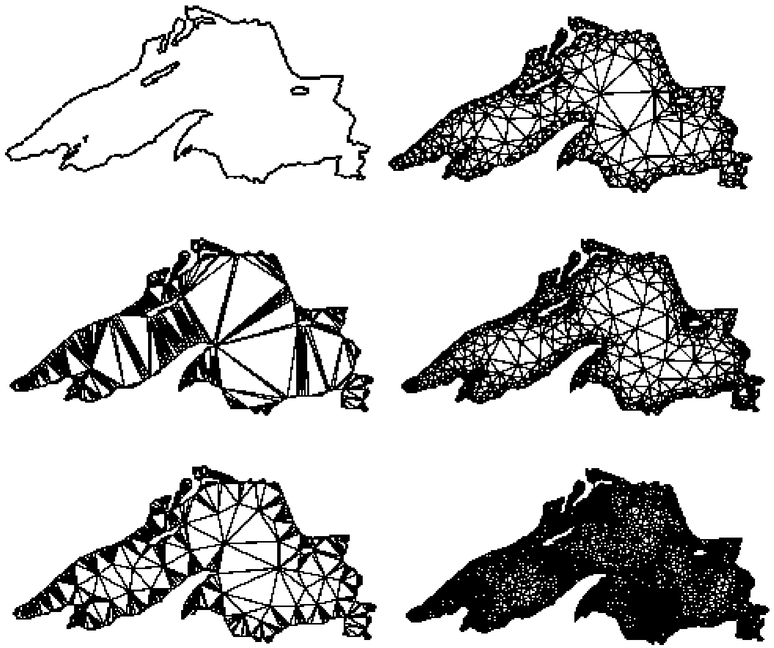
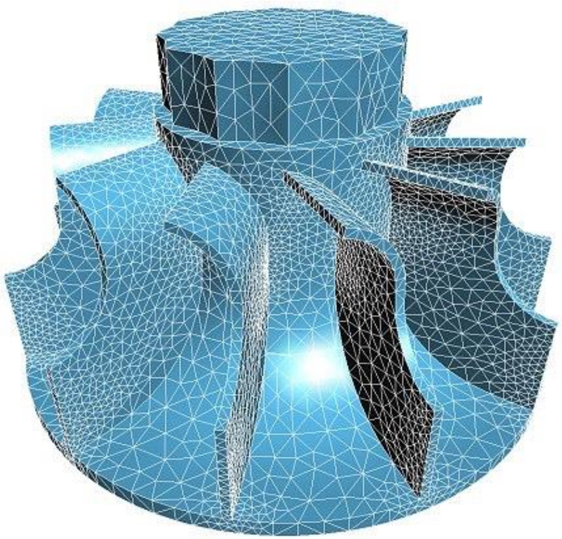


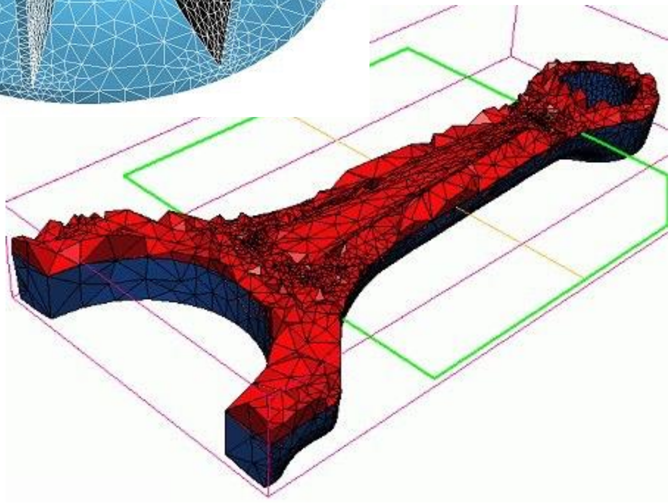
Figure 6.7: *Schönhardt polyhedron : valid and non-decomposable (without adding an internal point) constrained triangulation of a regular prism.*

Delaunay-based methods (with theoretical guarantees)



Triangle
Jonathon Shewchuk
http://www-2.cs.cmu.edu/~quake/triangle.html





Tetmesh-GHS3D
INRIA, France
http://www.simulog.fr/tetmesh/

Softwares

- Commercial:
  - `Tetmesh-GHS3D`, INRIA, Rocquencourt, Distene France.
  - `MeshSim`, SCOREC, RPI, Simmetrix Inc. USA.
  - `VisTools/Mesh`, AeroAstro, MIT, Vki Inc. USA.
  - `SolidMesh`, AFLR mesh generator, SimCenter, Mississippi State Uni.
- Open source:
  - `Netgen`, TU Vienna.
  - `Gmsh`, Uni. Liege & Uni. Catholique de Louvain.
  - `GRUMMP`, University of British Columbia.
  - `Pyramid*`, UC Berkeley.
  - `CGALmesh`, INRIA, Sophia-Antipolis.
  - `TetGen`, Weierstrass Institute, Berlin.

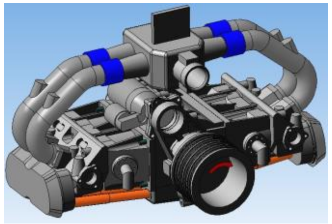
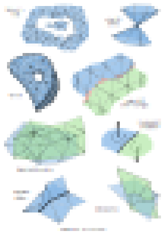
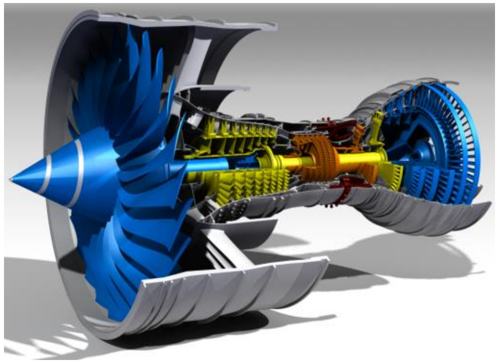
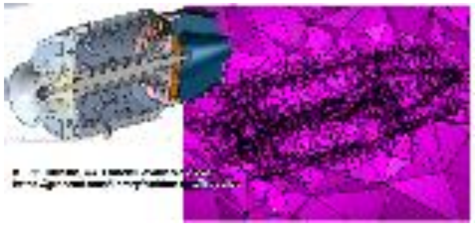
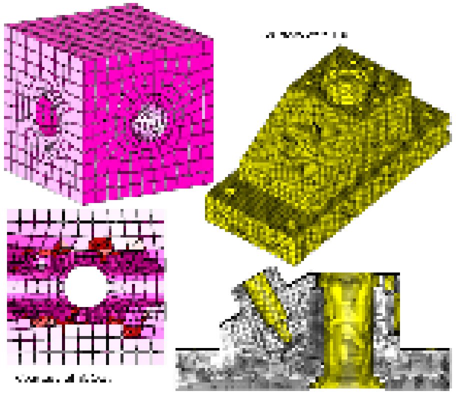
Comprehensive lists of meshing softwares are found in

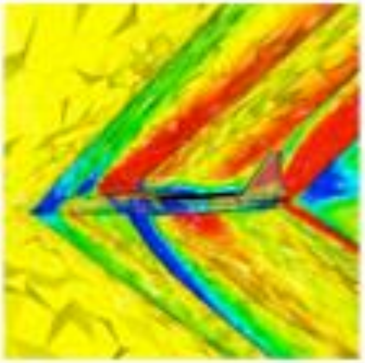
- Steven Owen, **A Survey of Unstructured Mesh Generation Technology**, Proceedings, 7th International Meshing Roundtable, Sandia National Lab, pp.239-267, October 1998.
- Robert Schneiders, **Mesh Generation & Grid Genration on the Web**, `http://www.robertschneiders.de/meshgeneration/meshgeneration.html`.

The Challenges

1. CAD geometry preparation, cleaning.
2. 3d surface and volume mesh generation.
3. Mesh adaptation, anisotropic meshes.

Automation, Robustness, Efficiency, …

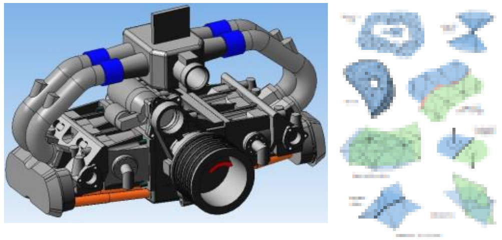


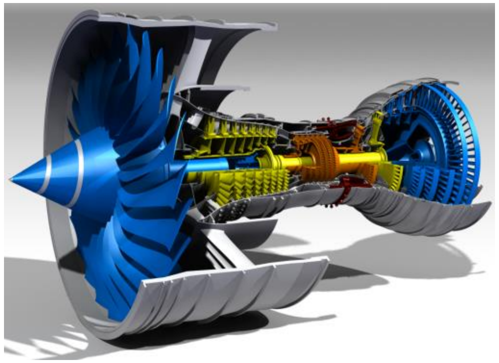
Images from Adrien Loiselle s Phd

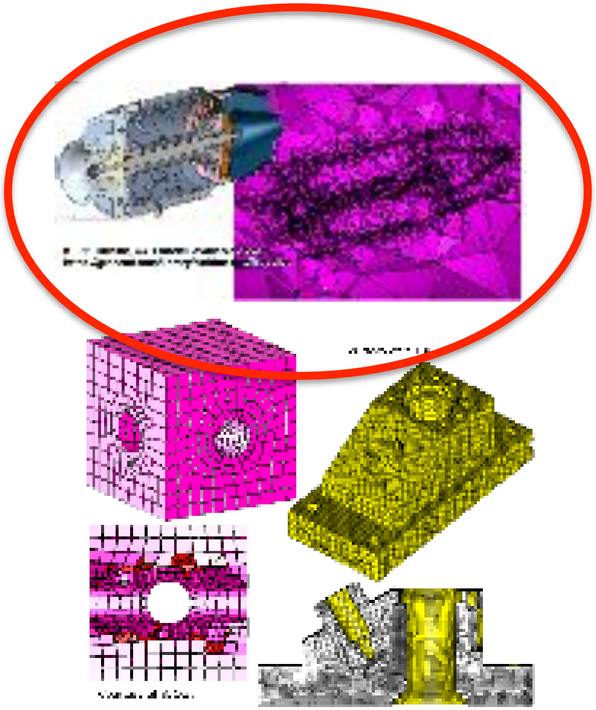
Topics of this course

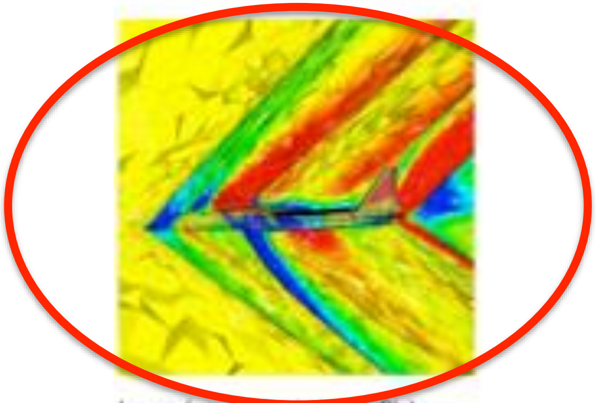
1. CAD geometry preparation, cleaning.
2. 3d surface and volume mesh generation.
3. Mesh adaptation, anisotropic meshes.

Automation, Robustness, Efficiency, …





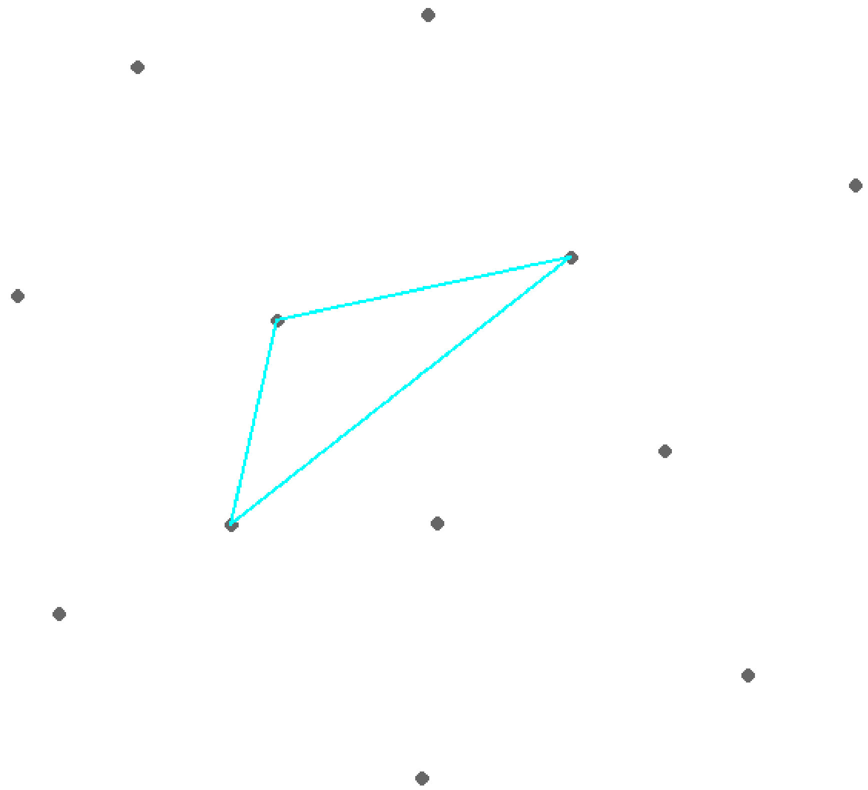




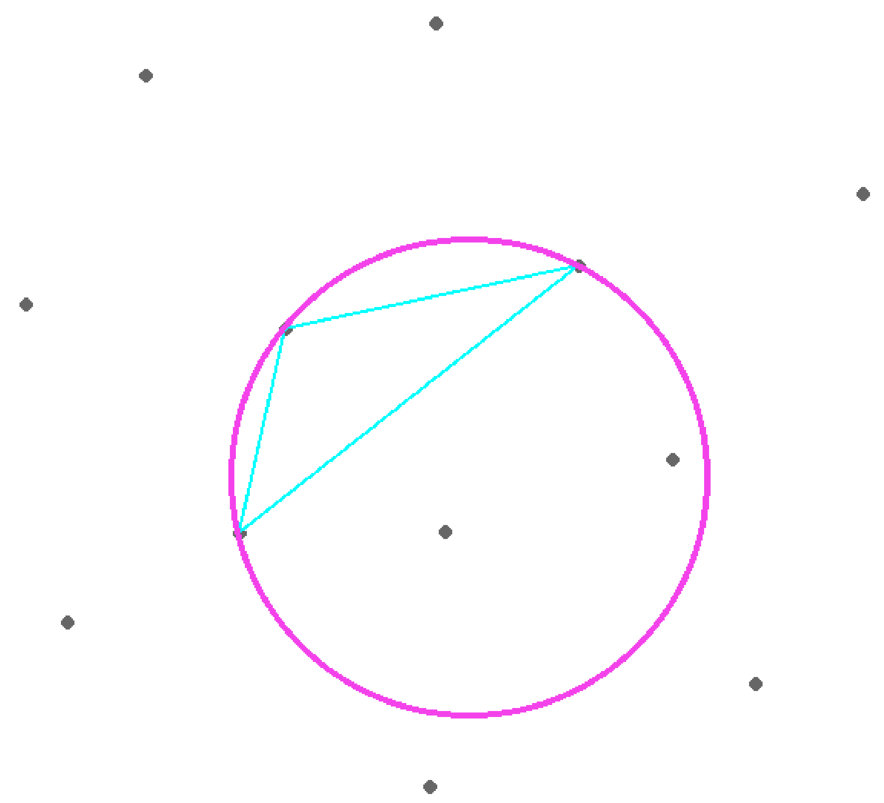
Outline

Delaunay Triangulations

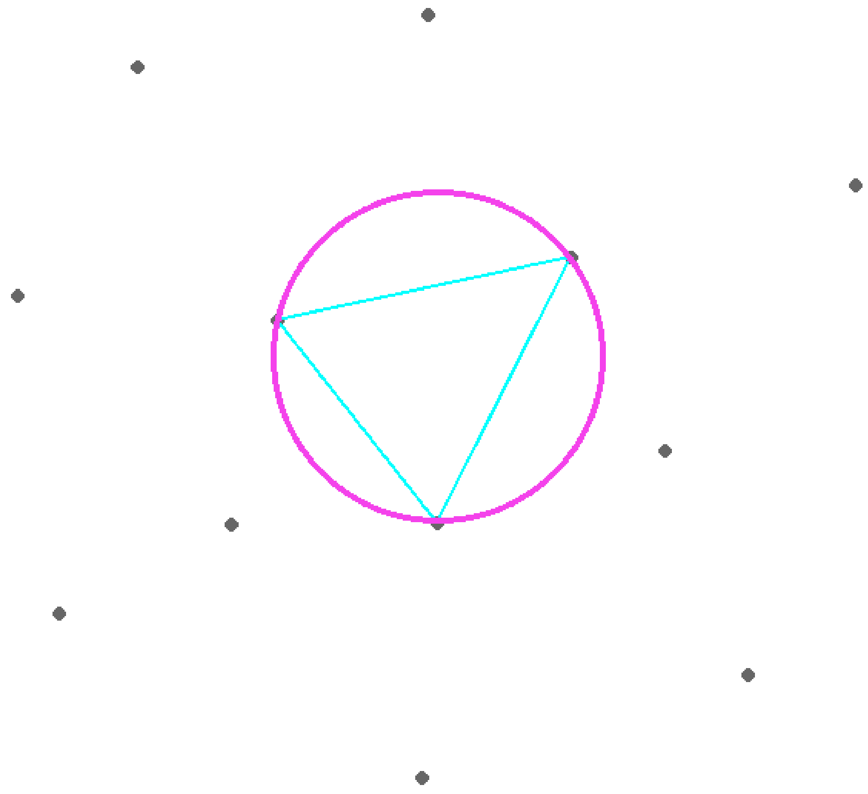
A finite point set S in the plane



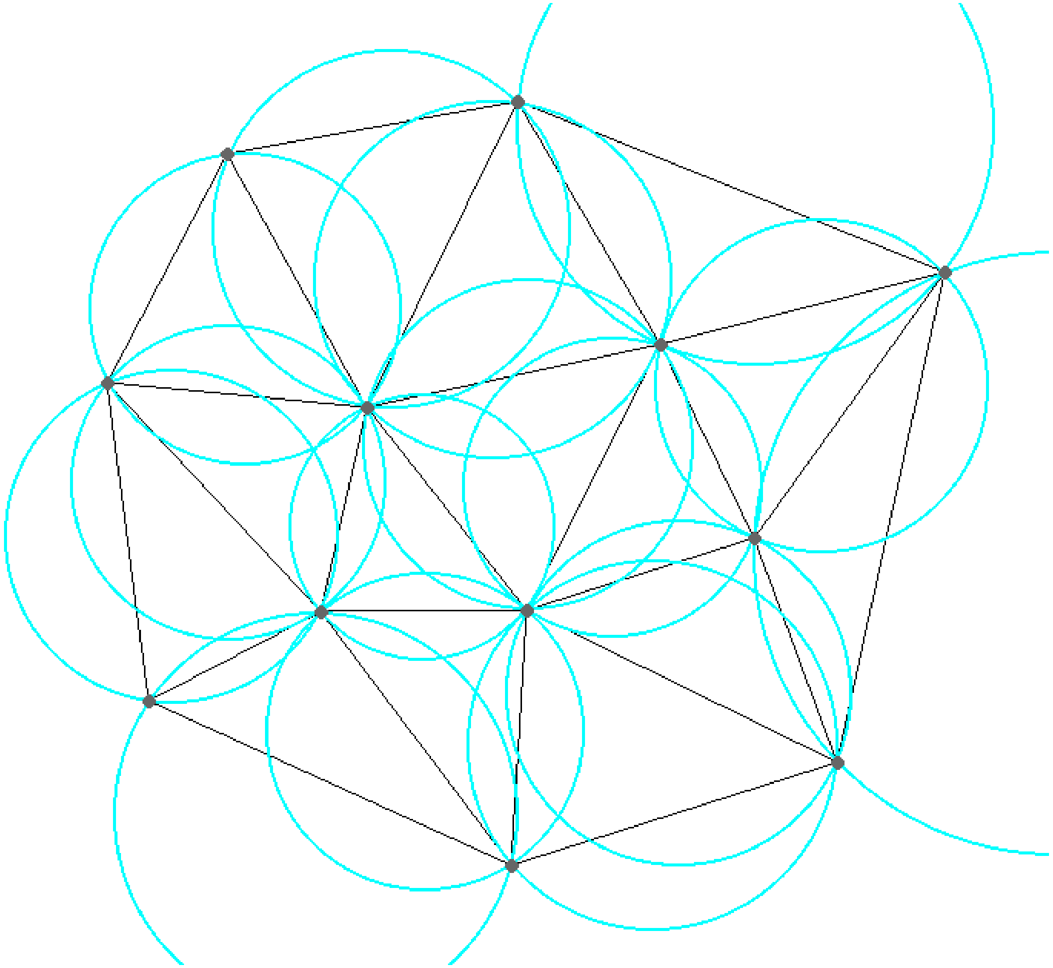
A triangle of S



the circumcircle of a triangle



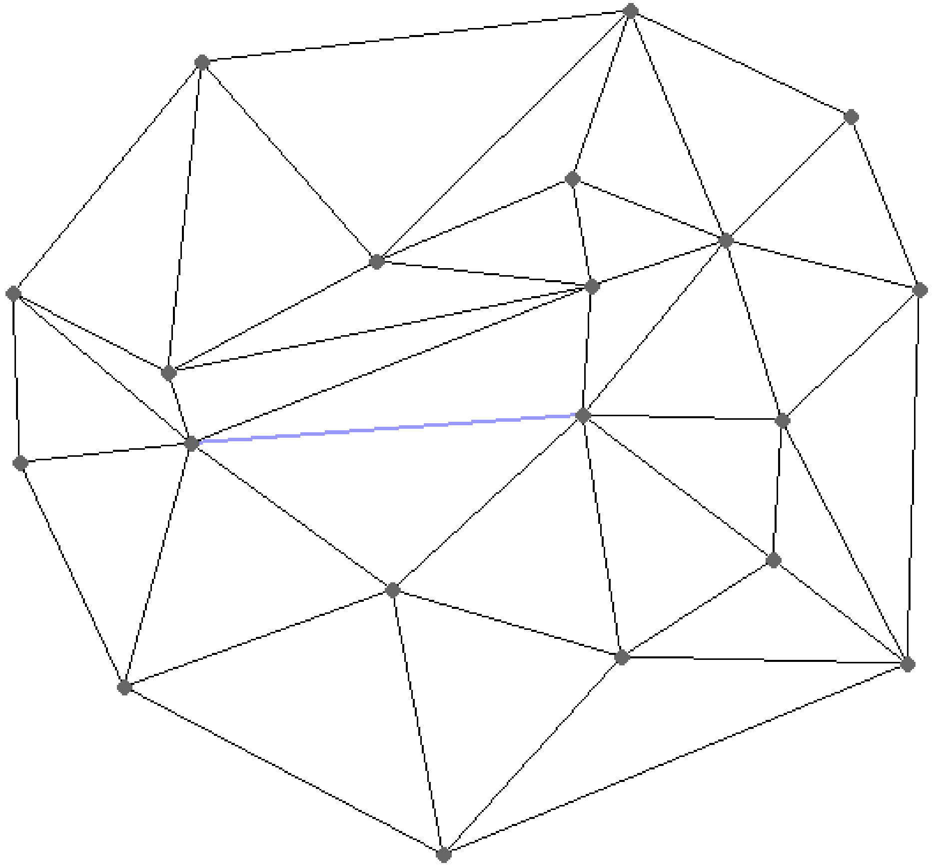
A triangle is called a Delaunay triangle if its circumcircle contains no other vertices of S.



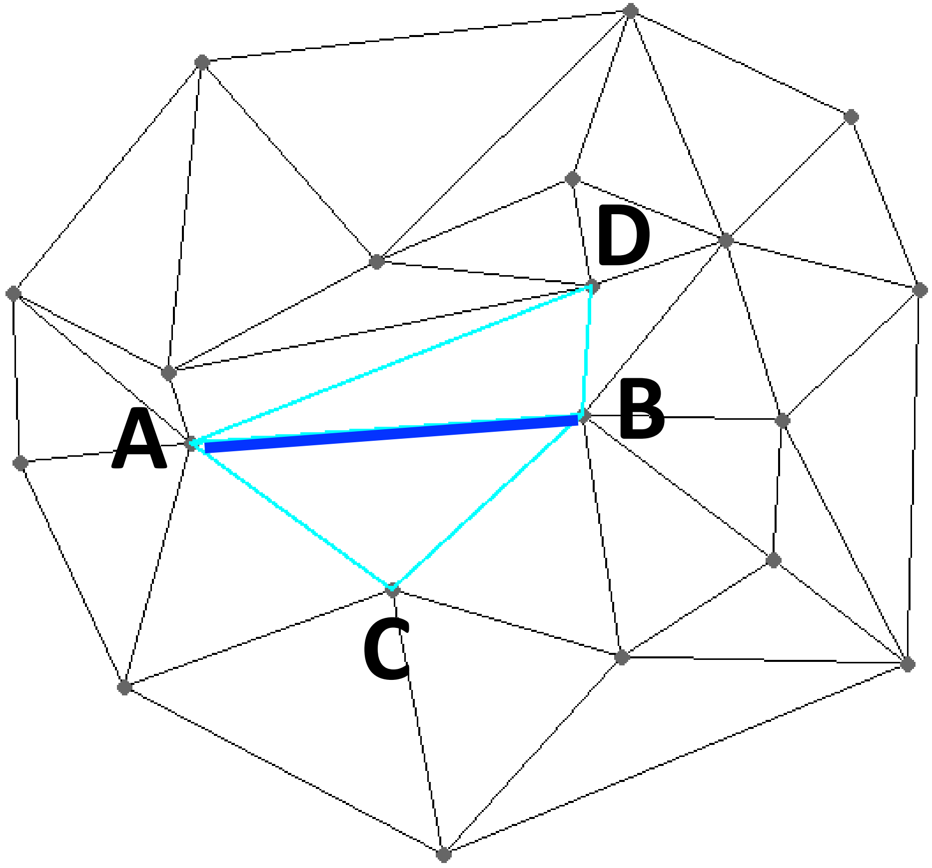
A triangulation of S is called Delaunay triangulation if all of its triangles are Delaunay triangles.

Geometric properties of Delaunay triangulations

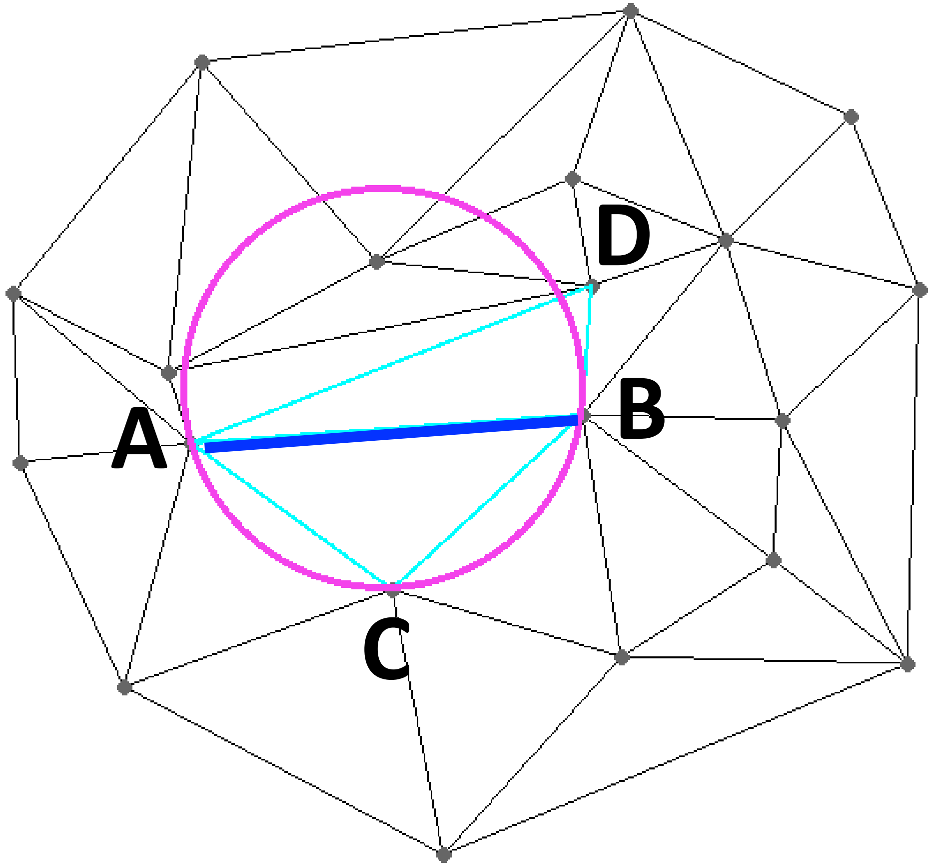
- DT maximizes the minimum angle of traingles.
  Lawson (1977 [?]) and Sibson (1978 [19])
- DT maximizes the arithmetic mean of the radius of inscribed circles of the triangles. Lambert (1994 [14])
- DT minimizes roughness (the integral of the squared gradients). Rippa (1990 [17])
- DT minimizes the maximum containing radius (the radisu of the smallest sphere containing the simplex).
  D'Azevedo and Simpson 1989 [6], Rajan (1991 [16])



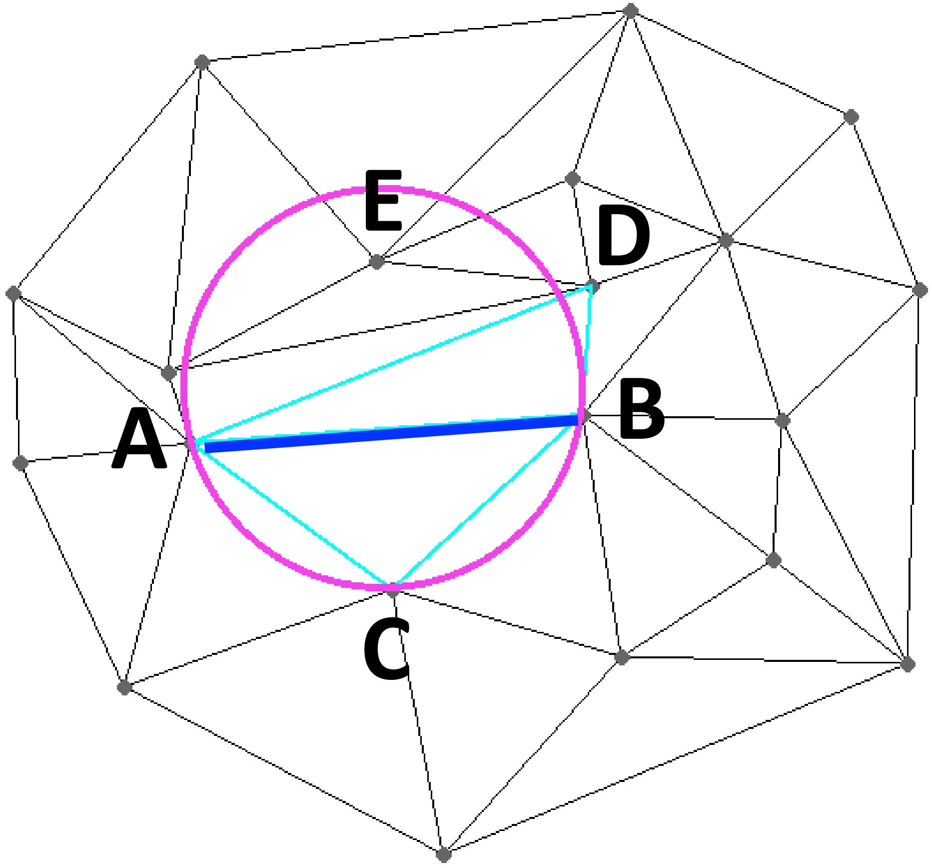
Given a triangulation of S, how to know whether it is Delau



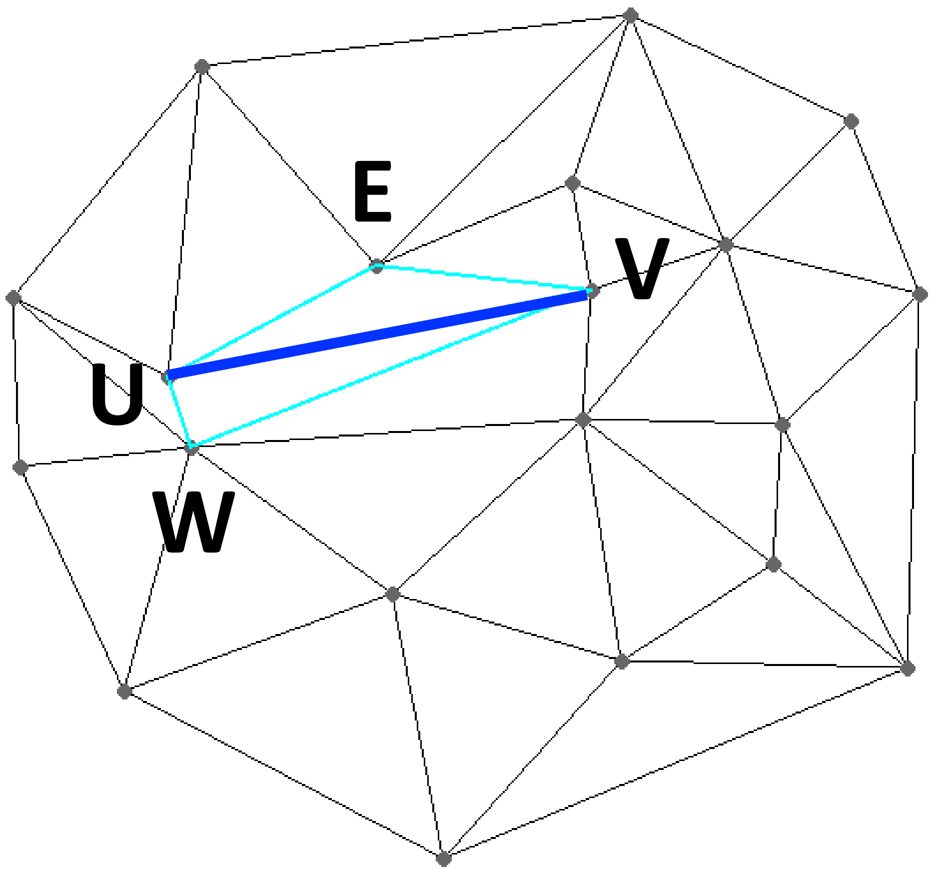
Let AB be an edge of T, it is shared by two triangles, ABC and ABD of T.



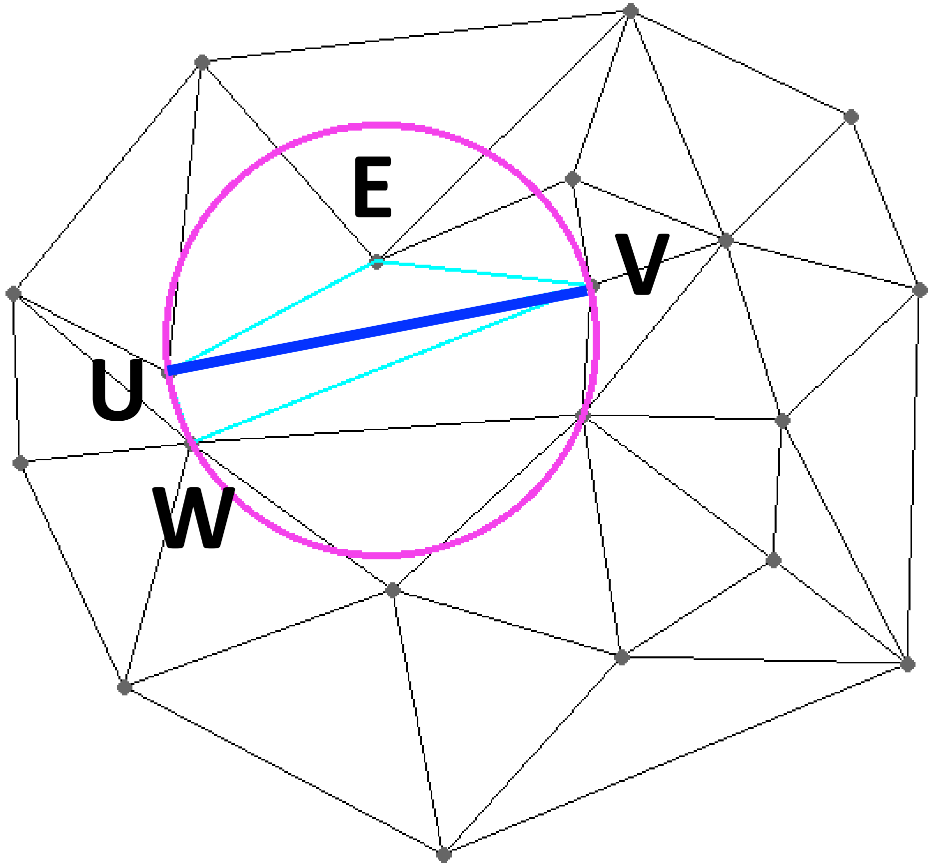
Ab is locally Delaunay if the circumcircle of ABC does not



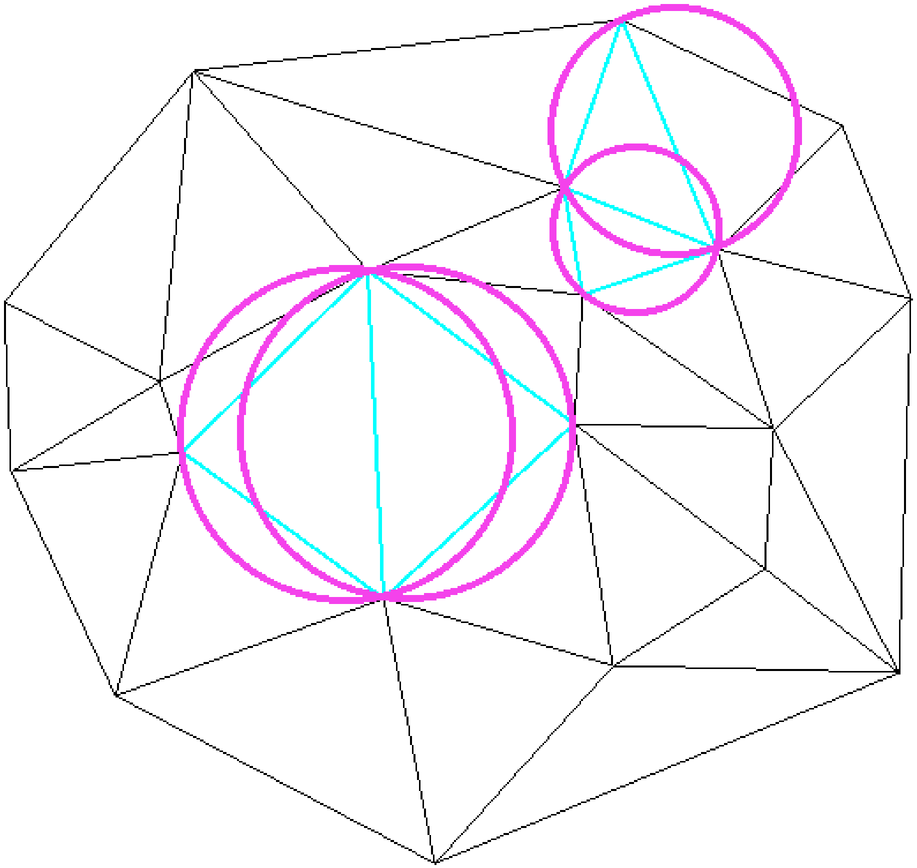
Even AB is locally Delaunay, it might not be a Delaunay



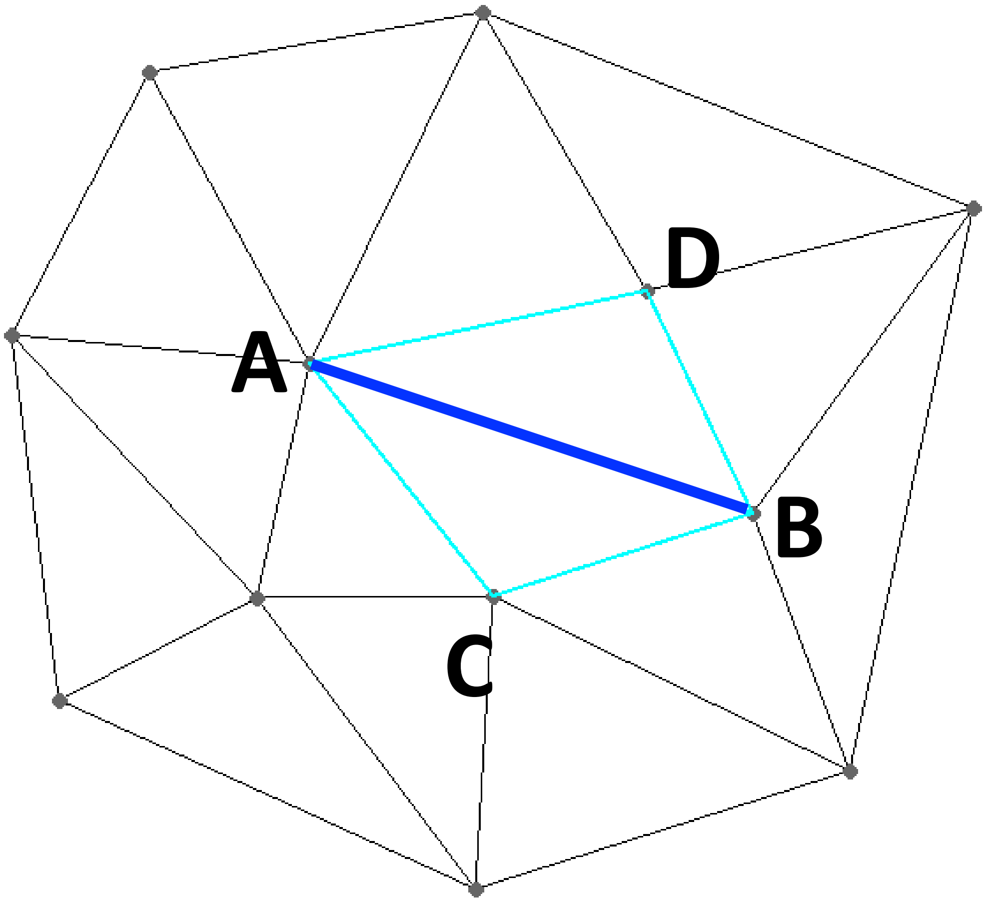
Let UV be an edge of T



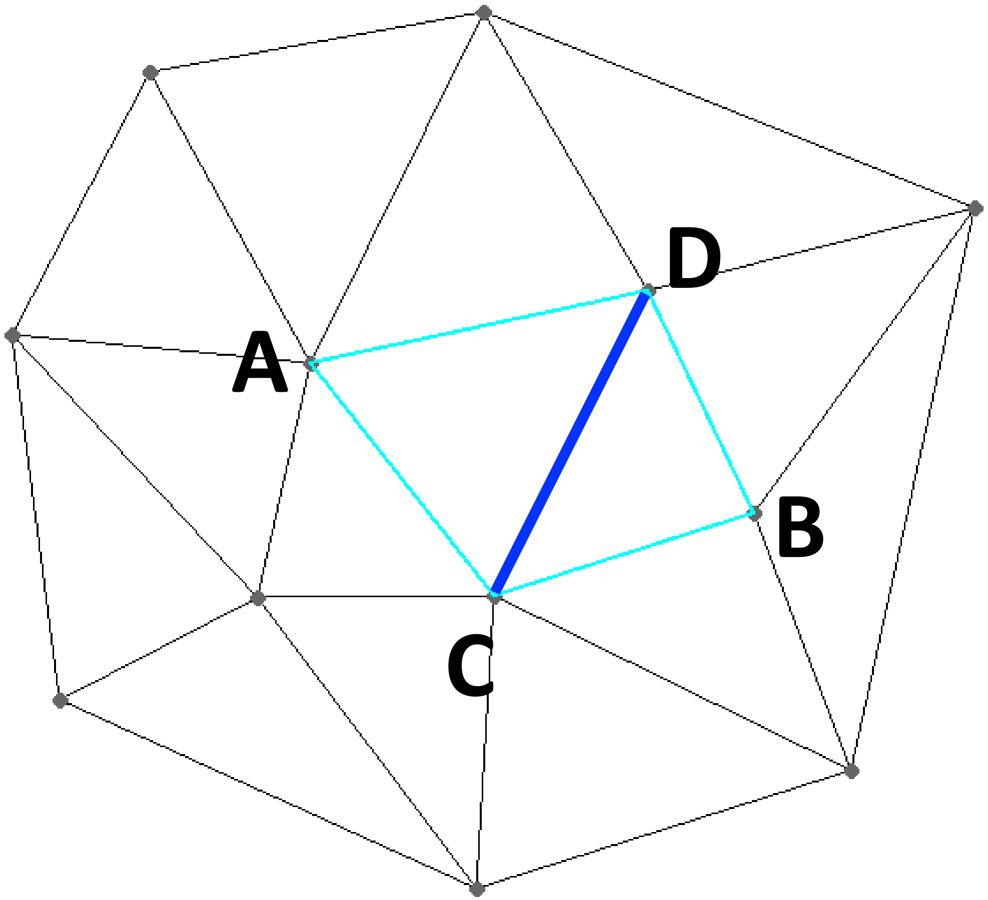
UV is not locally Delaunay



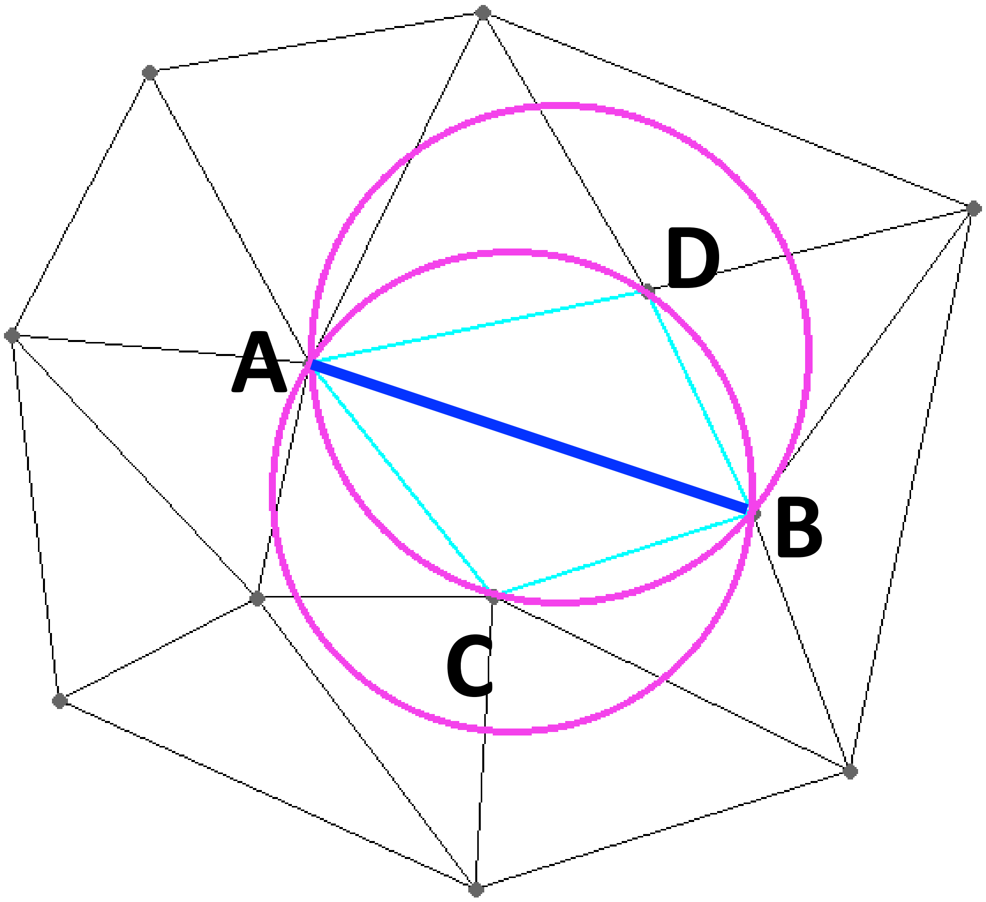
The Delaunay Lemma: If every edge of T is locally Delauna



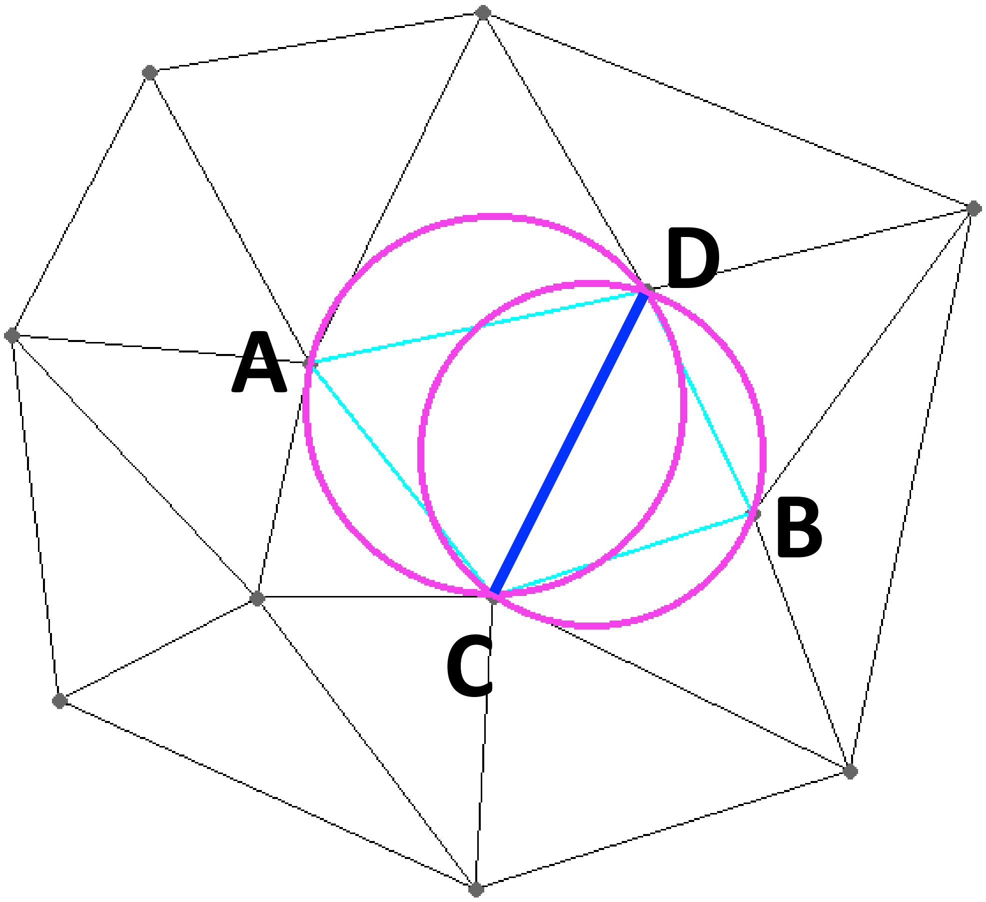
Edge flip



Edge flip



AB is not locally Delaunay

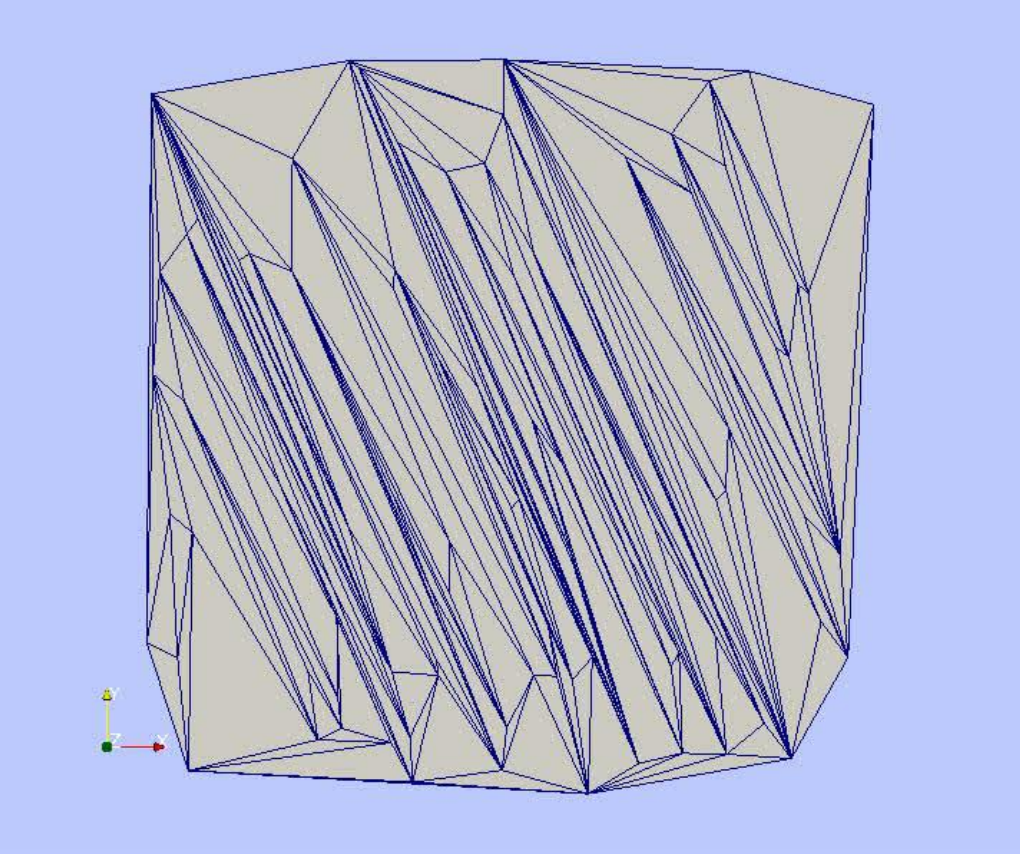


CD is locally Delaunay

Lawson's Flip Algorithm [1977]

- Let $S = \{p_1, p_2, \ldots, p_n\}$ be a finite set of points in $\mathbb{R}^2$.
- Compute an initial triangulation $\mathcal{T}$ of a point set S.

```
while ∃ a locally non-Delaunay edge ab ∈ T
    flip ab;
end while
```

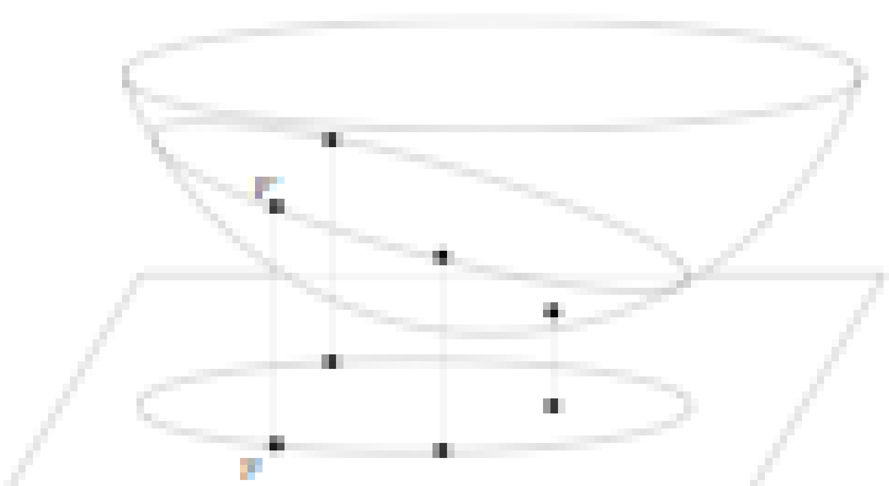


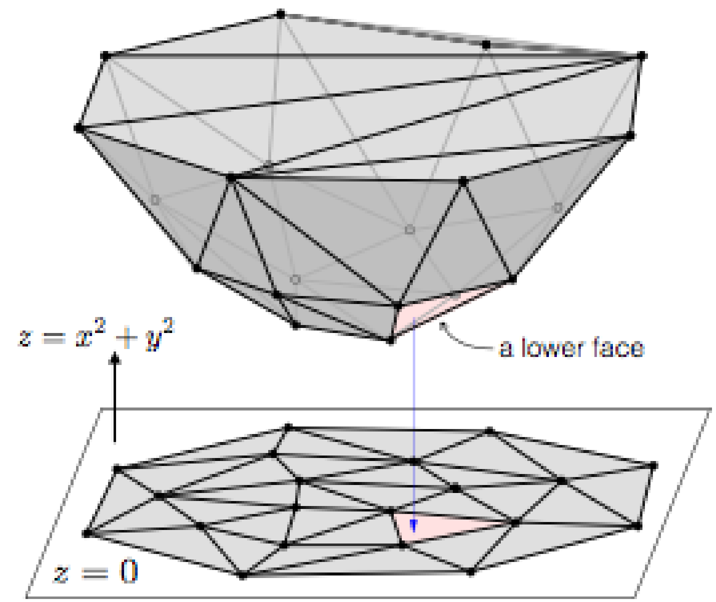
Convex hulls and Delaunay triangulations

- 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

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

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

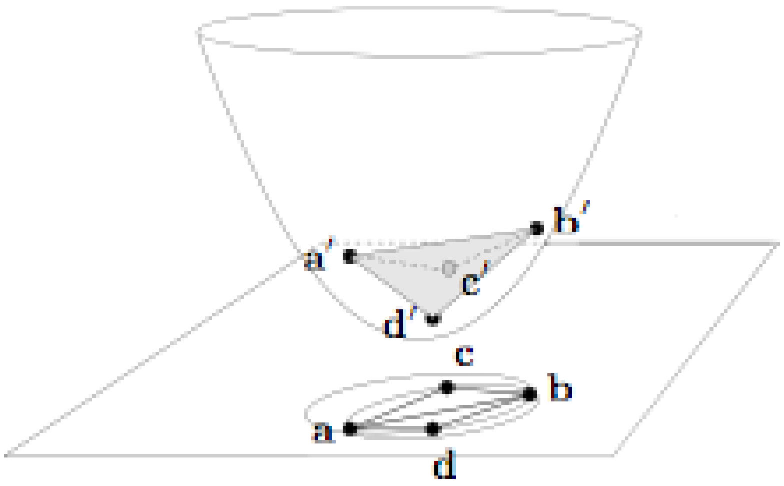


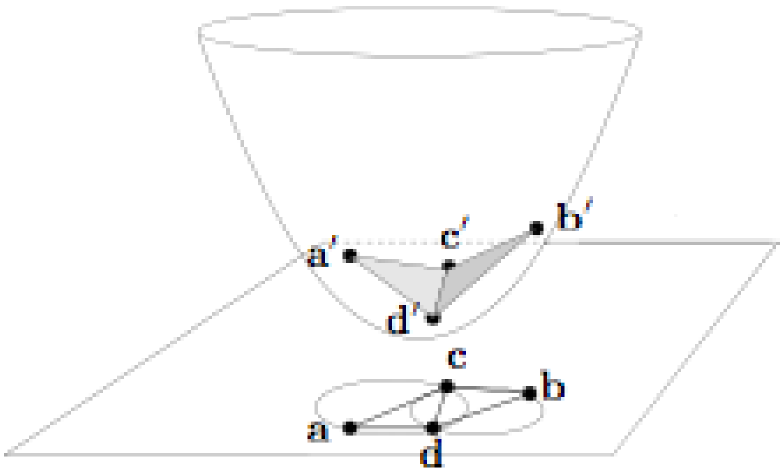


Lawson's Flip Algorithm [1977]

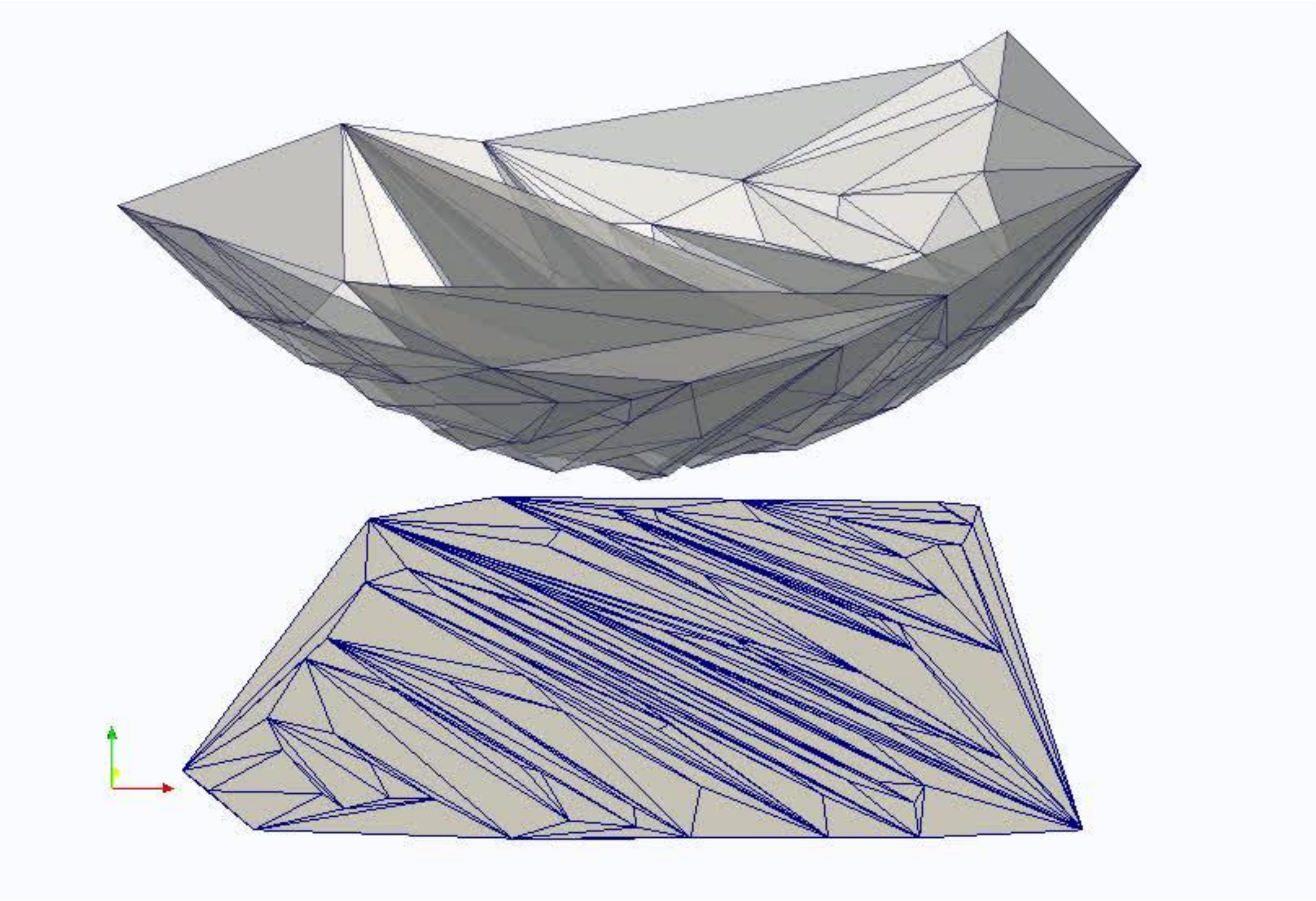
- Let $S = \{p_1, p_2, ..., p_n\}$ be a finite set of points in $\mathbb{R}^2$.
- Compute an initial triangulation $\mathcal{T}$ of a point set S.

```
while ∃ a locally non-Delaunay edge ab ∈ T
    flip ab;
end while
```



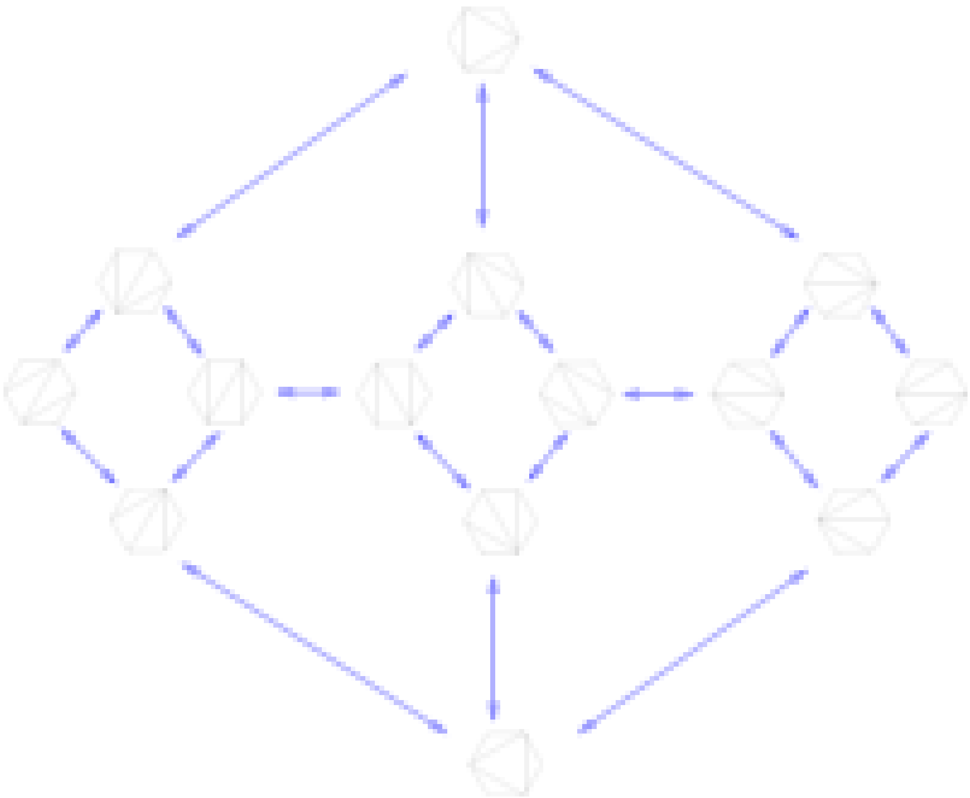


Lawson's Flip Algorithm [1972, 1977]



The flip graph

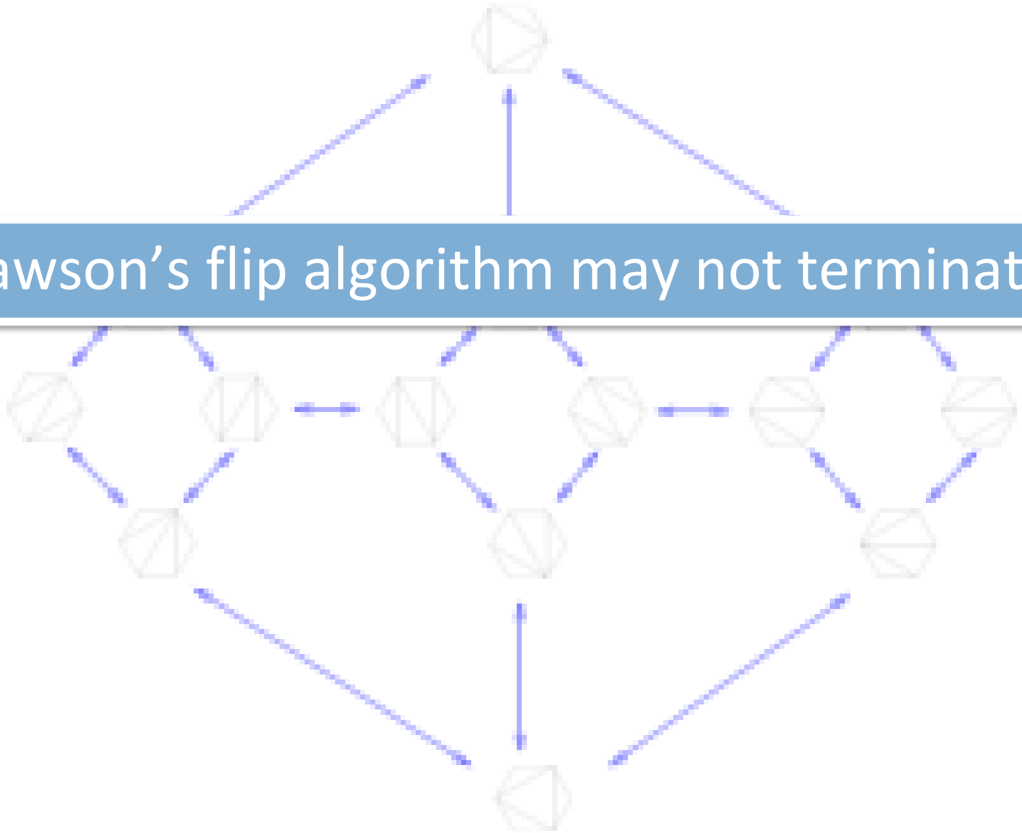
- All triangulations of the same point set can be transformed into each other by a sequence of edge flips – The flip graph of any point set in $\mathbb{R}^2$ is connected.



Example: The Flip Graph of the hexagon

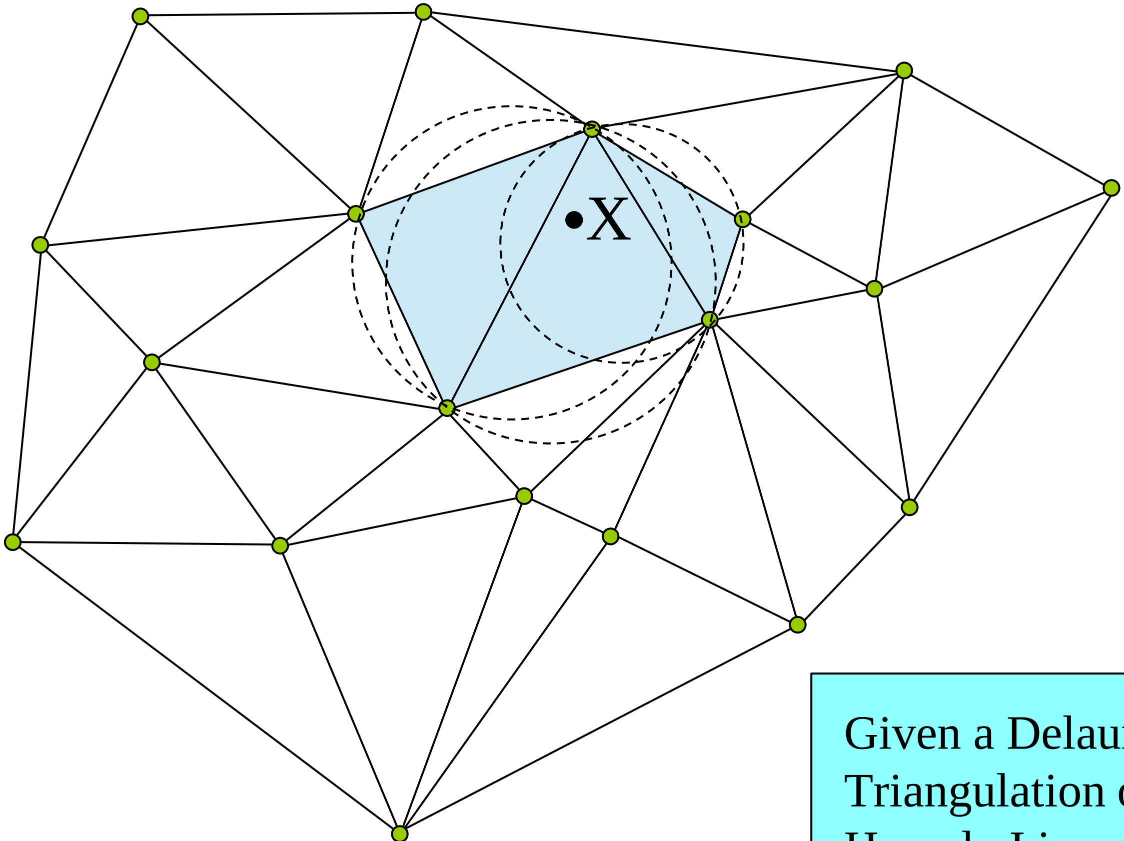
The Flip Graph

- All triangulations of the same point set can be transformed into each other by a sequence of edge flips – The flip graph of any point set in $\mathbb{R}^2$ is connected.



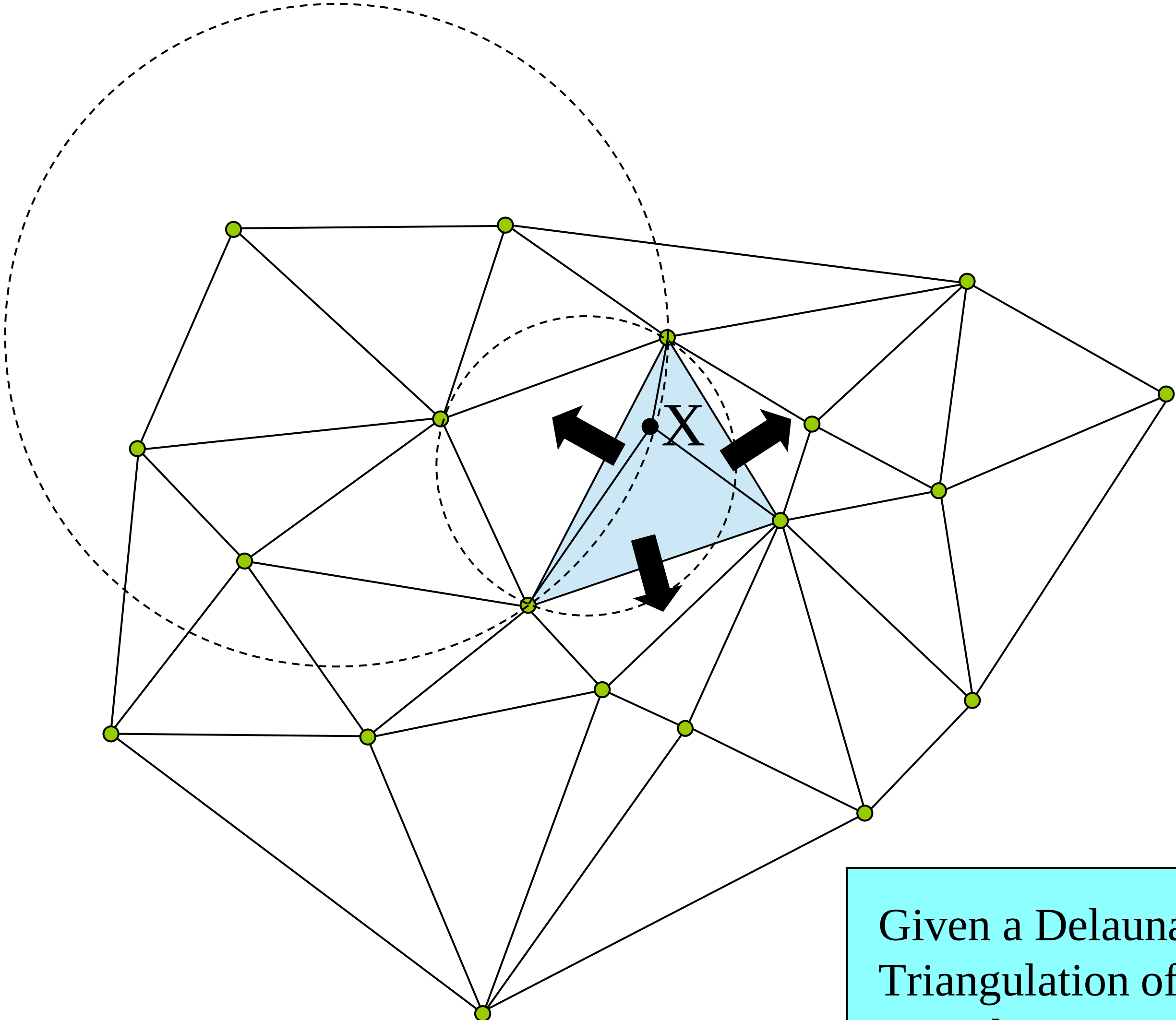
However, Lawson's flip algorithm may not terminate in 3d.

Incremental Flipping

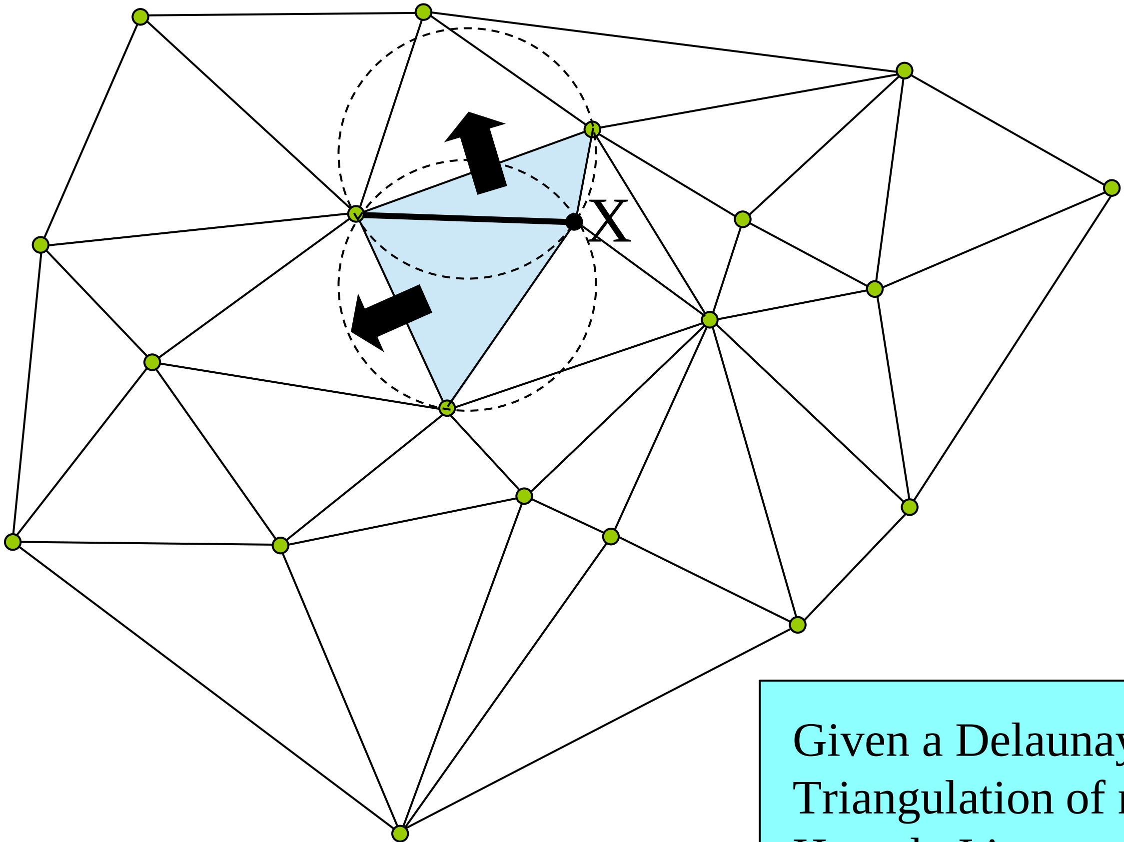


Given a Delaunay Triangulation of n nodes, How do I insert node n+1 ?

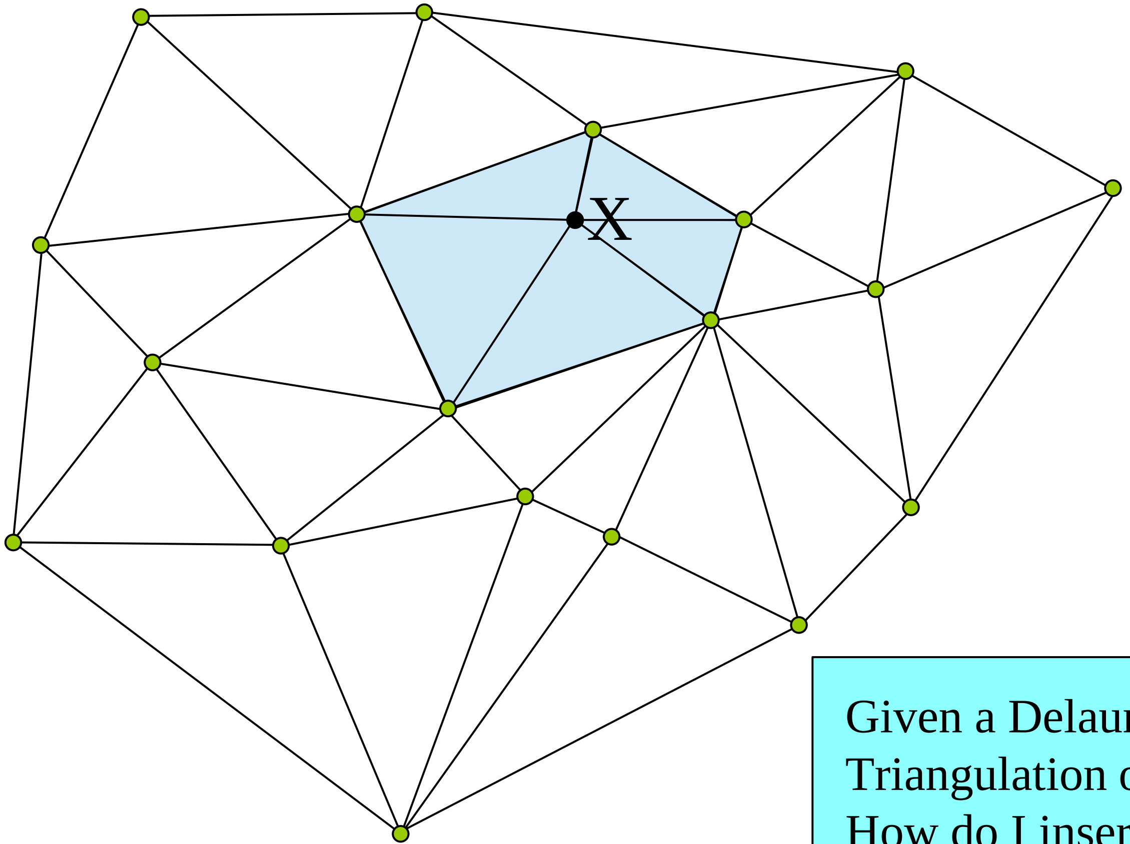
Incremental Flipping



Given a Delaunay Triangulation of n nodes,
How do I insert node n+1 ?



Given a Delaunay
Triangulation of n nodes,
How do I insert node n+1 ?

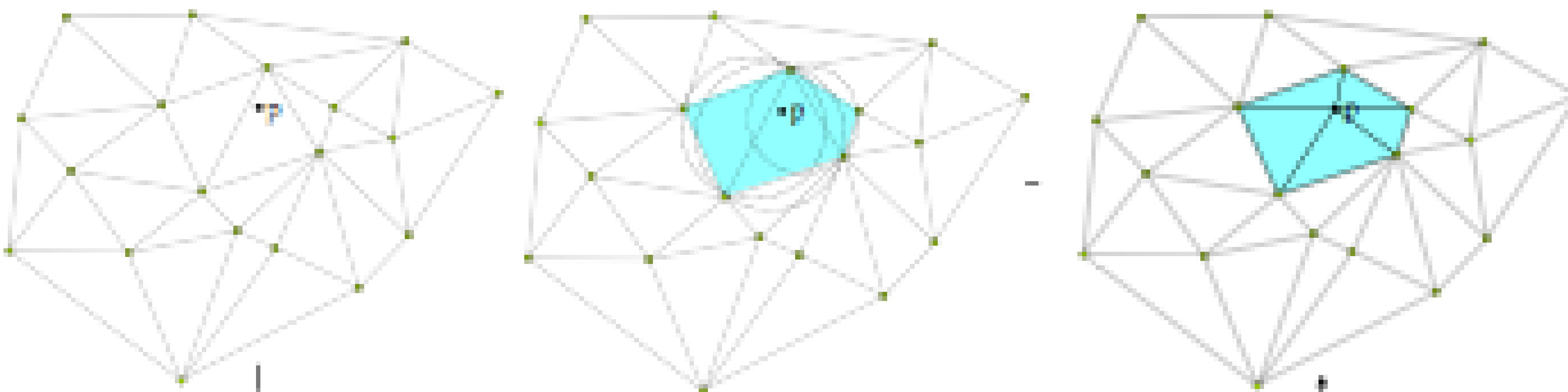


Given a Delaunay
Triangulation of n nodes,
How do I insert node n+1 ?

Incremental Flip

- 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 $\mathcal{D}_0$ consists of only the tetrahedron $[\mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}]$;
for $i = 1$ **to** n **do**
    find $[\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s}] \in \mathcal{D}_i$ that contains $\mathbf{p}_i$;
    add $\mathbf{p}_i$ with a 1-to-4 flip;
    while $\exists$ triangle $[\mathbf{a}, \mathbf{b}, \mathbf{c}]$ not locally Delaunay;
        flip $[\mathbf{a}, \mathbf{b}, \mathbf{c}]$;
    endwhile
endfor

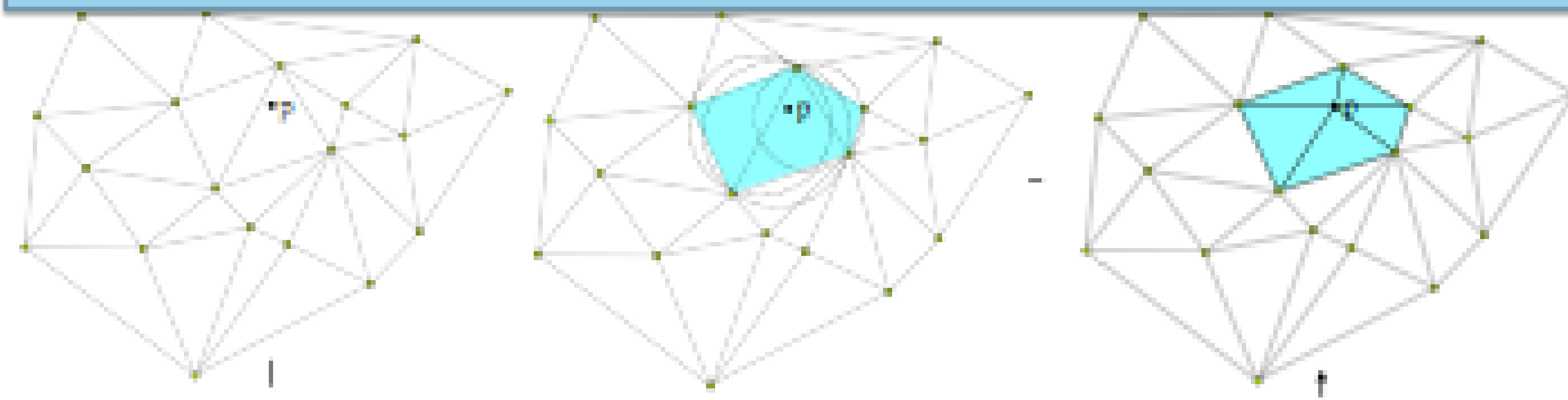


Incremental Flip [Edelsbrunner & Shah 1996]

- Let $S = \{p_1, p_2, \ldots, p_n\}$ be a finite set of points in $\mathbb{R}^3$.
- Let $[w, x, y, z]$ be a sufficiently large tetrahedron that contains all points of S.

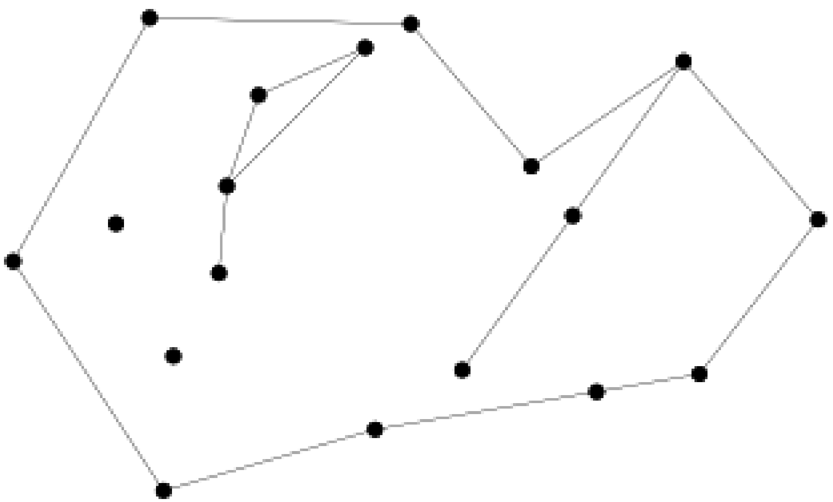
Let $\mathcal{D}_0$ consists of only the tetrahedron $[w, x, y, z]$;
for $i = 1$ to n do
    find $[p, q, r, s] \in \mathcal{D}_i$ that contains p_i;
    add p_i with a 1-to-4 flip;

Edelsbrunner, H. & Shah, N. R. Incremental topological flipping works for

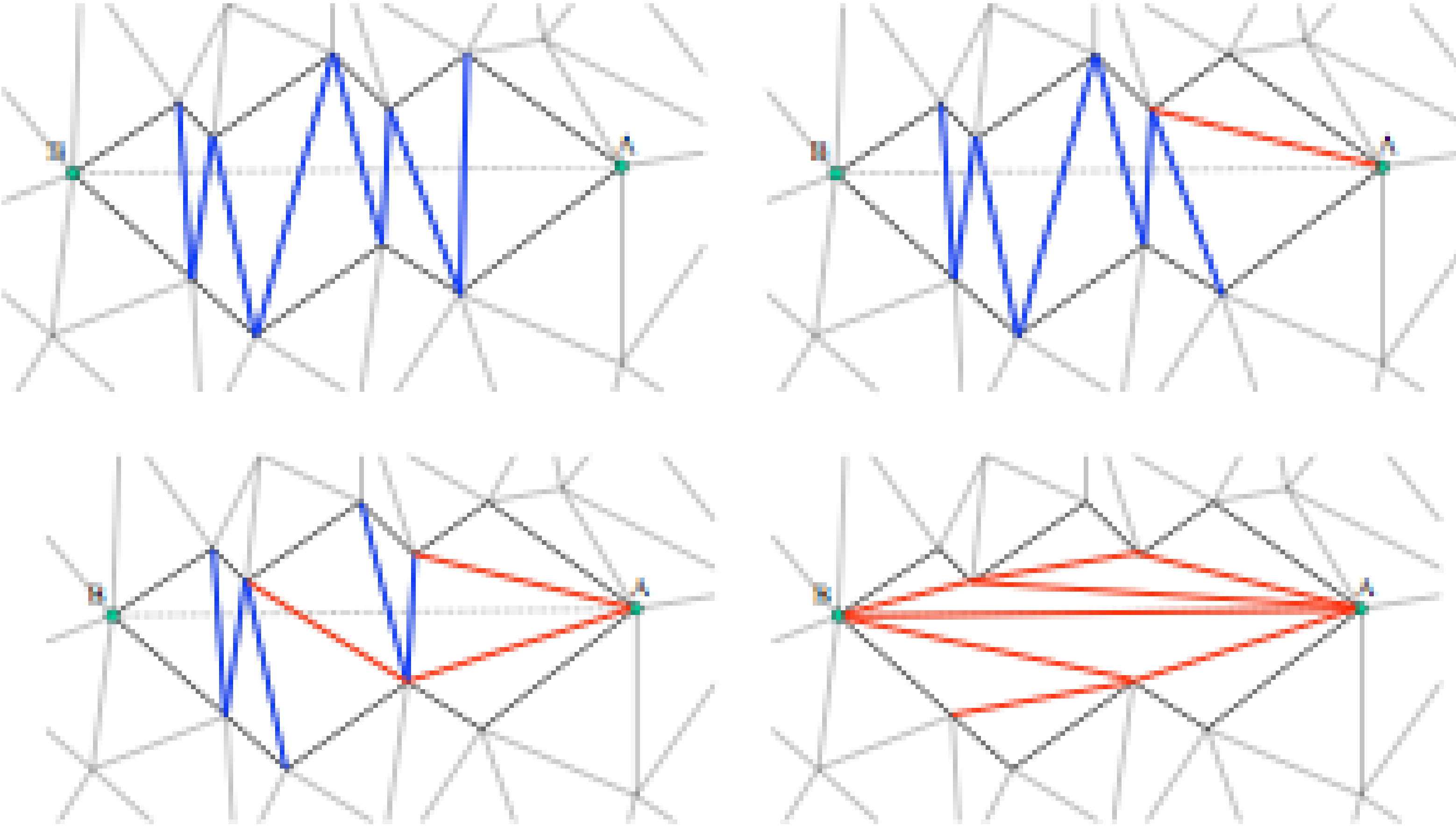


Constrained Triangulations

Meshing a Planar Straight Line Grap

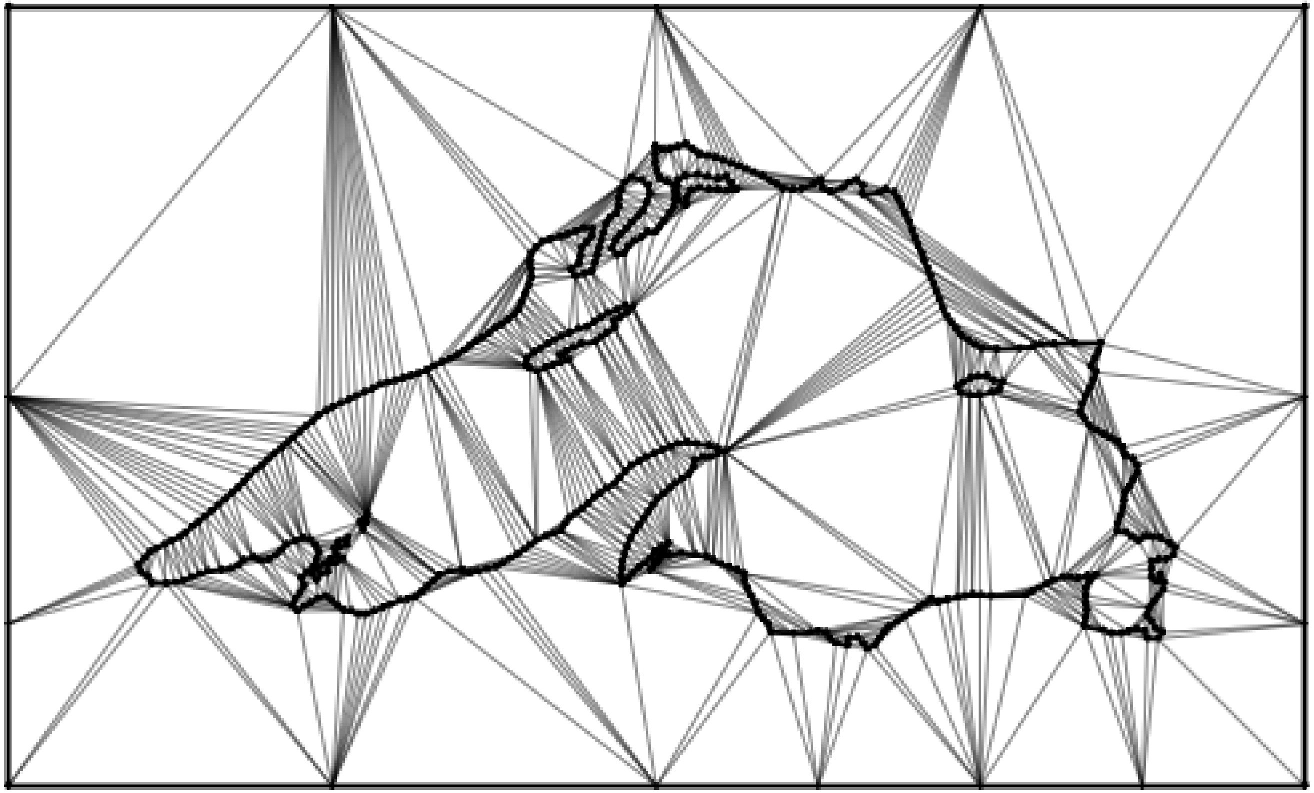


Constrained triangulations



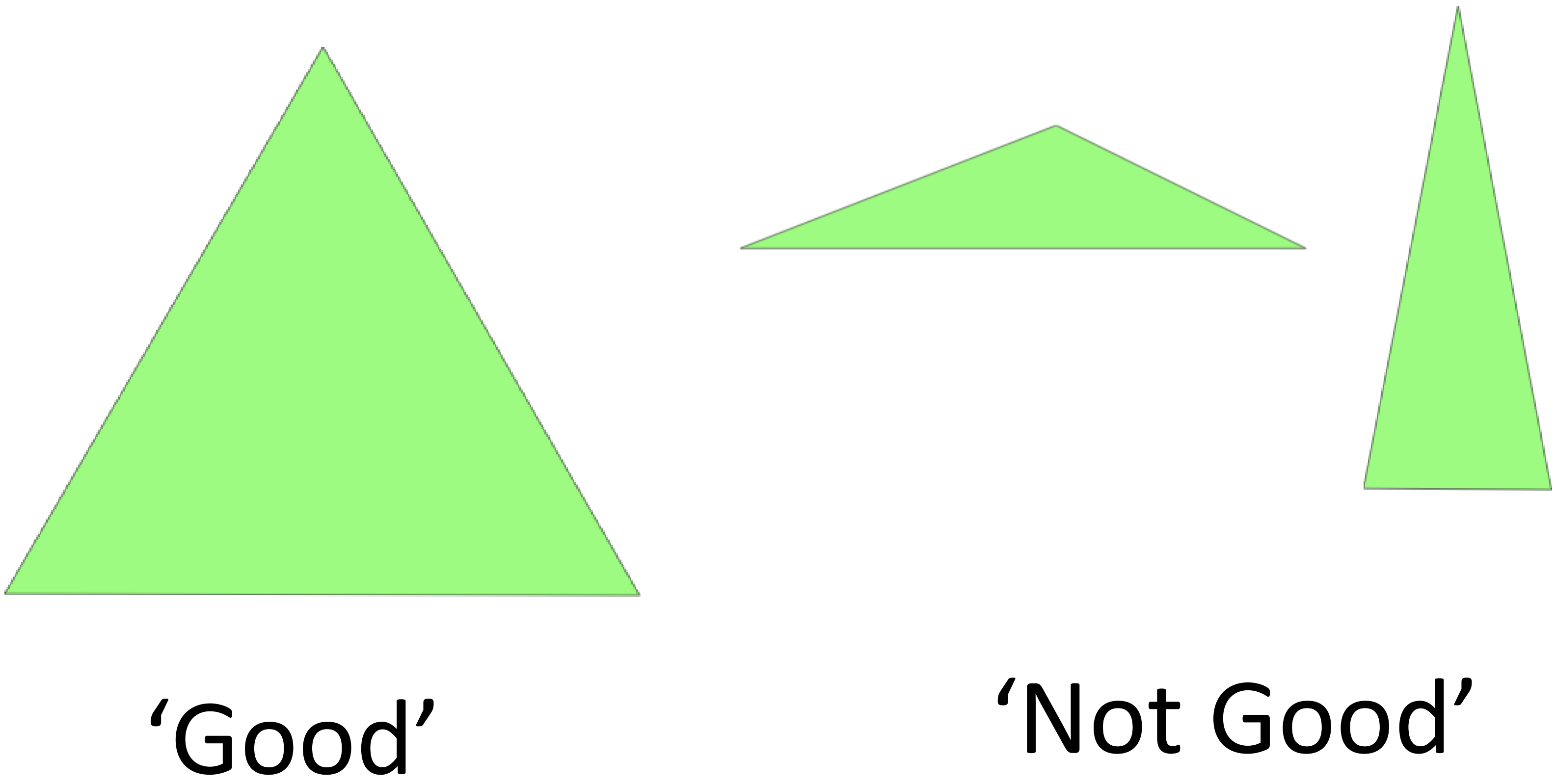
Constrained triangulations

Constrained triangulations



Quality Mesh Generation

Quality of triangles



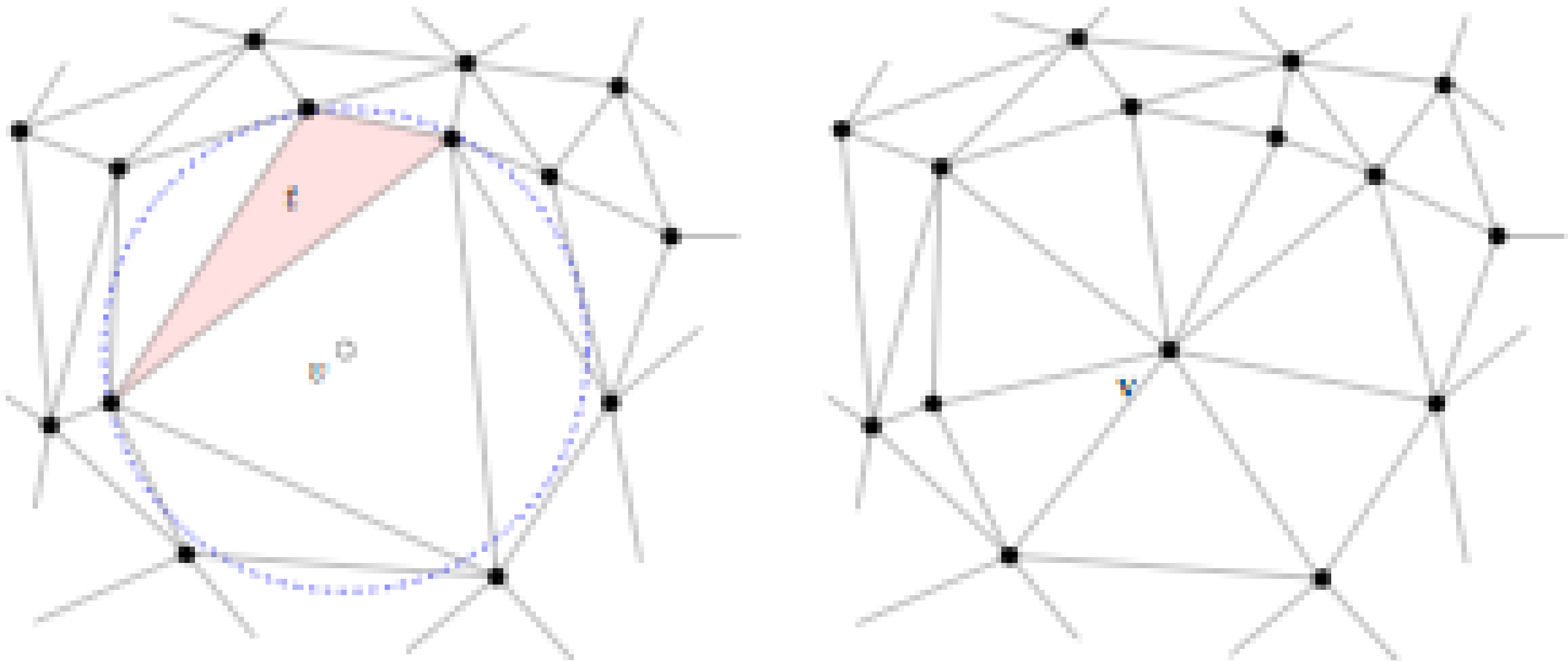
Quality of triangles

- minimal angle $\min_{\theta \text{ of } \tau} \theta$
- mean ratio $\sum_k d_k^2 / |\tau|^{2/n}$ (or its reciprocal)
- aspect/radius ratio *inradius / circumradius*

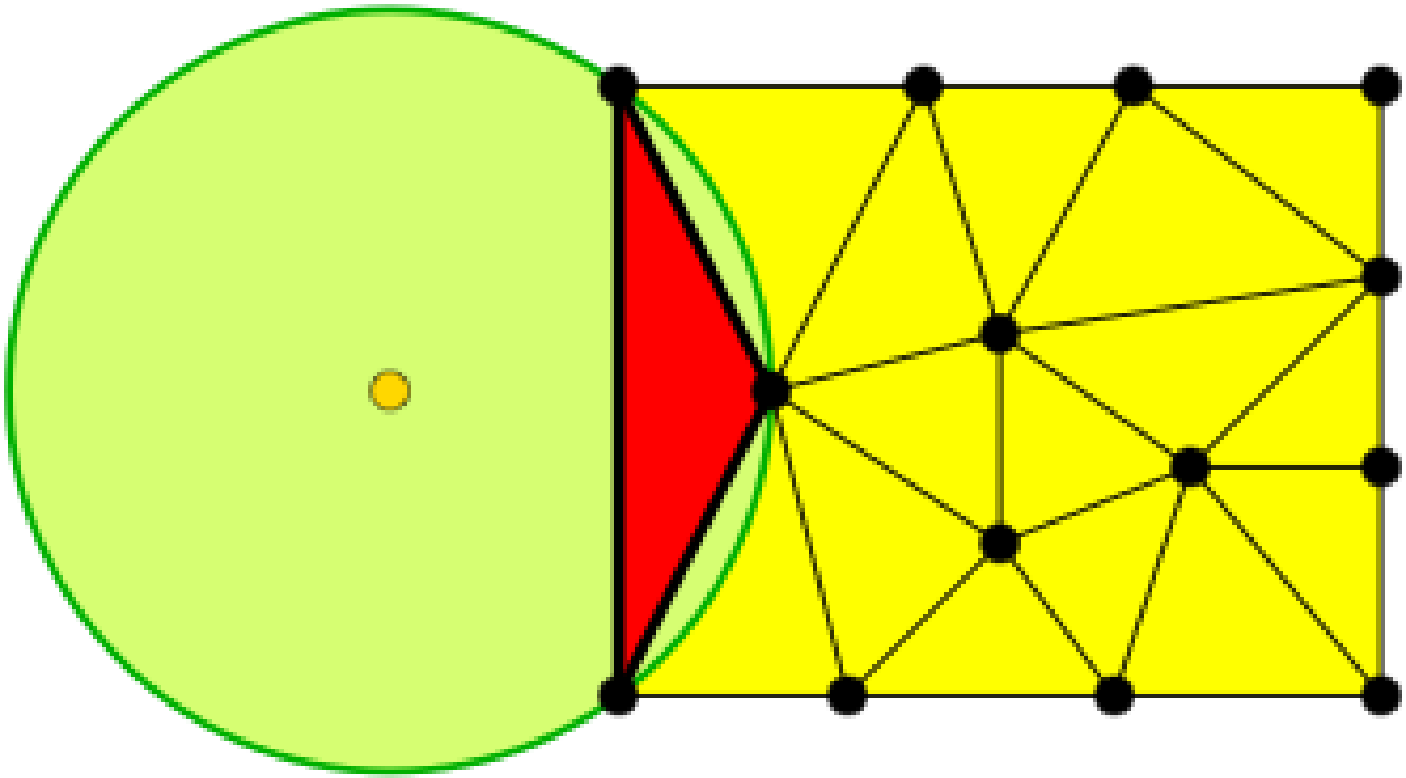
There are lots of geometric qualities ... .

Delaunay refinement [Chew 1989, Ruppert 1995]

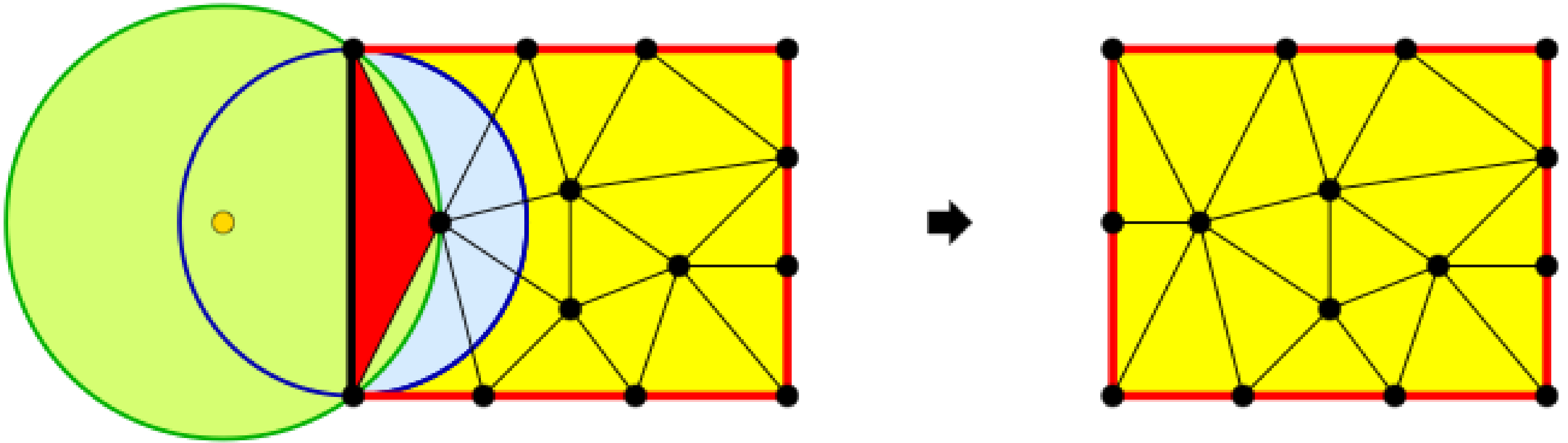
- Kill bad elements by insertion of their circumcenter.
- Bad elements: badly-shaped, oversized, etc.



Circumcenter may lie outside of the domain

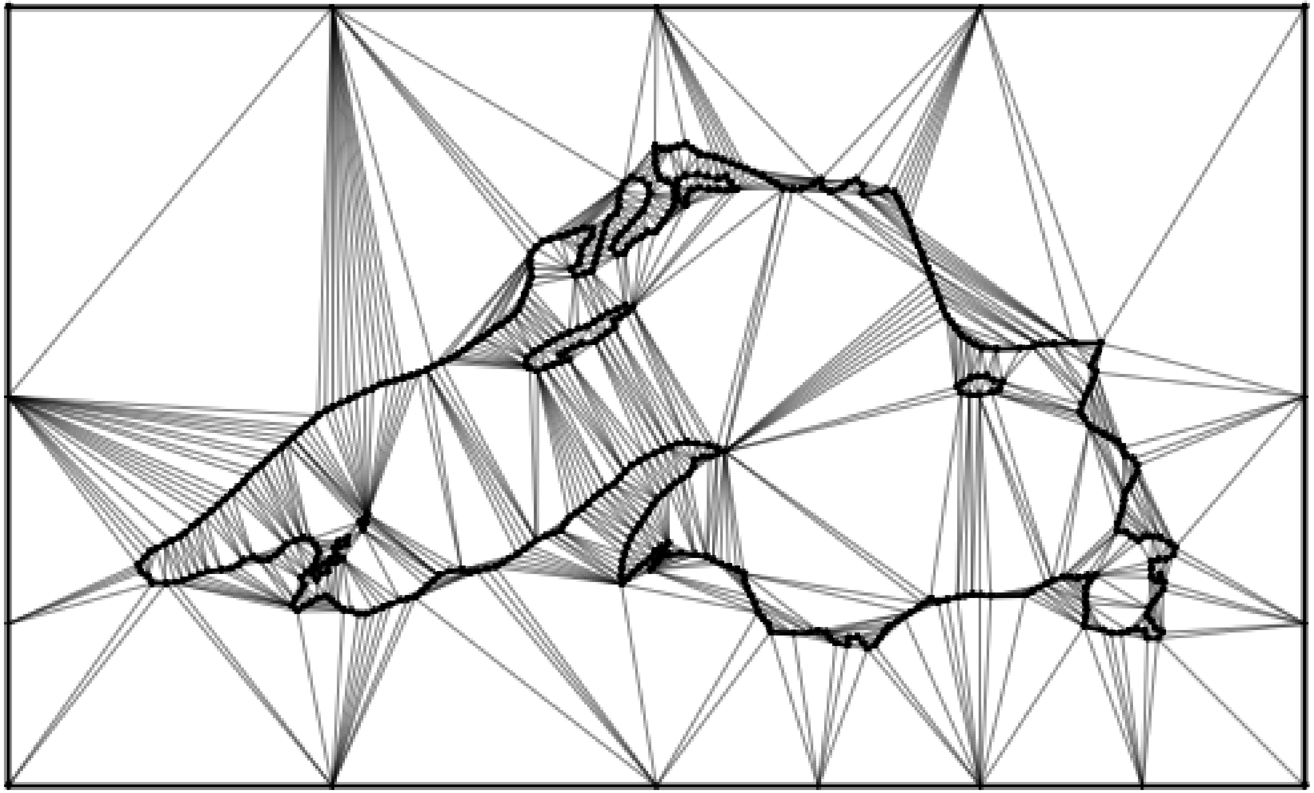


Boundary protection

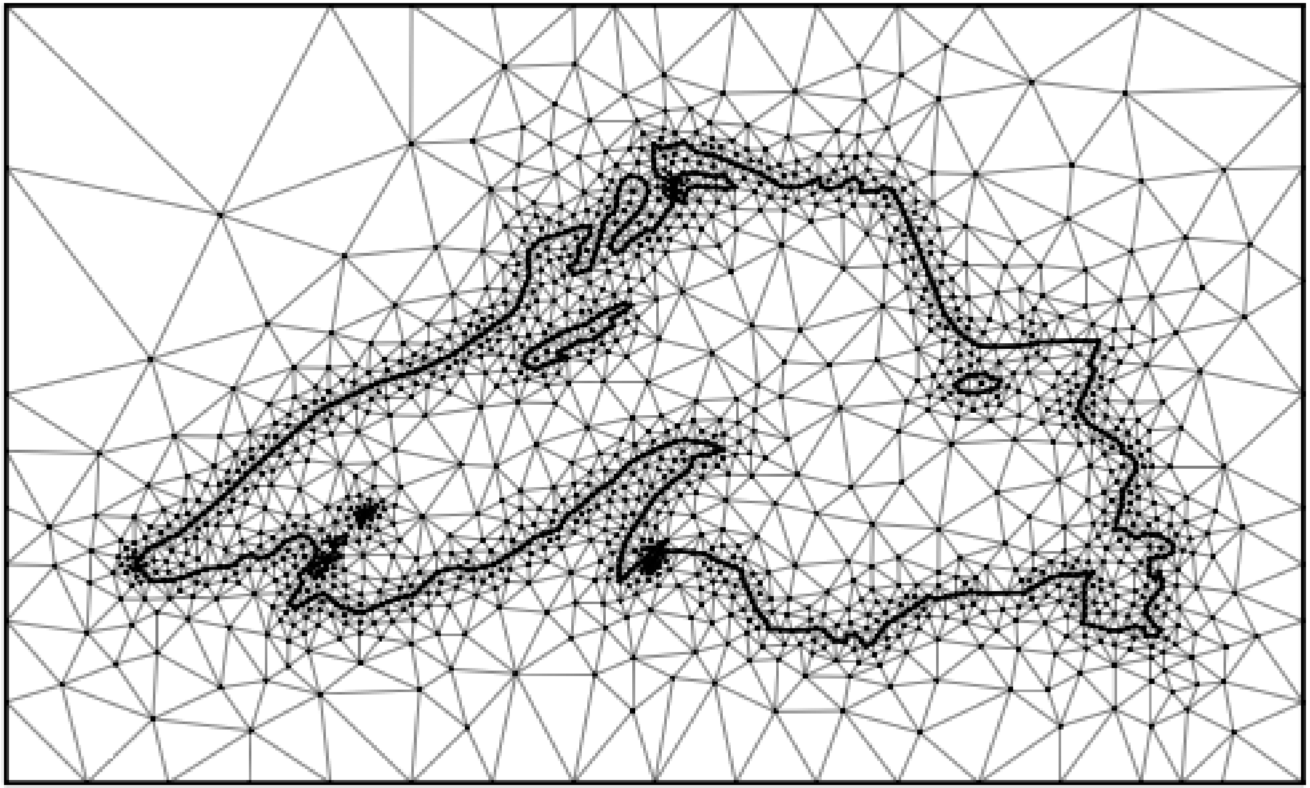


Split segments if its diametral circumcircle is not empty

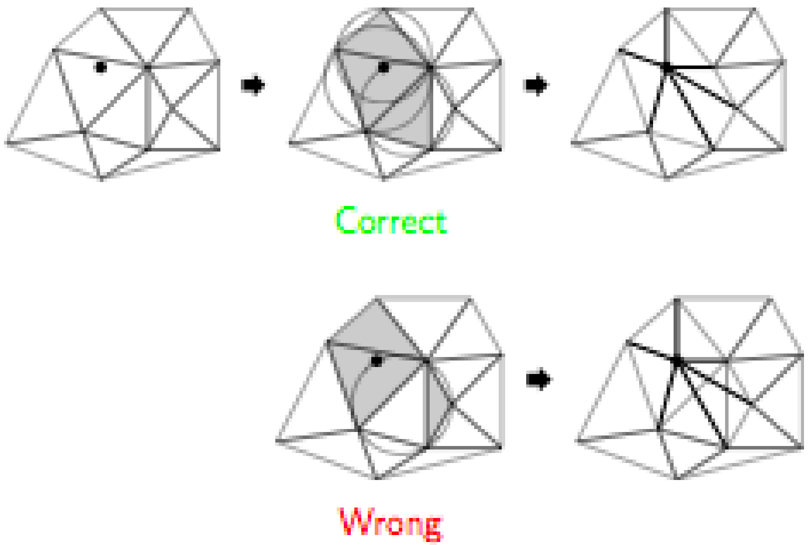
An input constrained triangulation



A result of Delaunay refinement

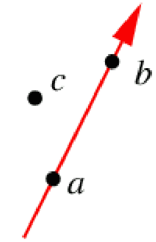


Robustness

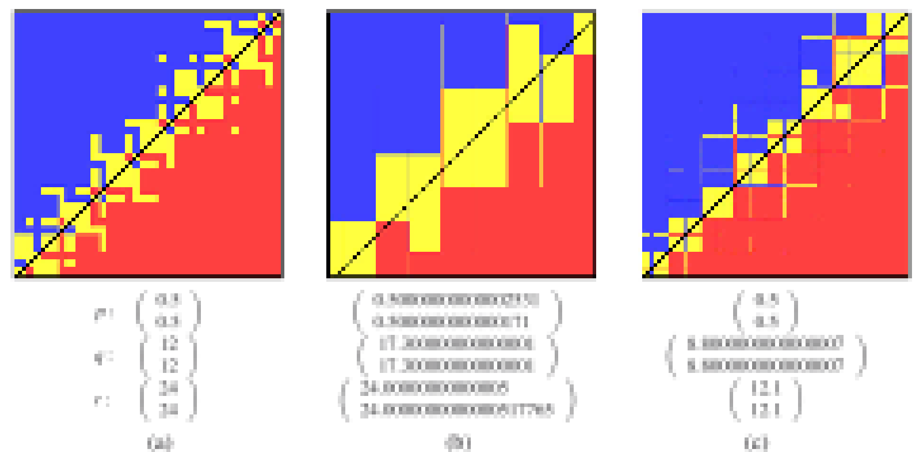


Orientation

Does c lie on, to the left of, or to the right of $\vec{ab}$?



$$\begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix} = \begin{vmatrix} a_x - c_x & a_y - c_y \\ b_x - c_x & b_y - c_y \end{vmatrix}$$



256 x 256 pixel image

> 0, $= 0$, < 0

[Kettner et al. 2008].

Filtered Exact Predicates

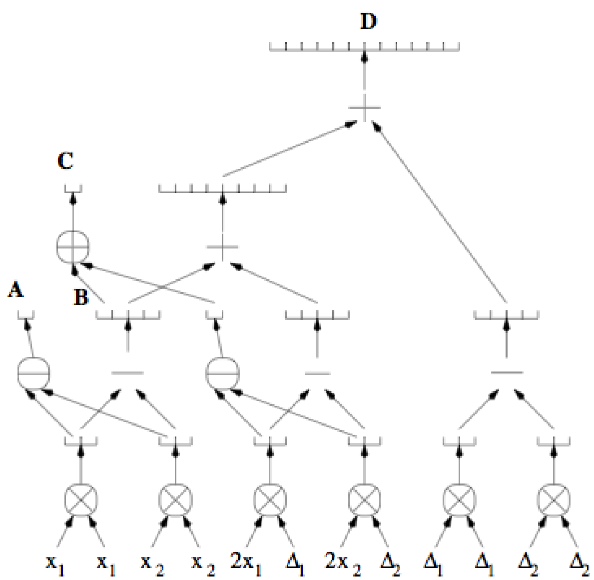
"filters out" the easy cases

let F = E (X) in **floating point**
  if F > **error bound** then 1 else
    if –F > error bound then –1 else
      increase precision and repeat
      or switch to exact arithmetic

Code with static filtering (for entries bounded by 1):

```
int filtered_orientation(double px, double py,
                         double qx, double qy,
                         double rx, double ry)
{
  double pqx = qx - px,  pqy = qy - py;
  double prx = rx - px,  pry = ry - py;

  double det = pqx * pry - pqy * prx;

  const double E = 1.33292e-15;

  if (det >  E)  return  1;
  if (det < -E)  return -1;

  ... // can't decide => call the exact version
}
```

CGAL



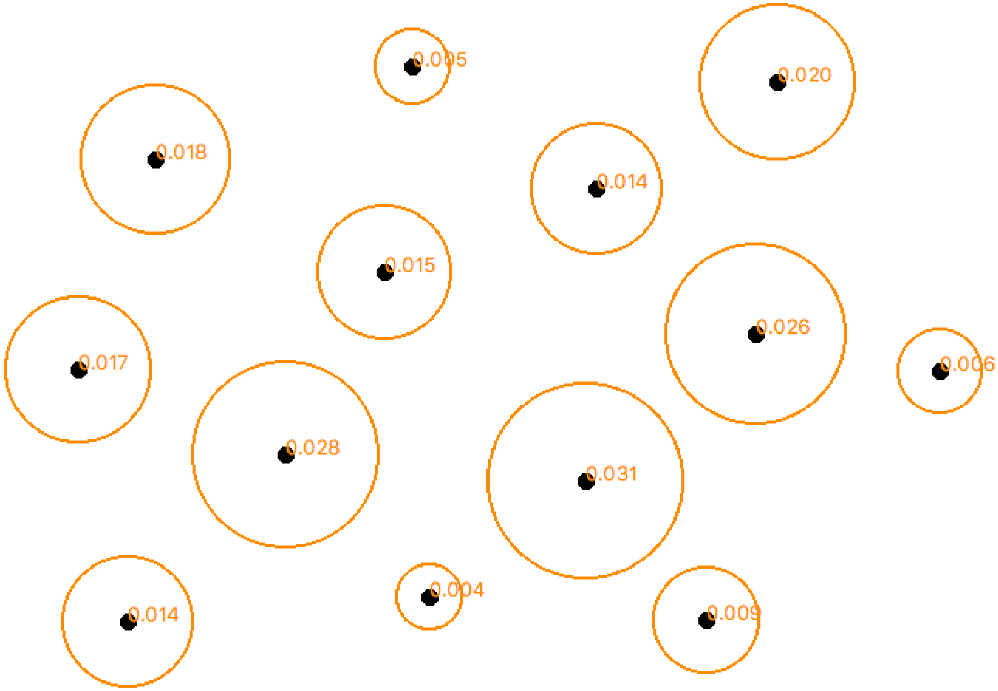
Shewchuk's adaptive predicates

Weighted Delaunay Triangulations

Weighted Points

- Let S be a finite set of points in $\mathbb{R}^d$, and assign a real valued weight w_p to each point $\mathbf{p} \in S$ to obtain a weighted point $\hat{\mathbf{p}} = (p_1, p_2, \cdots, p_n, p_{n+1})$, where

$$\begin{aligned} p_{n+1} &= \|\mathbf{p}\|^2 - w_p \\ &= (p_1^2 + p_2^2 + \cdots + p_n^2) - \underbrace{w_p}_{\text{weight}} \end{aligned}$$



Weighted Distances

- The weighted distance between two weighted points $\hat{\mathbf{p}}$ and $\hat{\mathbf{z}}$ is

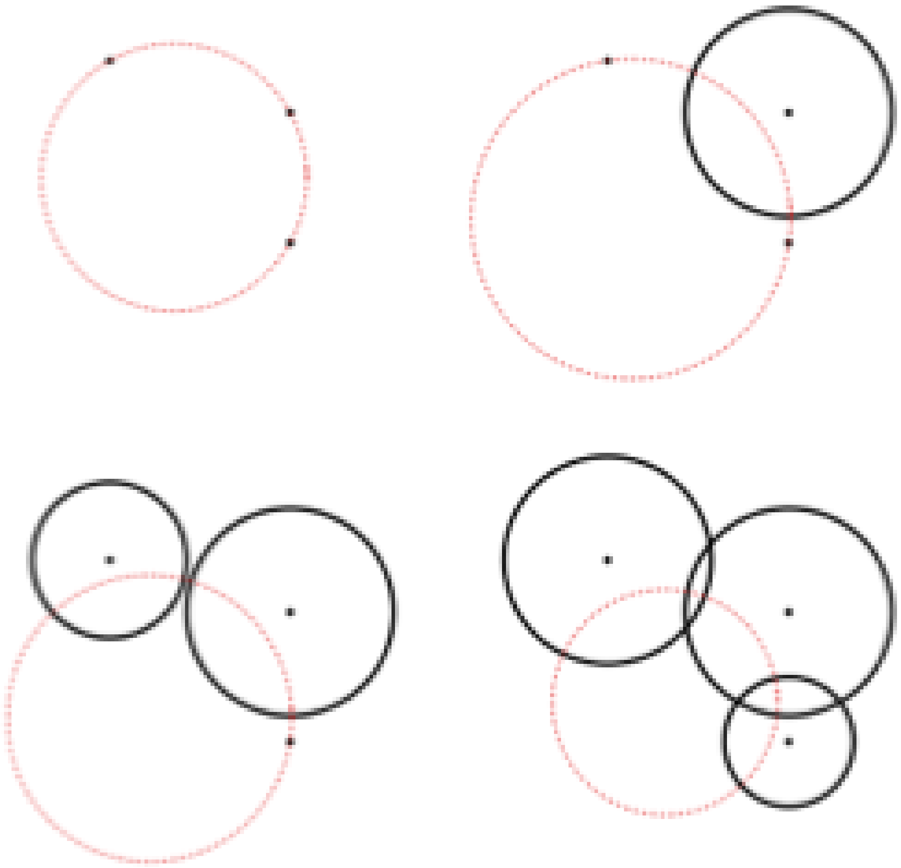
$$\pi_{\hat{p},\hat{z}} = \|\mathbf{p} - \mathbf{z}\|^2 - w_p - w_z.$$

- Two weighted points $\hat{\mathbf{p}}$ and $\hat{\mathbf{z}}$ is orthogonal to each other if their weighted distance is zero, i.e.,

$$\|\mathbf{p} - \mathbf{z}\|^2 = w_p + w_z.$$

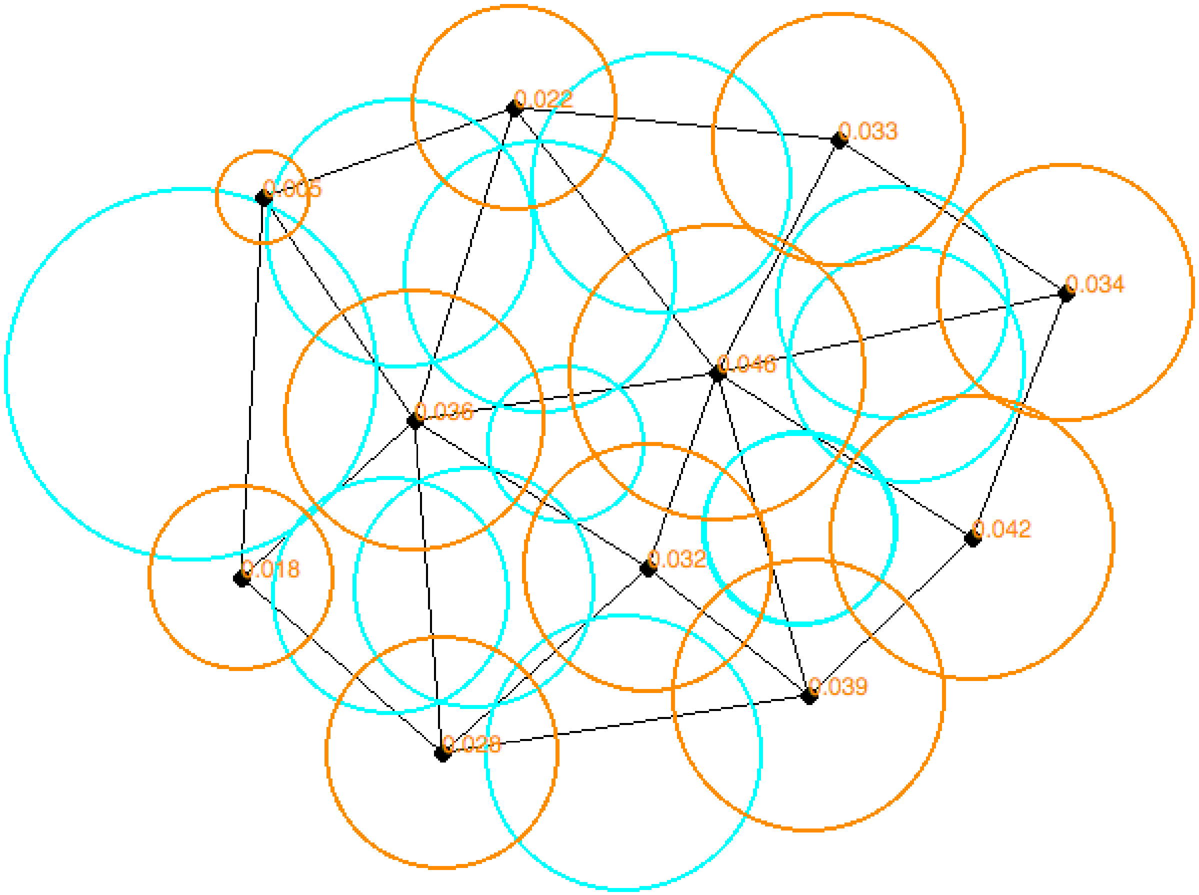
Orthocircles

- Let $\hat{S} \subset \mathbb{R}^d \times \mathbb{R}$ be a finite set of weighted points. $d+1$ weighted points define a unique common orthosphere which is orthogonal to them.

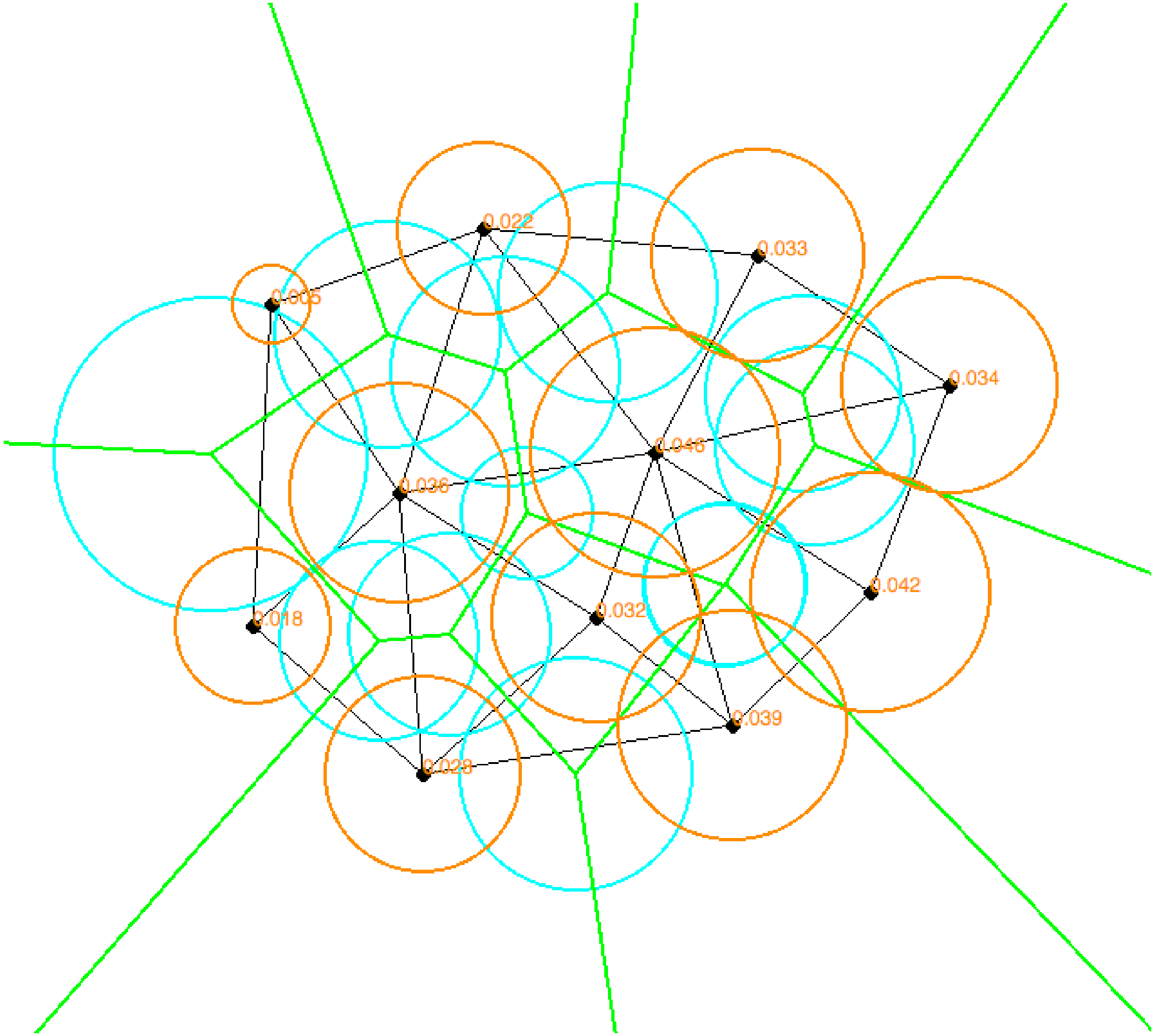


Weighted Delaunay Triangulations

- The weighted Delaunay triangulation of $\hat{S}$ consists of simplices with vertices in $\hat{S}$ such that their orthosphere is empty.



Power (weighted Voronoi) diagrams

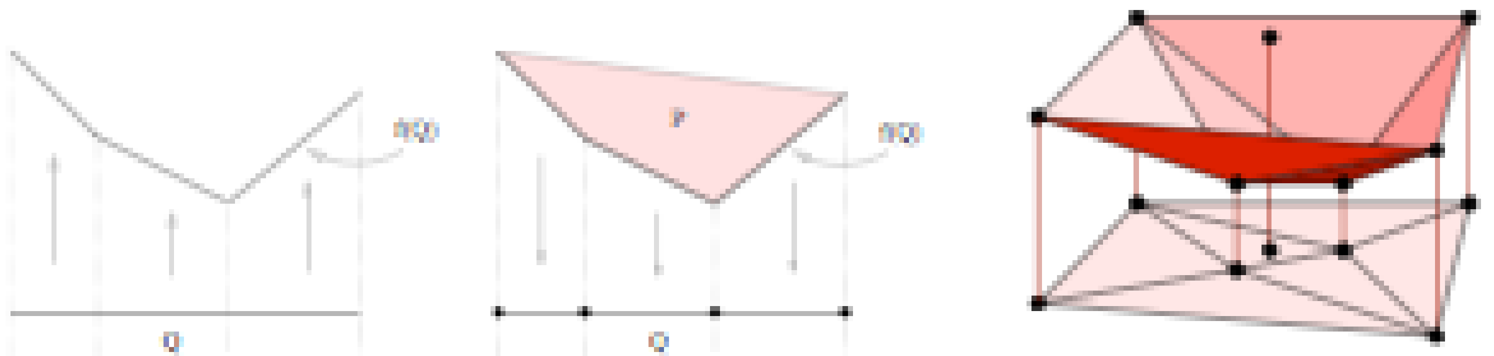


Regular Triangulations

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

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

  The orthogonal projection of the lower envelope of P determines a regular subdivision of Q.
- A particular choice for f is the function $f(\mathbf{x}_i) = ||\mathbf{x}_i||^2$. The obtained regular subdivision is called the Delaunay triangulation of S [Delaunay 1934].

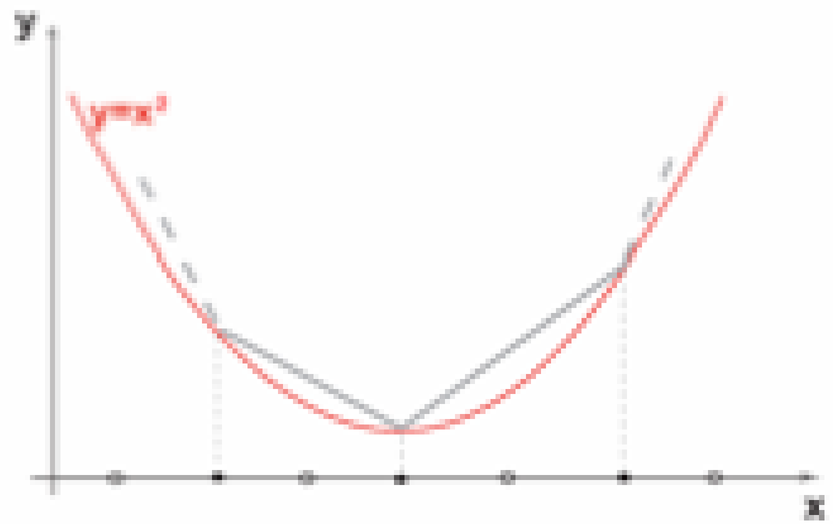


Courtesy of Jörg Rambau

- **Theorem [Chen and Xu 2004]**: Denote $Q(\mathcal{T}, f, p) = \|f - f_{I,\mathcal{T}}\|_{L^p}$, where $f_{I,\mathcal{T}}$ is the linear interpolation of f based on the triangulation $\mathcal{T}$ of a point set $S \subset \mathbb{R}^d$.
  If f is convex, then

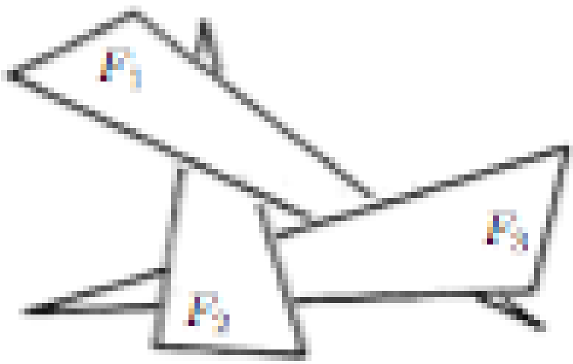
$$Q(\mathcal{R}, f, p) := \min\{Q(\mathcal{T}, f, p) \; : \; \mathcal{T} \in \mathcal{P}_S\}, \; 1 \le p < \infty,$$

  where $\mathcal{R}$ is the regular subdivision of S.
- A Delaunay triangulation is the optimal triangulation for piecewise linear interpolation to the function $\|\mathbf{x}\|^2$ [Rippa 1992].

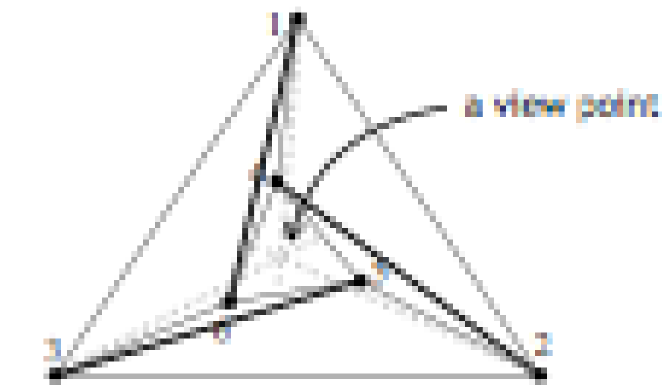


The Acyclic Theorem

- 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.



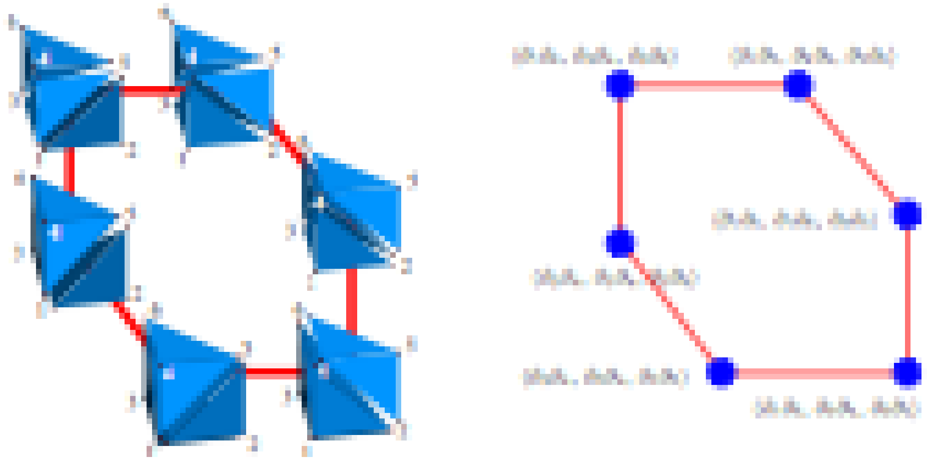
$F_1 \prec F_2 \prec F_3 \prec F_1$



$t_{1,4,6} \prec t_{3,5,6} \prec t_{2,4,5} \prec t_{1,4,6}$

The Flip Graph of Regular Triangulations

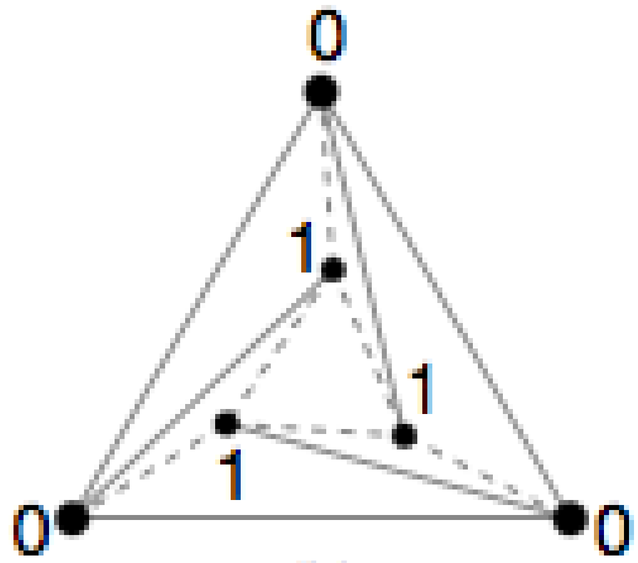
- In the incremental construction, it is assumed that an initial DT is given. Otherwise, there is no guarantee of termination.
- The flip graph of a point set S: each vertex represents a triangulation of S, each edge represents a flip between two triangulations of S.
  - The flip graph of any 2D point set is connected [**Lawson 1977**].
  - The flip graph of all regular triangulations is connected [**Gel'fand, Kapranov & Zelevinski 1990,1994**].
  - From $\mathbb{R}^5$, the flip graph can be not connected [**Santos 2000, 2005**].
  - The question is open in $\mathbb{R}^3$ and $\mathbb{R}^4$.



(Figures from J. Pfeifle's thesis, TU-Berlin, 2003).

Non-regular Triangulations

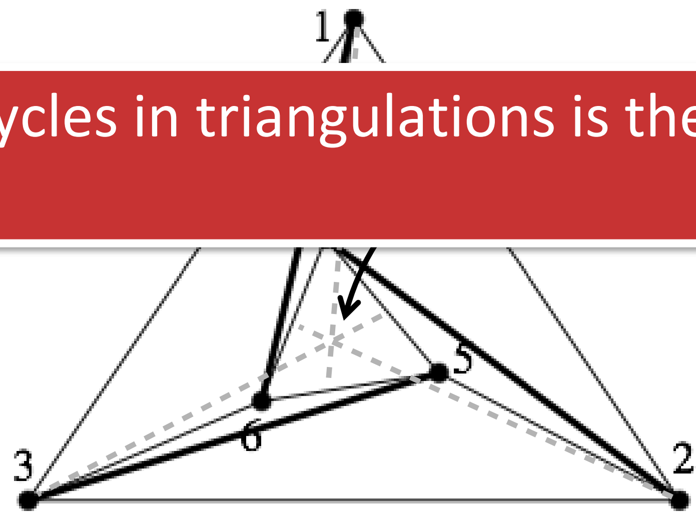
- A subdivision of a point set S is non-regular if it is not a regular subdivision of S.
- There are many non-regular subdivisions. For example, most triangulations of cyclic polytopes are non-regular [Rambau 1996].



A non-regular triangulation

Non-regular triangulations and Cycles

Unlike the regular triangulaitons (**Acyclic Theorem [Edelsbrunner 1990]**
cycles of simplices from a fixed view point.



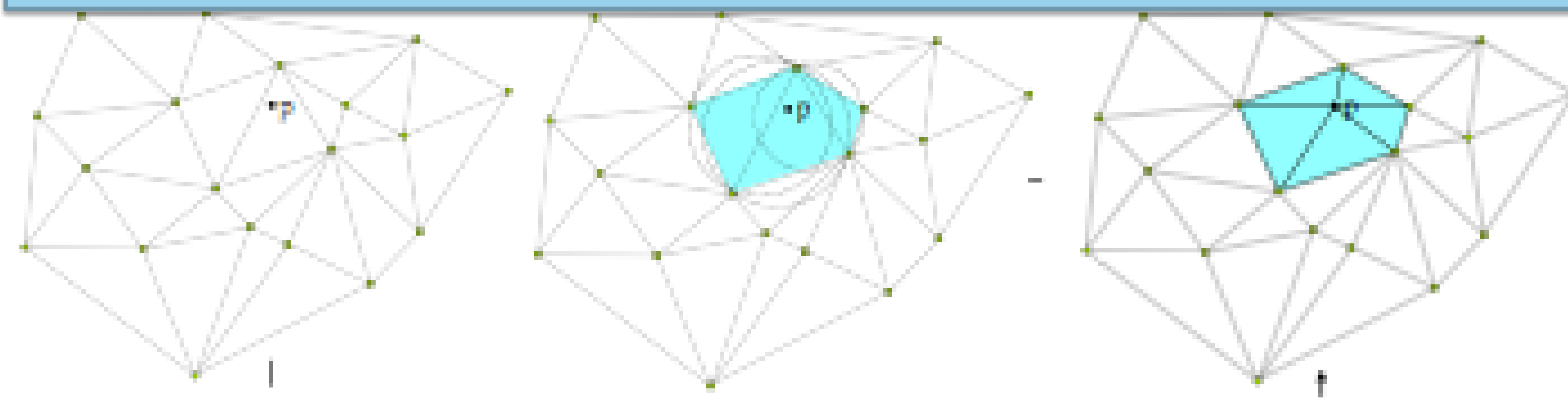
The existence of cycles in triangulations is the reason that causes the f

Incremental Flip [Edelsbrunner & Shah 1996]

- Let $S = \{p_1, p_2, ..., p_n\}$ be a finite set of points in $\mathbb{R}^3$.
- Let $[w, x, y, z]$ 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;
```

Edelsbrunner, H. & Shah, N. R. Incremental topological flipping works fo



Detri2

- Detri2 is a C++ program and library for generating weighted Delaunay triangulations as well as power Voronoi diagrams for weighted point sets in 2d.

- It generates boundary constrained Delaunay triangulations and good-quality triangular meshes for arbitrary polygonal domains in 2d.

- It generates (isotropic and anisotropic) adapted meshes from a user-specified sizing function.

http://www.wias-berlin.de/people/si/detri2.html

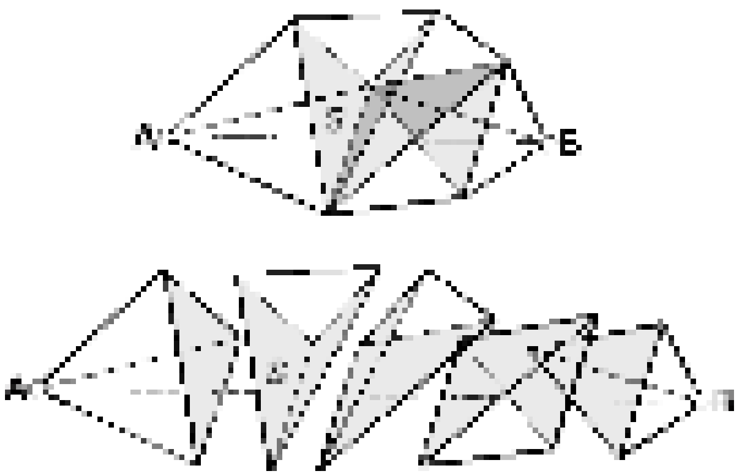
Outline

1. Introduction
2. Triangular Mesh Generation
3. Tetrahedral Mesh Generation
4. Mesh Adaptation
5. Further Topics

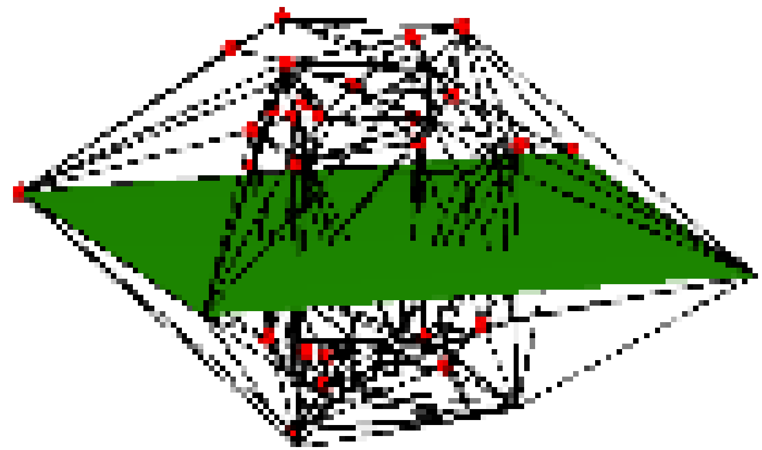
Constrained Triangulations in 3d

Triangulations with constraints

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



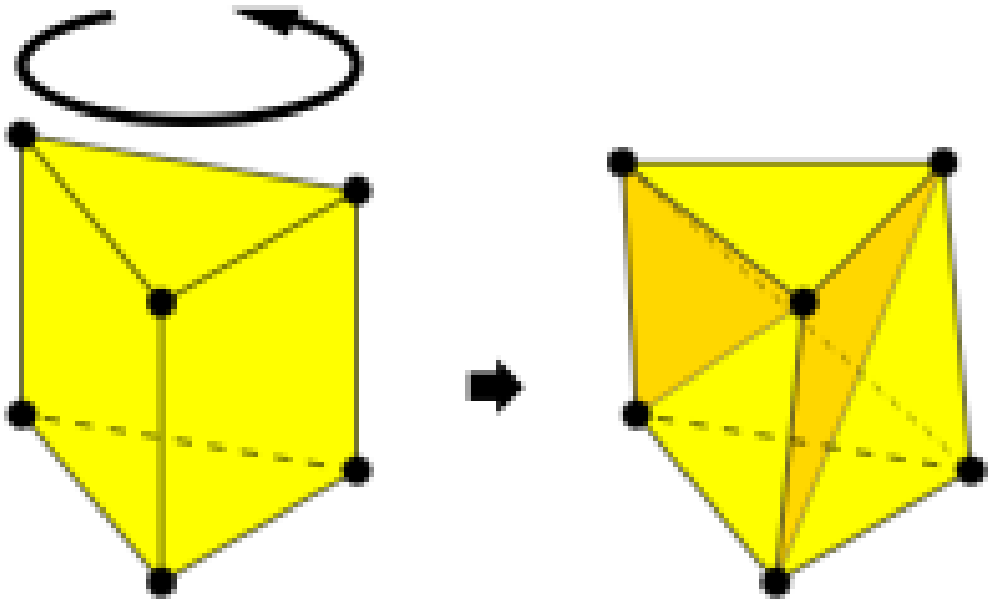
How to recover the edge AB?
Image from [Owen 1999]



How to recover the rectangular face

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].

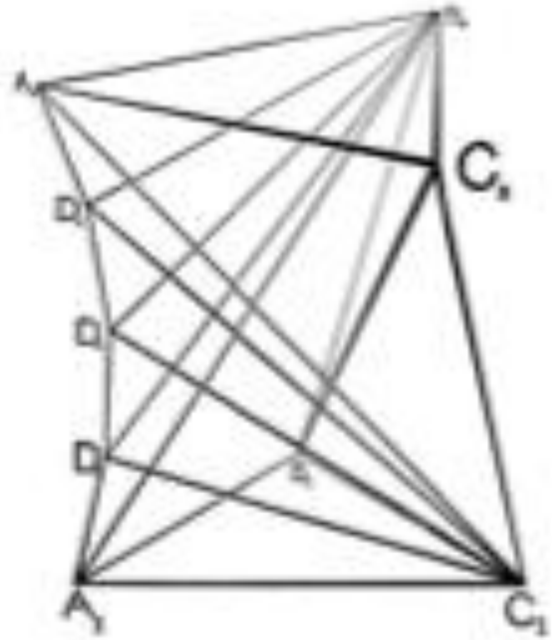


The Schönhardt Polyhedron [1928]

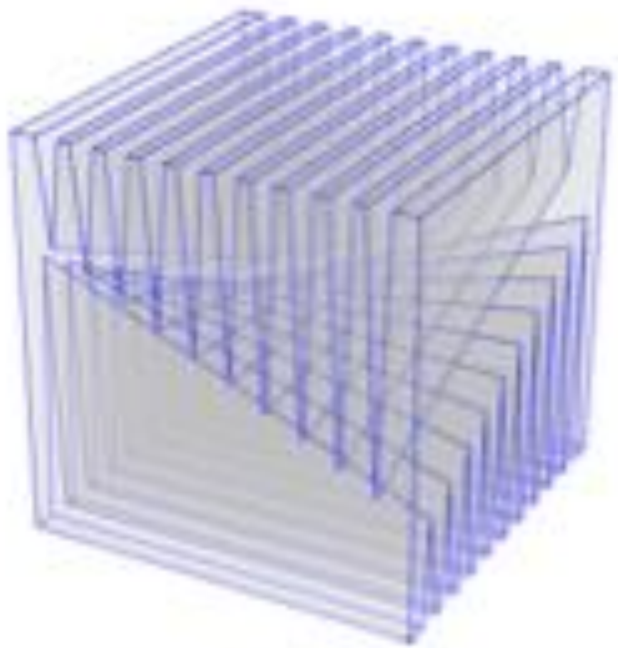
(1) Jakob Steiner (1796 – 1863), a Switzerland native and a geometer from Berlin.

Steiner Points

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



A Bagemihl's Polyhedron Π_5
(Π_5 = Schönhardt's polyhedron)
[Bagemihl 1948]

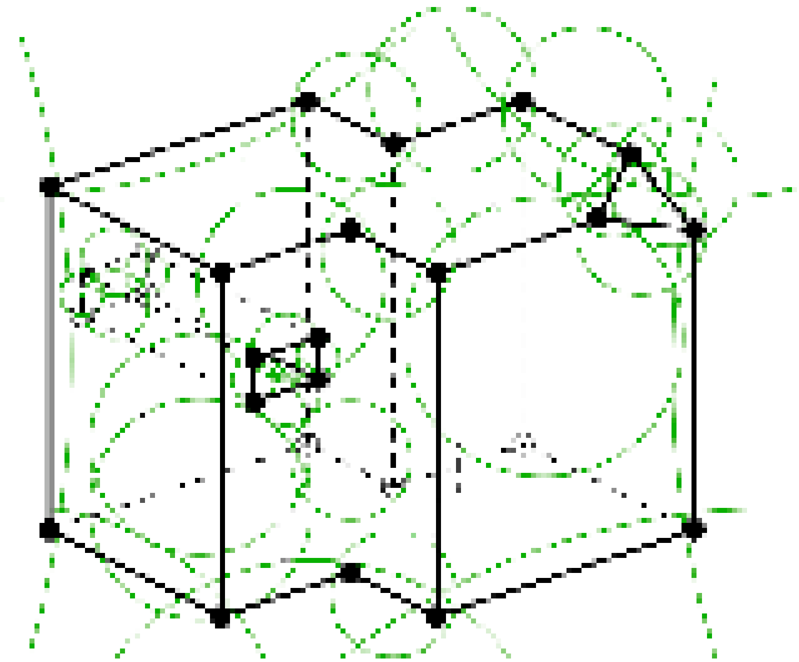


A Chazelle's polyhedron

[Chazelle 1984]

The existence of CDTs in 3d

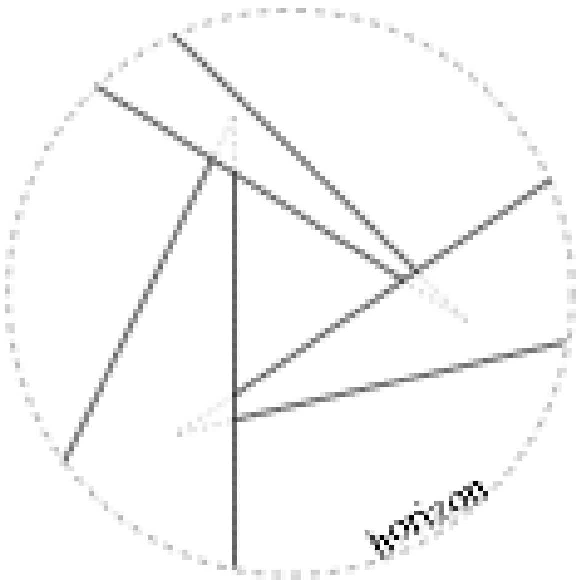
- An edge e in a PLC $\mathcal{X}$ is *strongly Delaunay* if there exists a circumball of e such that no other vertex of $\mathcal{X}$ lies inside or on the boundary of the ball.
- **Theorem [Shewchuk 1998].** If every edge of the PLC is strongly Delaunay, then it has a CDT.
- A *Steiner CDT* of $\mathcal{X}$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



Courtesy of J. Shewchuk

Proof of CDT Theorem (Shewchuk)

Lemma 3 *From any fixed vantage point p, X contains no cycle of consecutively overlapping strongly Delaunay constraining simplices.*



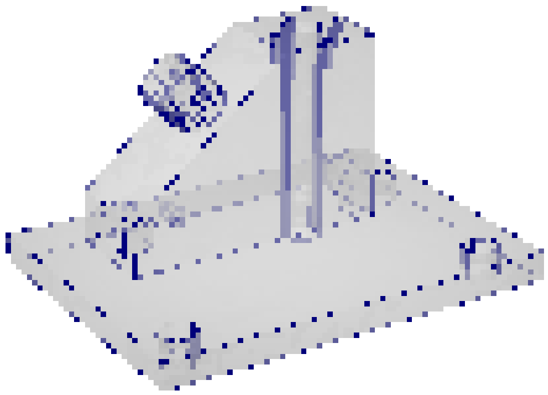
CDT algorithms

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

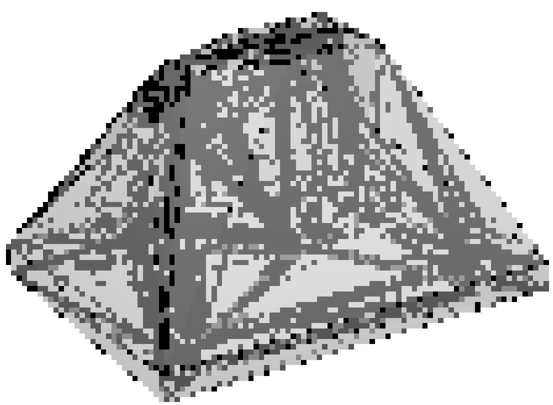
(1) **Initialization**: Creating a Delaunay tetrahedralization of the vertices of $\mathcal{X}$;

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

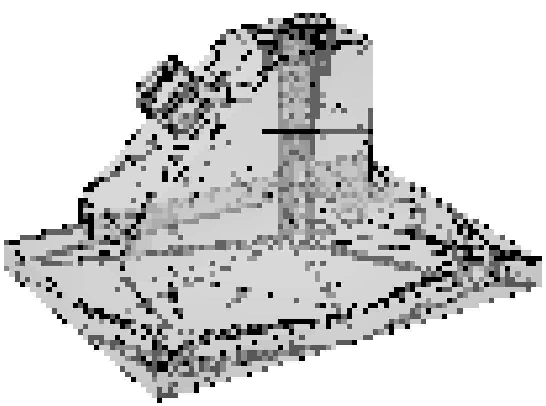
(3) **Polygon insertion**: Generating the Steiner CDT of $\mathcal{X}$.



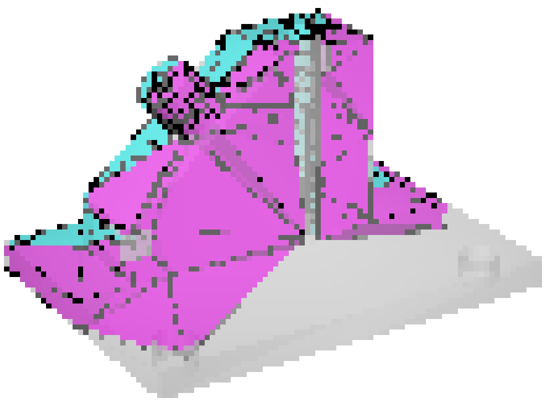
An input PLC $\mathcal{X}$



(1) Initialization

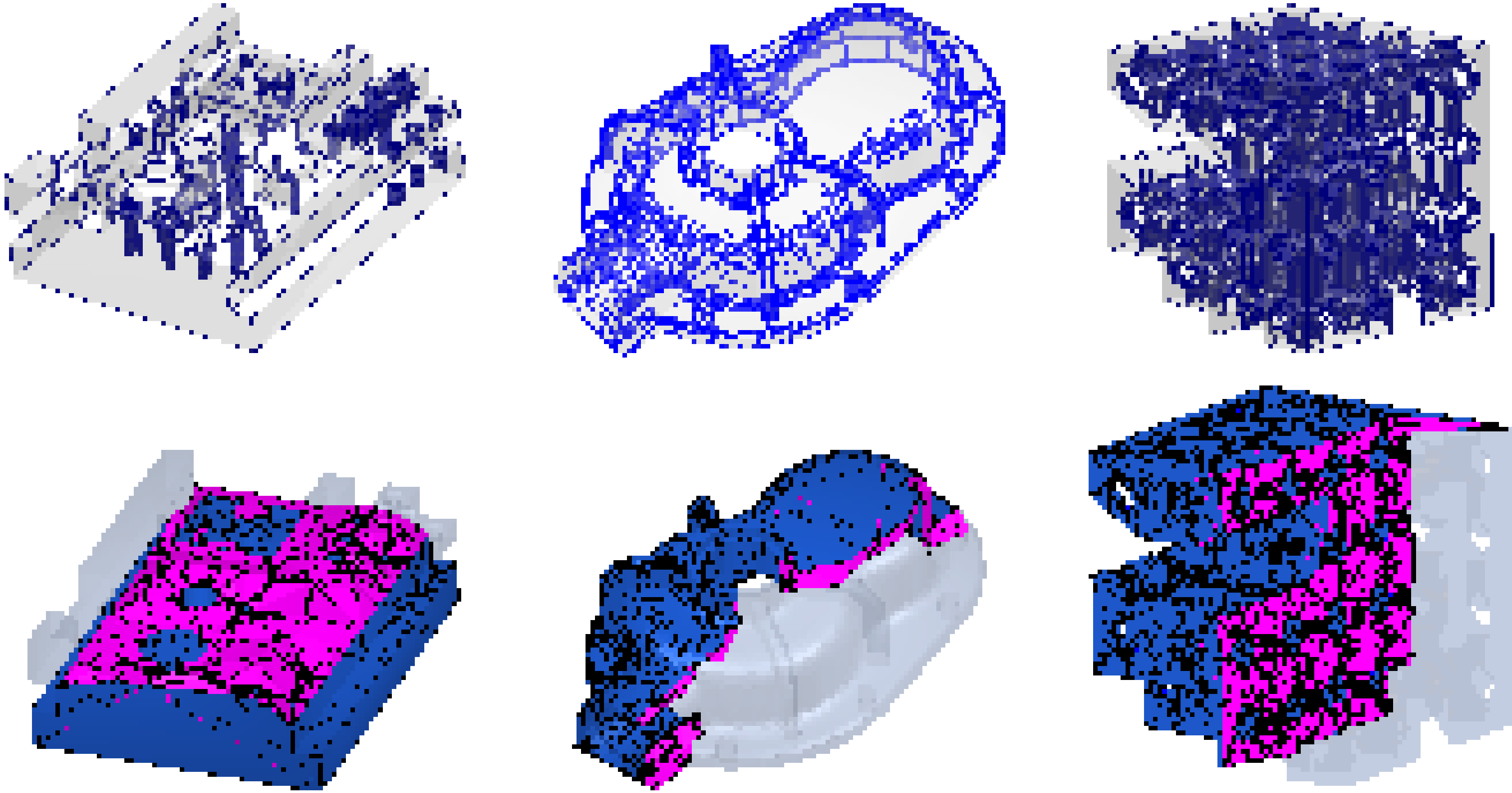


(2) Segment insertion



(3) Polygon insertion

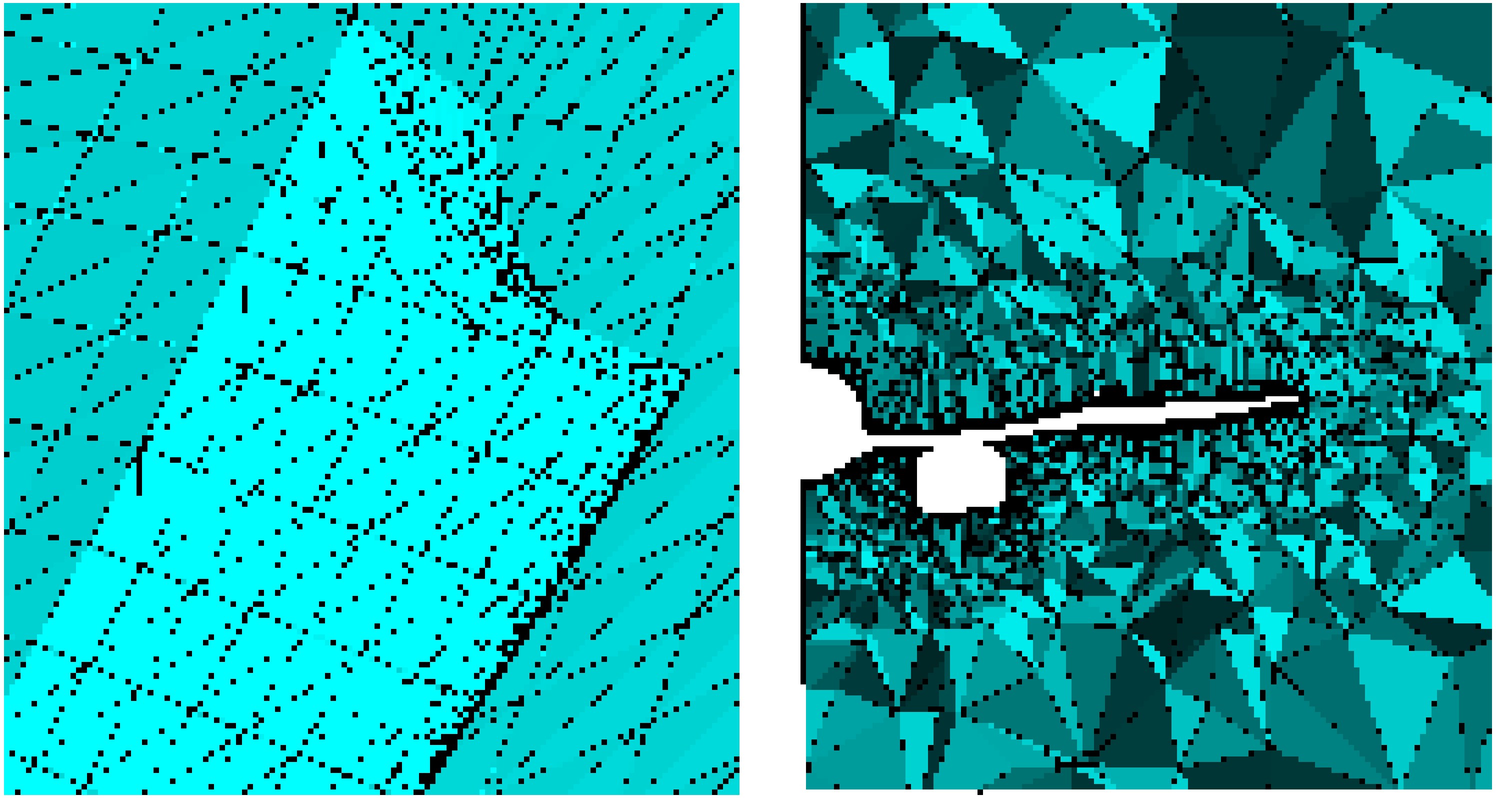
Examples



mohne-a	cognit	nasty_cb
Input: 2,760 pts, 5,560 tris	2,992 pts, 5,792 tris	8,630 pts, 17,782 tris
Added: 6,995 Steiner pts	8,490 Steiner pts	42,408 Steiner pts
CPU time: 0.57 sec.	0.54 sec.	5.34 sec.

Boundary recovery

- In many applications, a pre-discretized surface mesh is used as input, and it is required that this surface mesh be exactly preserved in the generated tetrahedral mesh, i.e., no subdivision of the surface mesh is allowed.



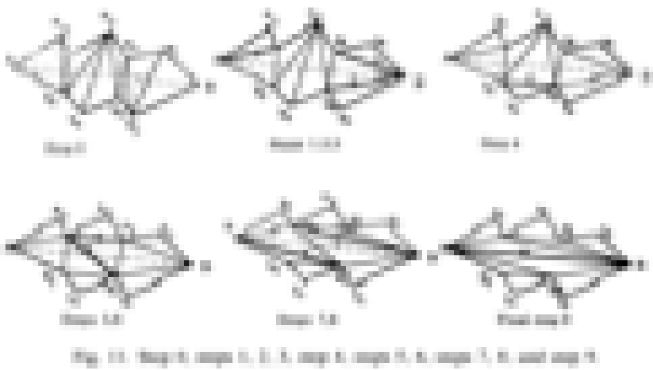
courtesy of acelab utexas

Classical boundary recovery methods

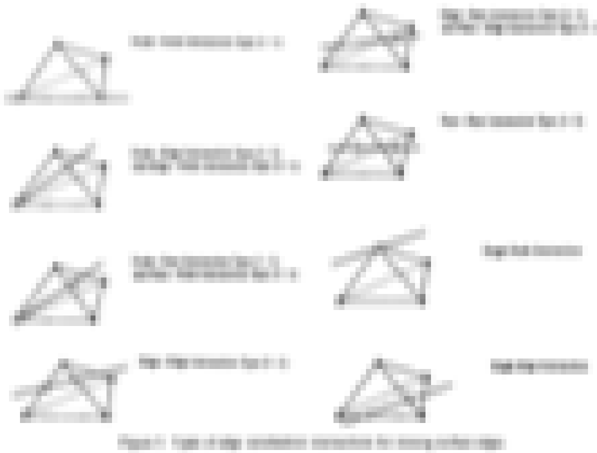
(1) Use edge/face swaps together with interior Steiner points insertion [George, Hecht, & Saltel 1991] (in TetMesh-GHS3D).

(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).

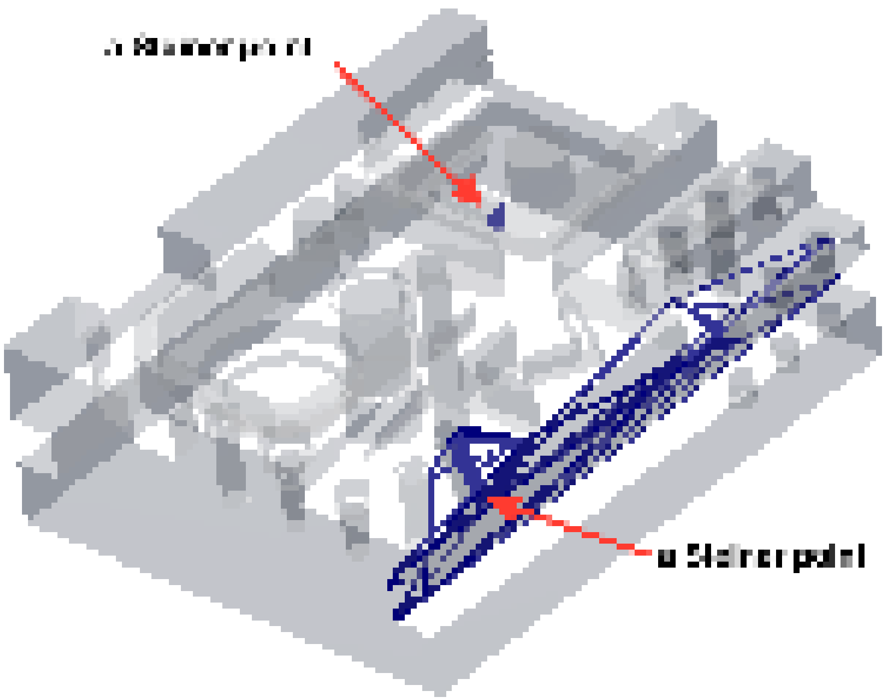


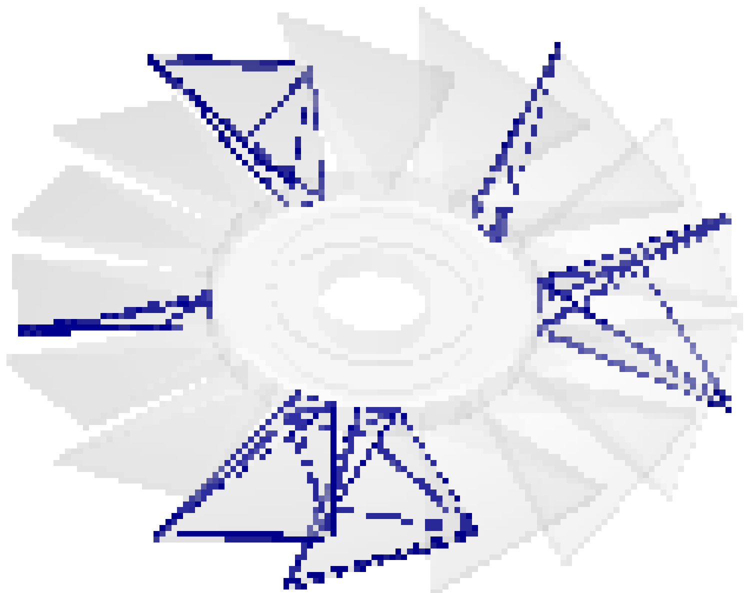
(1) [George, Hecht, and Saltel 1991]



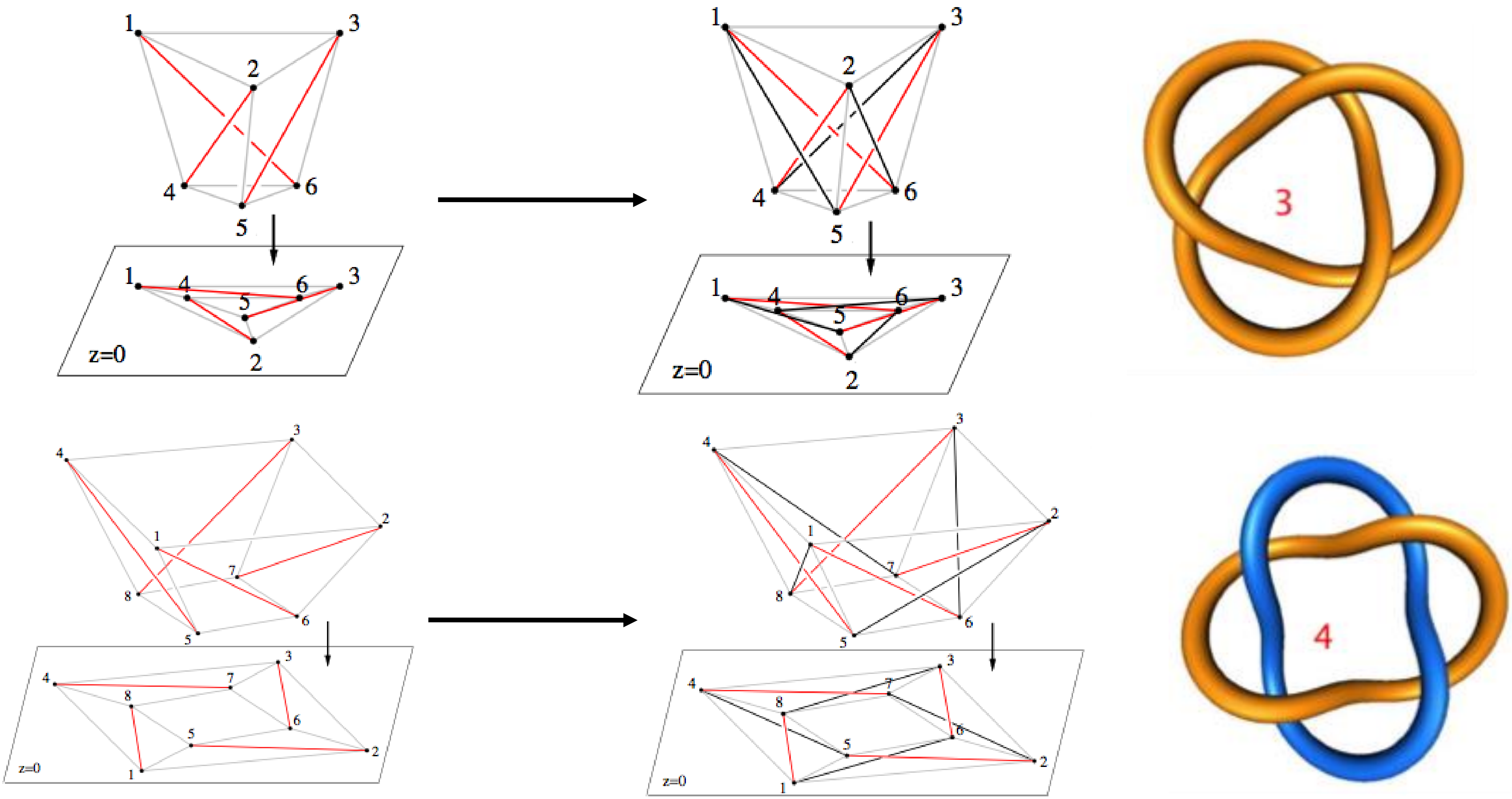
(2) [Weatherill and Hassan 1994]

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





Knots and Links

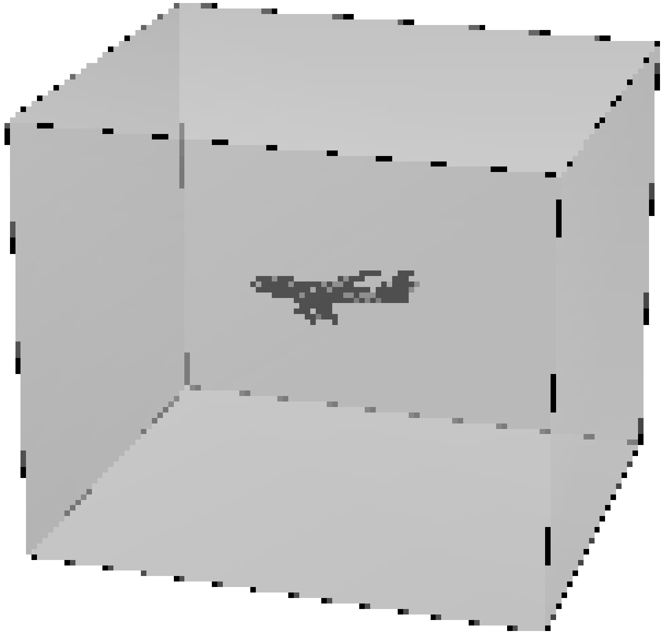


H. Si, Y. Ren, N. Lei, X. Gu, ***On tetrahedralisations containing knotted and linked line segments.***

Quality Mesh Generation in 3d

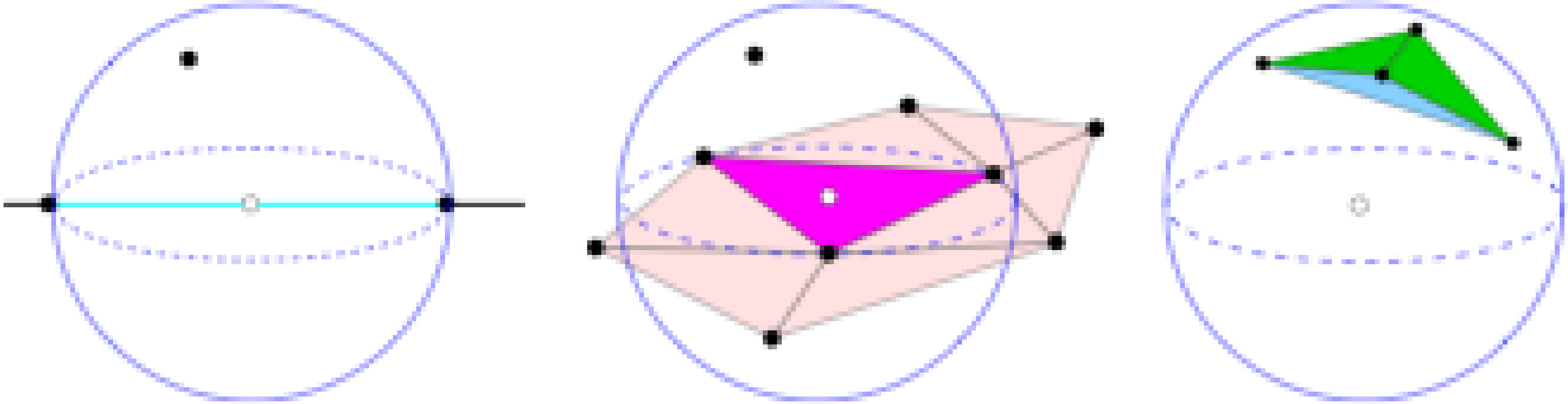
Mesh points creation

- 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];



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**

The algorithm [Ruppert and Shewchuk]

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;

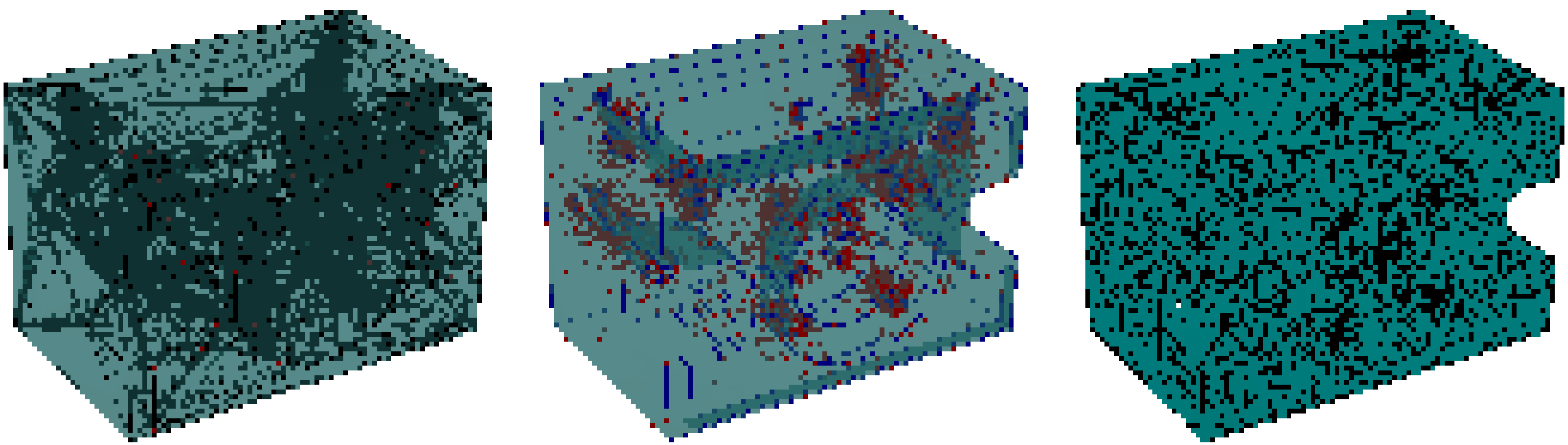
repeat:

  Create a new point by rule i, $i \in \{1, 2, 3\}$;

  Add v to V, update $\mathcal{D}$ of V;

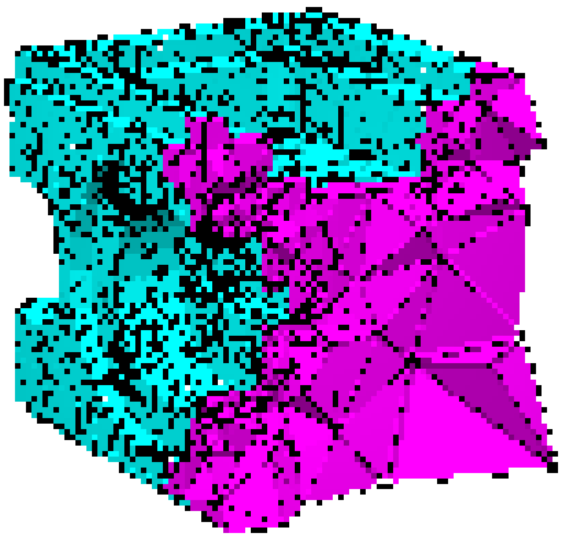
until {no new point can be generated};

return $\mathcal{D}$ of V;

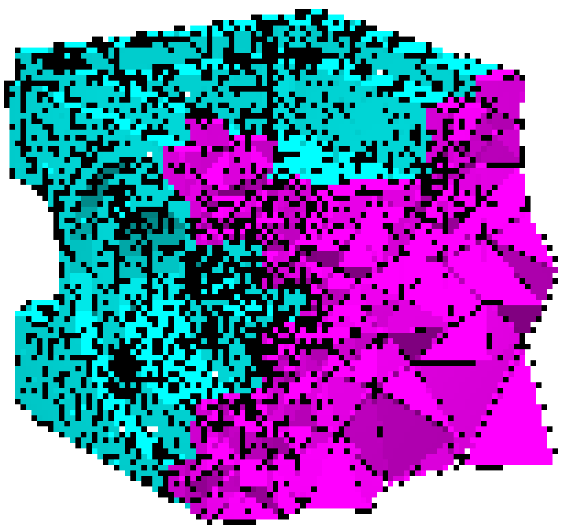


Guarantees in mesh quality and mesh size

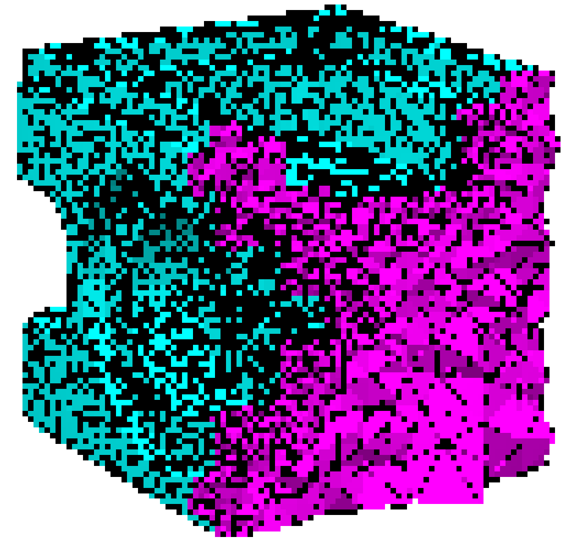
- (Mesh quality) Well-shaped tetrahedra, $\rho(t) \leq \rho_0, \forall t \in \mathcal{T}$.
- (Mesh size) Well-graded mesh, $\|\mathbf{v} - \mathbf{w}\| \geq \frac{\mathrm{lfs}(\mathbf{v})}{D+1}$, $D = \frac{(3+\sqrt{2})\rho_0}{\rho_0 - 2}$.
- (Mesh property) It is a conforming Delaunay tetrahedral mesh.



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



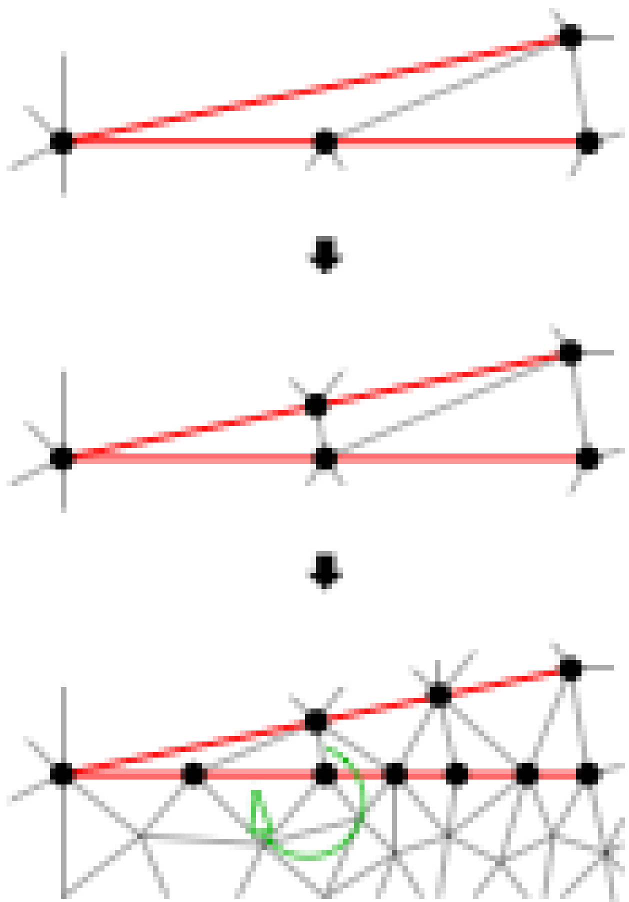
min. angle = 20.7°
$14,653$ nodes, 2.5 sec.
2.5 sec.



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

Small angles

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

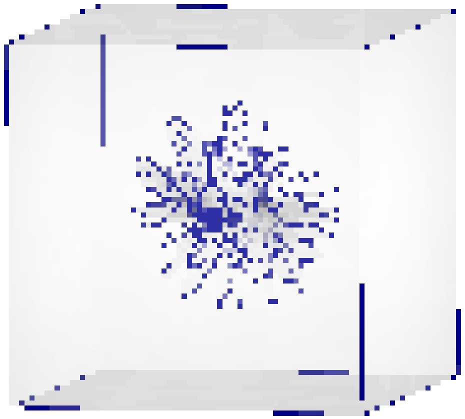


A subsegment is split.
New vertex encroaches upon another subsegment.

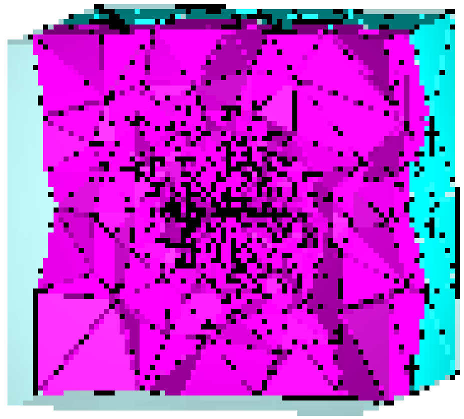
Another vertex is inserted, creating a very short edge.
Oops!

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

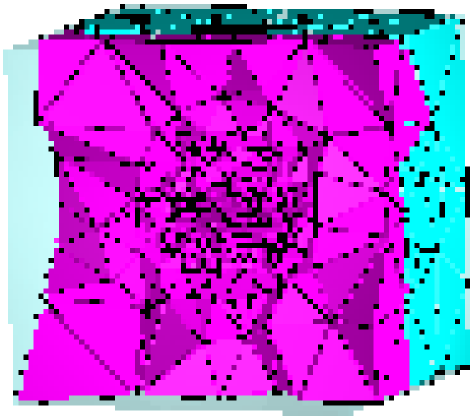
Constrained Delaunay refinement [Shewchuk and Si 2014]



Input test-64-6
161 vertices, 70 polygons

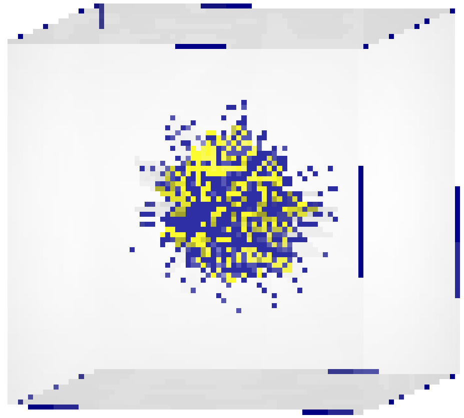


Tet mesh, 3,733 vertices
(cut along the Z-axis)

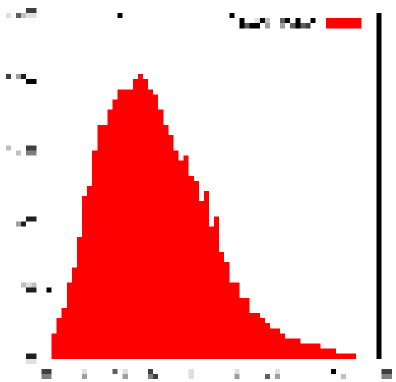


Tet mesh, 23,727 tets
(cut along the Y-axis)

refined "fan blades"



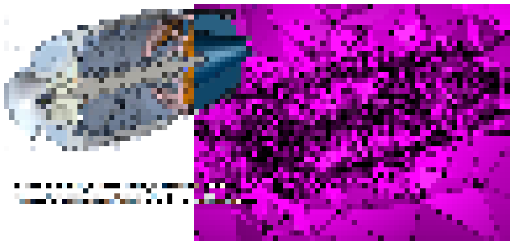
remaining skinny tetrahedra
(radius-edge ratios > 2)



plane angles

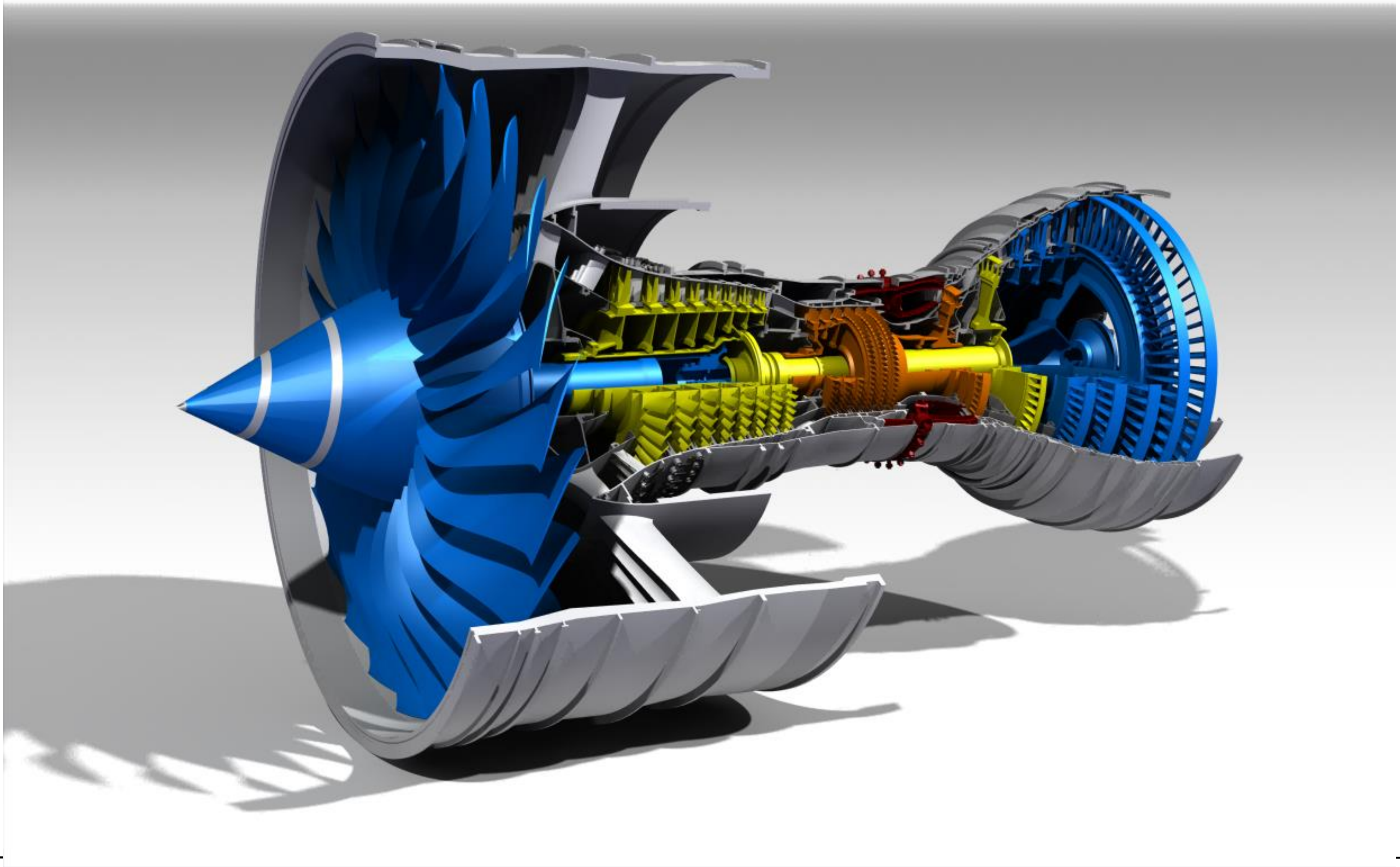
The TetGen Project

- 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.

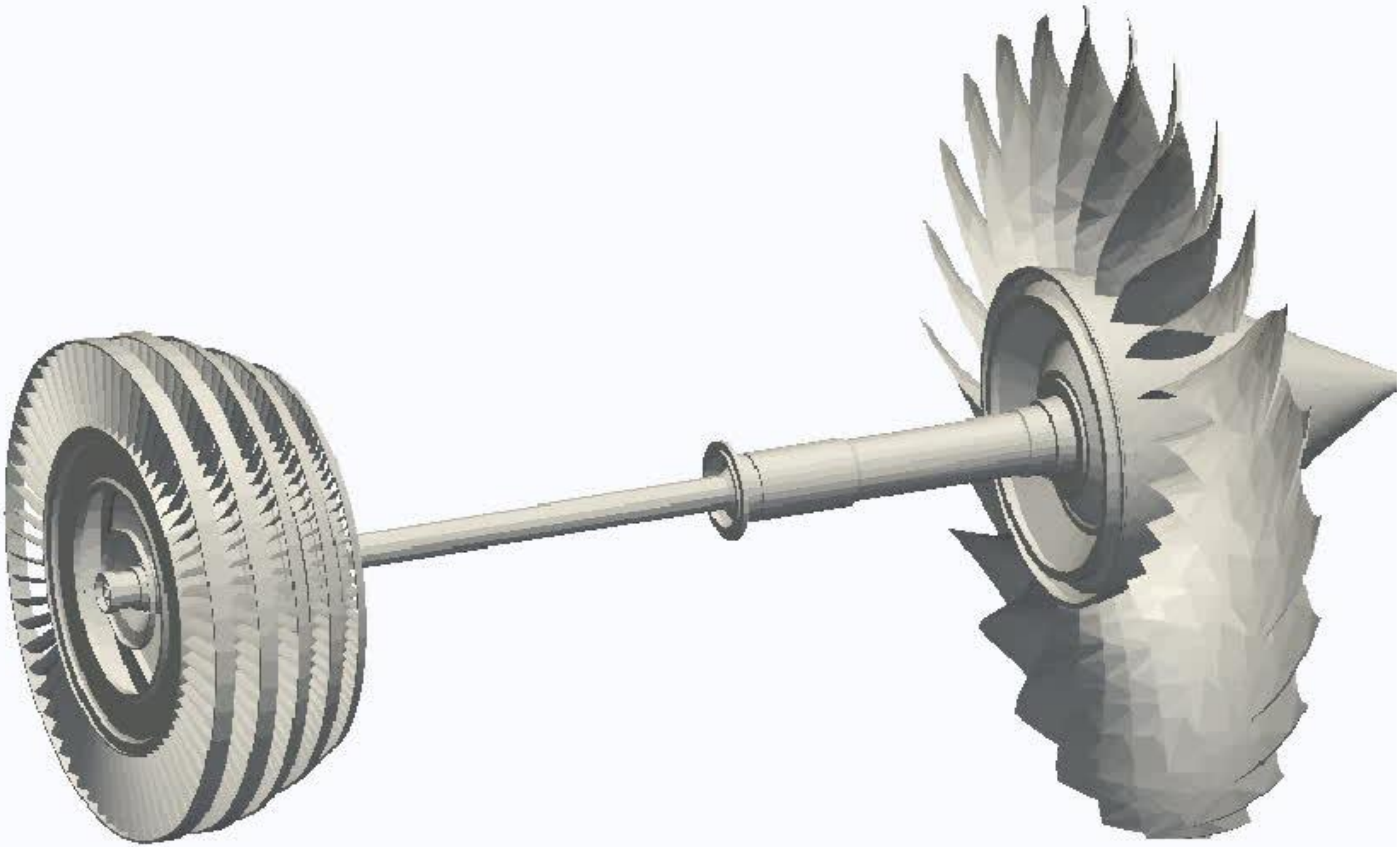


- H. Si, TetGen, a Delaunay-based Tetrahedral Mesh Generator, ACM Trans. Math. Softw., **41** (2):11:1–11:36, February **2015**.

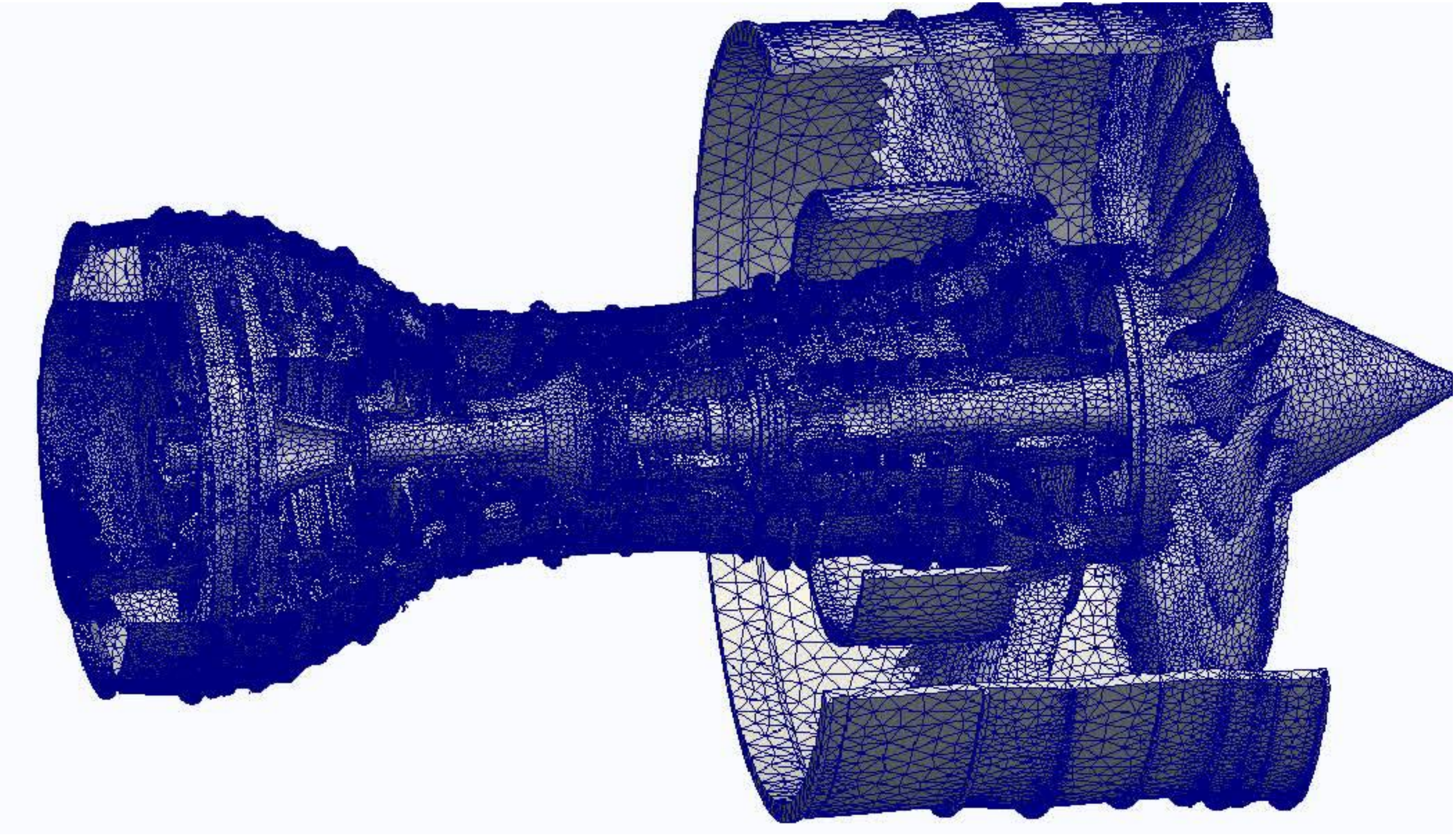
An example: RR-Trent900



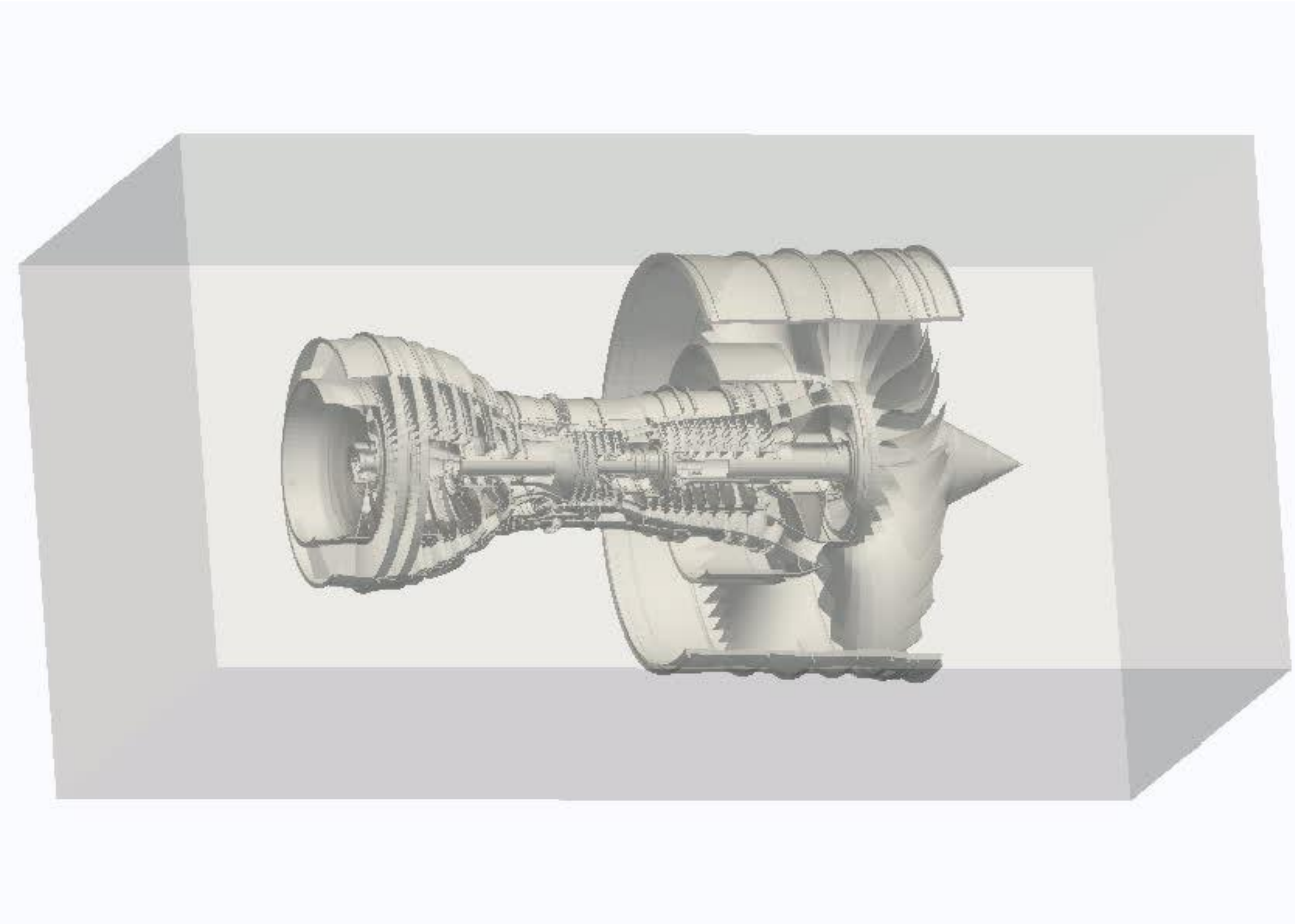
An example: RR-Trent900



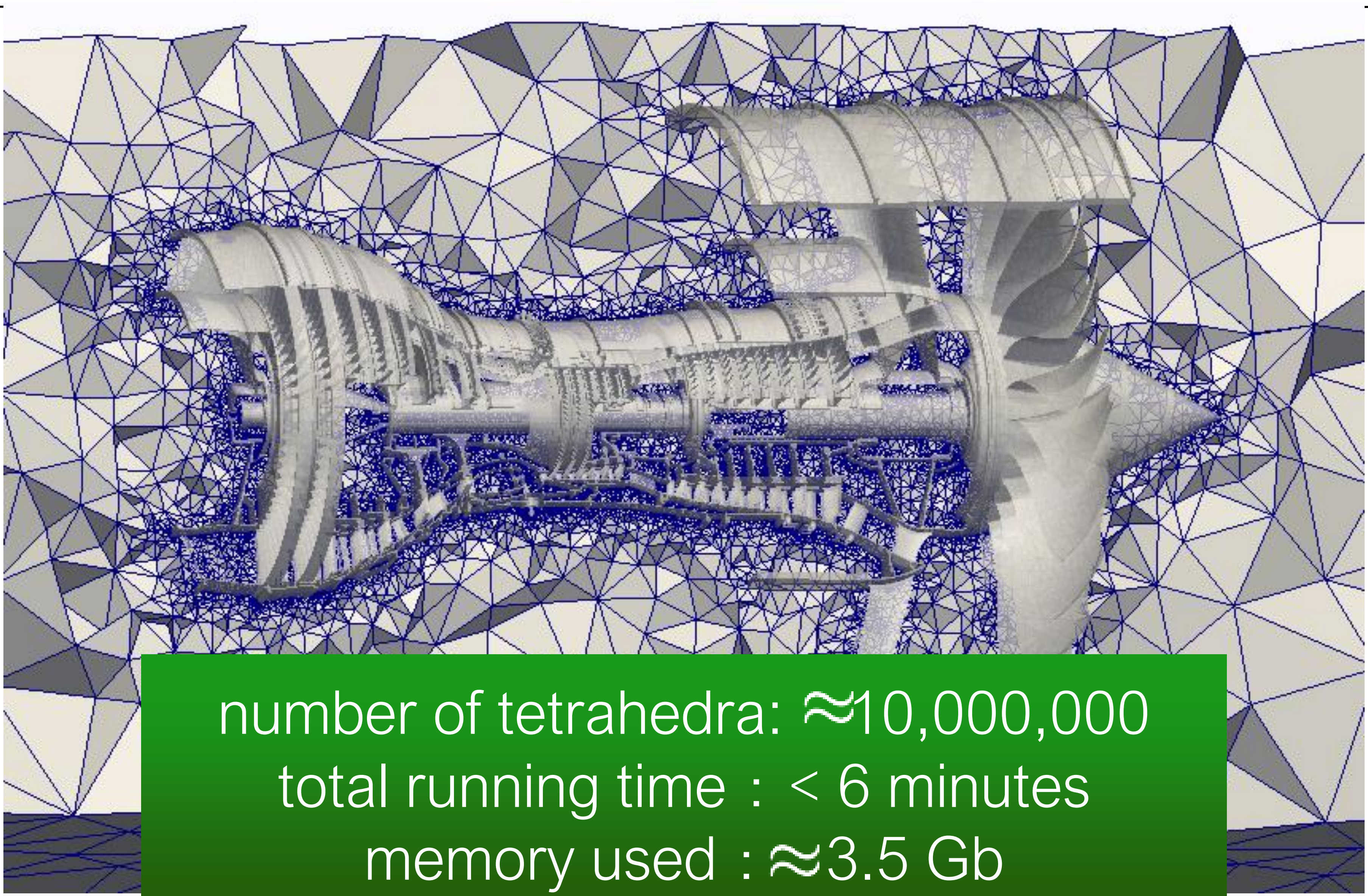
An example: RR-Trent900



An example: RR-Trent900



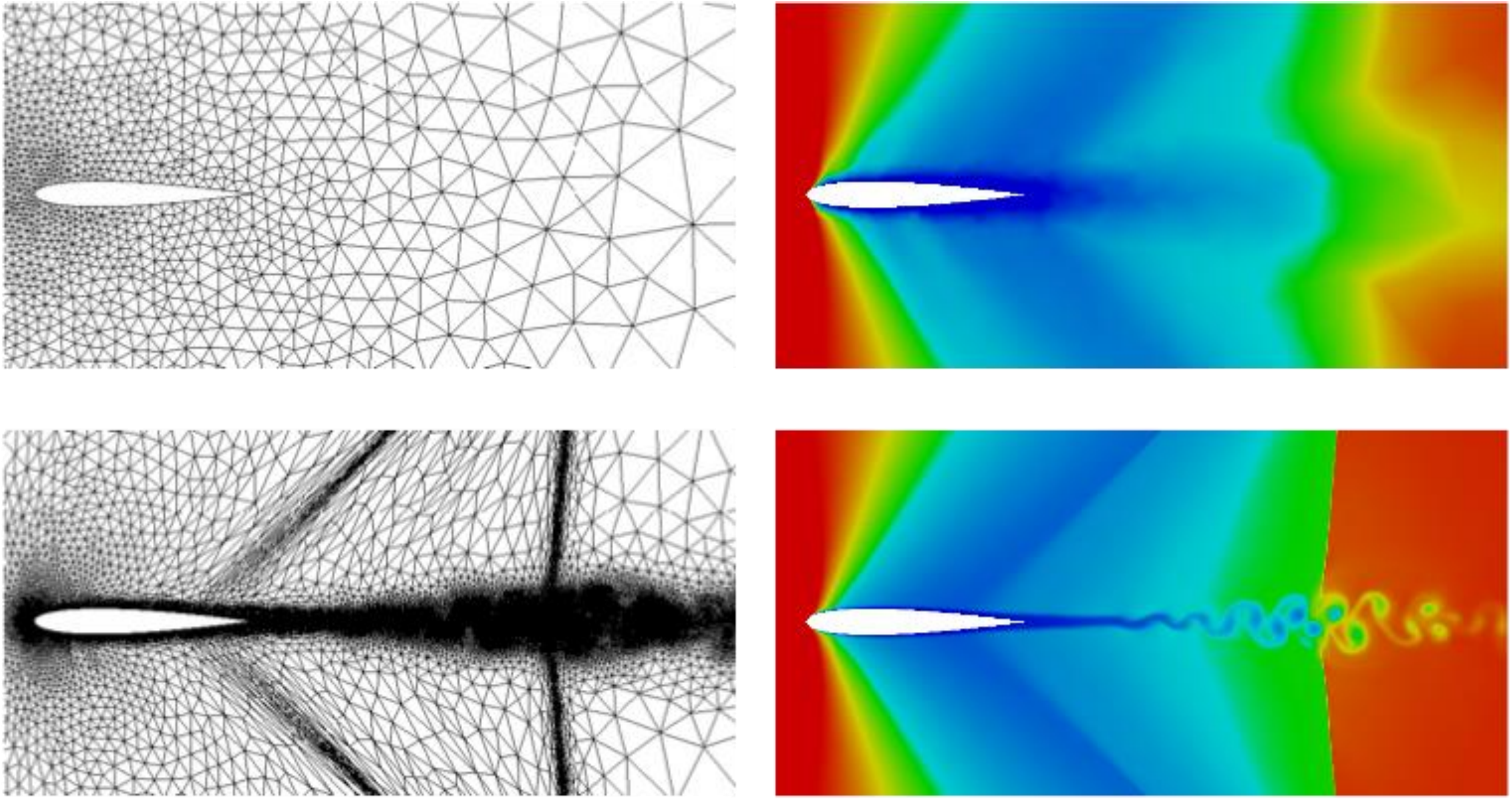
An example: RR-Trent900



number of tetrahedra: ≈10,000,000
total running time : < 6 minutes
memory used : ≈3.5 Gb

Outline

Why mesh adaptation



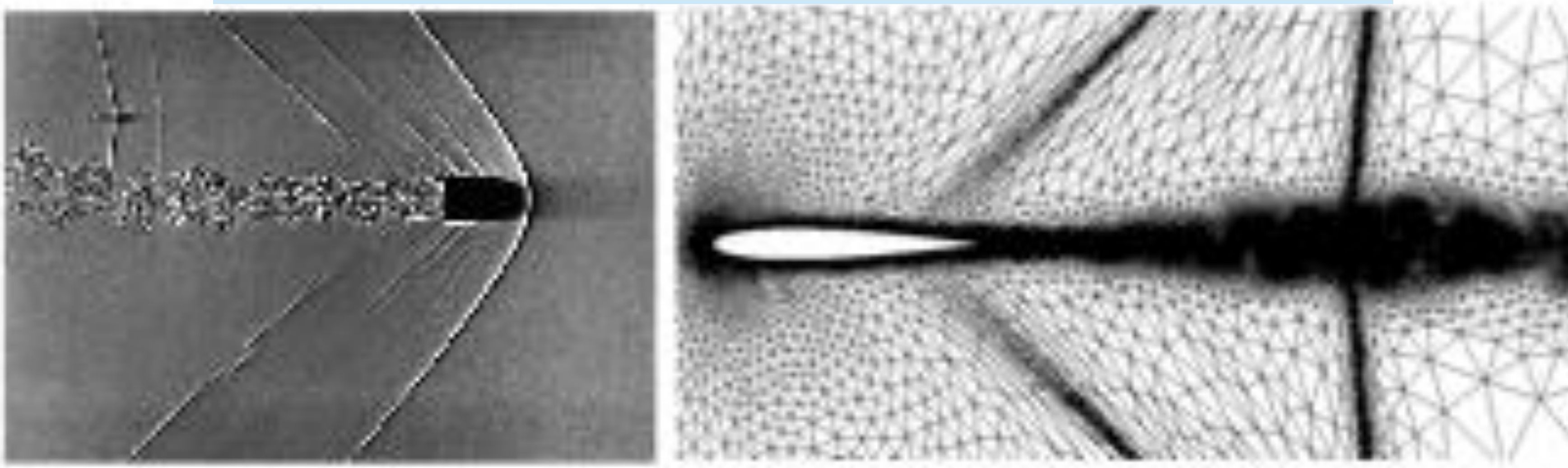
Adapted meshes and density fields (iter. 0, 9).

Images from Frey's IMR talk (2005)

Anisotropic mesh generation

- 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 com

Anisotropy: why and where?



Images from Frey's IMR talk (2005)

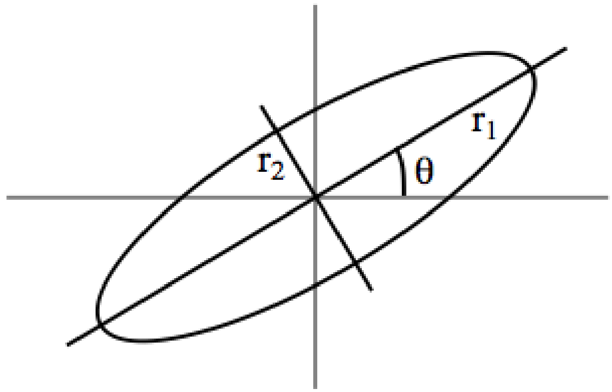
Metric-based Anisotropic Mesh Adaptation

Description of anisotropy using a metric tensor

- Anisotropy means the way distance and angles are distorted.
- Anisotropy can be described through a field $\mathcal{M}$ of metric tensors associated with a space domain $\Omega \subseteq \mathbb{R}^d$, where each metric tensor $M(\mathbf{x}) \in \mathcal{M}, \mathbf{x} \in \Omega$ is a $d \times d$ symmetric positive definite matrix.
- A metric tensor M can be decomposed as

$$M = R\Lambda R^T = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

The *unit ball* $\mathbf{x}^T M \mathbf{x} = 1$ is an oriented ellipse where $r_1 = \frac{1}{\sqrt{\lambda_1}}$ and $r_2 = \frac{1}{\sqrt{\lambda_2}}$.



Anisotropic distances

- Given an open curve $C \subset \Omega$, the length of C with respect to $\mathcal{M}$ is defined as:

$$l_{\mathcal{M}}(C) = \int_{t=0}^{1} \sqrt{\mathbf{v}(t)^t M(c(t))\mathbf{v}(t)} \mathrm{d}t,$$

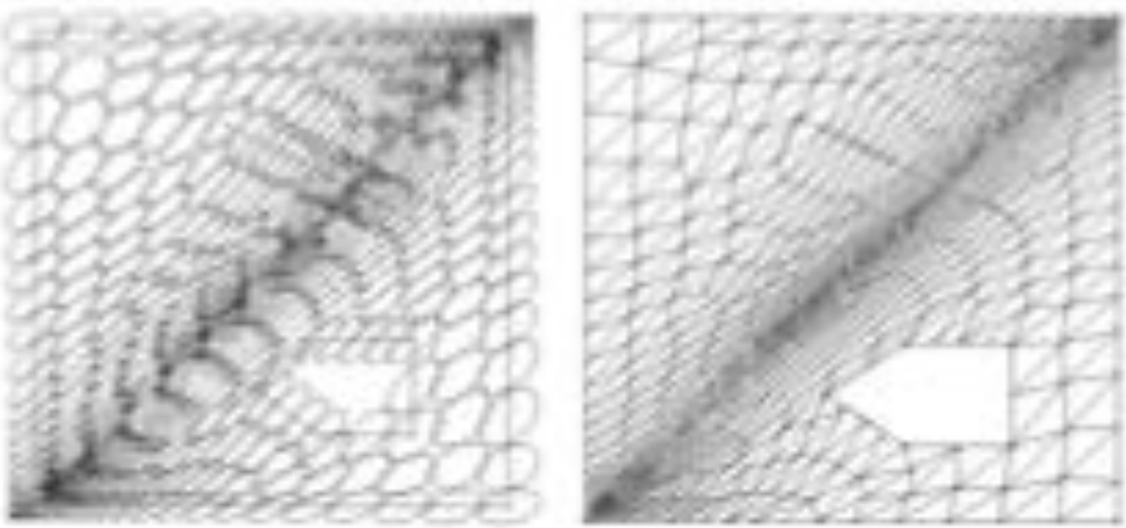
  where $c(t) : \mathbb{R} \to \mathbb{R}^d, t \in (0, 1)$ denotes a parameterization of C and $\mathbf{v}(t) = \partial c(t)/\partial t$ is the tangent vector.

- The *geodesic distance* $d_{\mathcal{M}}(\mathbf{x}, \mathbf{y})$ between two points $\mathbf{x}, \mathbf{y} \in \Omega$ is defined as the length of the (possibly non-unique) shortest curve C that connects $\mathbf{x}$ and $\mathbf{y}$:

$$d_{\mathcal{M}}(\mathbf{x}, \mathbf{y}) = \min(l_{\mathcal{M}}(C)).$$

Metric-based Mesh Adaptation

- In the majority of works concerning anisotropic mesh generation, a (discrete) metric tensor field $\mathcal{M}$ (e.g., defined on the vertices) is used to describe the anisotropic feature of the domain.
- Then, a *uniform mesh* with equal edge length with respect to the metric tensor field $\mathcal{M}$ is sought.



BAMG - a metric-based code

BAMG: Bidimensional Anisotropic Mesh Generator

Frédéric Hecht *

draft version v1.00 decembre 2006

The software **bamg** is a Bidimensional Anisotropic Mesh Generator, It a part of FreeFem++ software `www.freefem.org/ff++`

17, ε =0.35 0.27, 3797T 1936V

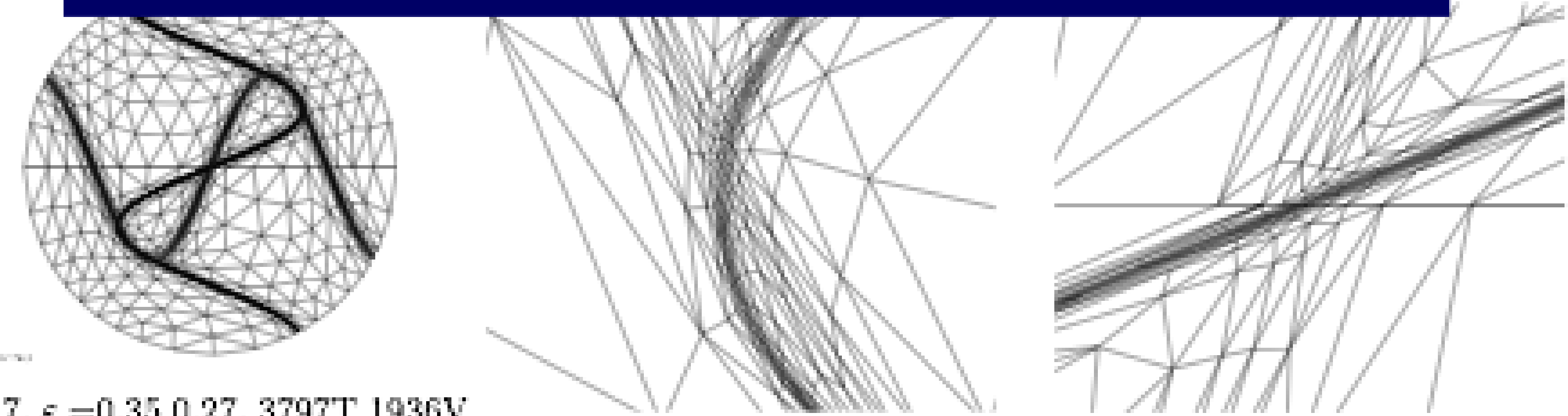
BAMG - a metric-based code

BAMG: Bidimensional Anisotropic Mesh Generator

Frédéric Hecht *

draft version v1.00 decembre 2006

The anisotrope metric adaption schema gives the best result but mathematically we have not real proof.

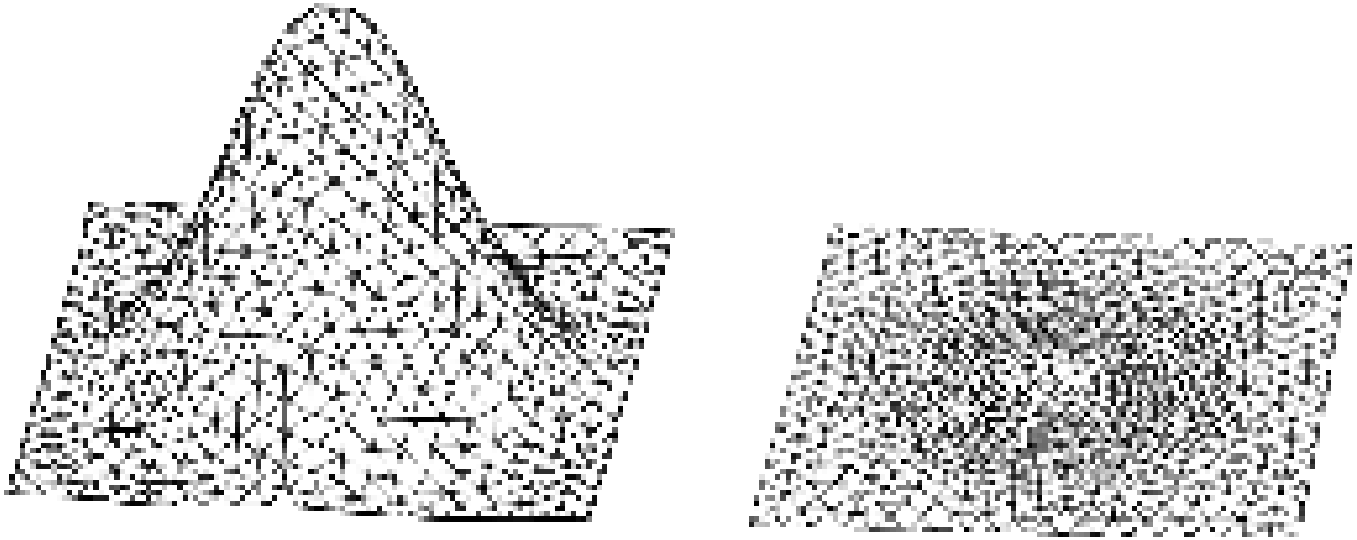


17, ε =0.35 0.27, 3797T 1936V

High Dimensional Embeddings

Anisotropy through High Dimension Embedding

The Idea: Use additional dimensions to resolve the anisotropy.



(Courtesy of B. Lévy)

This example shows that an anisotropic mesh in $\mathbb{R}^2$ corresponds to an isotropic mesh in $\mathbb{R}^3$.

Surface Emending in $\mathbb{R}^6$ [Canás and Gortler 2006, Lai et al 2010]

Let Γ be a surface in $\mathbb{R}^3$. Let $\phi : \Gamma \subset \mathbb{R}^3 \to \mathbb{R}^6$ be a map defined as,

$$\phi(A) := \begin{bmatrix} x \\ y \\ z \\ sn_x \\ sn_y \\ sn_z \end{bmatrix}$$

where A is a point in surface Γ whose coordinates are x, y and z, respectively, and n_x, n_y and n_z are the components of the normal to the surface Γ at the point p.
The constant $s \in (0, +\infty)$ is a parameter for capturing the anisotropy.

Lengths and angles in 6d

Define the scalar product in $\mathbb{R}^6$ to be:

$$(A, B)_{6d} = \underbrace{x_A x_B + y_A y_B + z_A z_B}_{I} + s^2 \underbrace{(n_x w_x + n_y w_y + n_z w_z)}_{II}.$$

This parameter will balance the contribution of the quantities I and II on whole value of $(A, B)_{6d}$. Since $I \in [-d^2, d^2]$ and $II \in [-1, 1]$, where d is the measure of the diagonal of the bounding box of Γ, we need an additional constant to make I and II almost comparable. We decide to modify $(A, B)_{6d}$ in such a way

$$(A, B)_{6d} = x_A x_B + y_A y_B + z_A z_B + (h_\Gamma s)^2 (n_x w_x + n_y w_y + n_z w_z) .$$

where

$$h_\Gamma = \frac{d_x + d_y + d_z}{3} ,$$

here d_x, d_y and d_z are the dimension of the bounding box of Γ.

Given two points A and B that lie on the surface Γ, we define the length of the segment l_{AB}^{6d} as

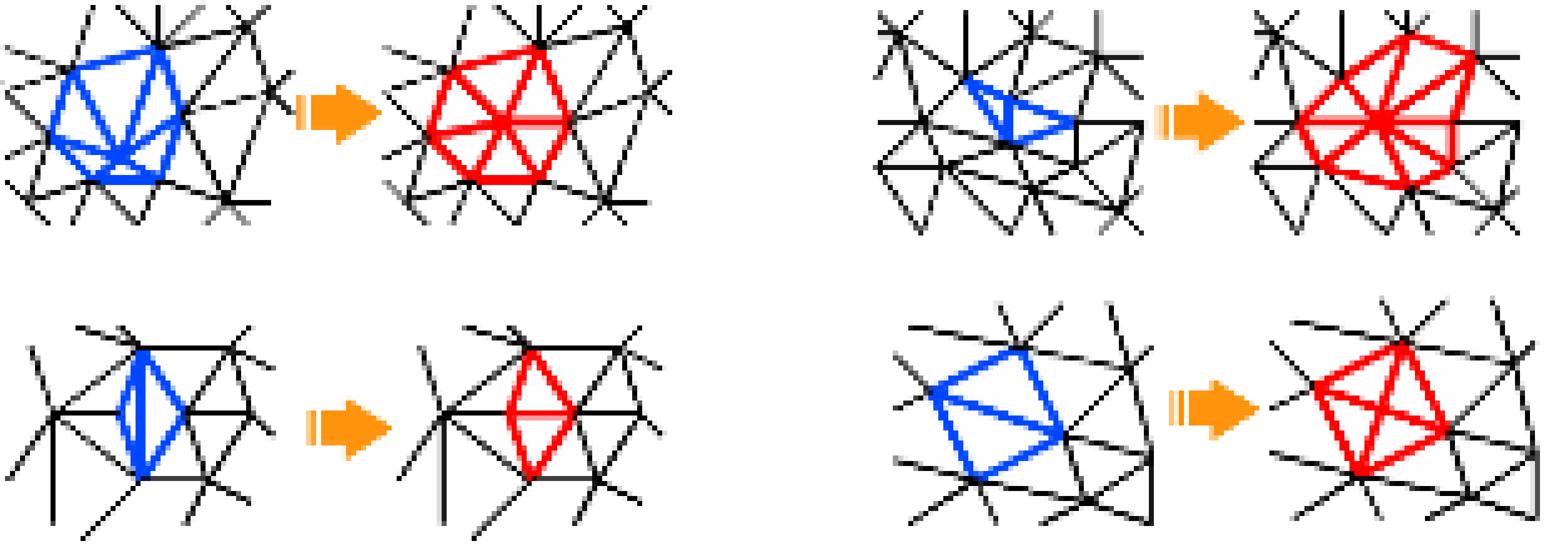
$$l_{AB}^{6d} := \|A - B\|_{6d} = \sqrt{(A - B, A - B)_{6d}} .$$

Given three points $A, B, C \in \Gamma$ we define the **6d-angle** ϑ as

$$\cos_{6d}(\vartheta) := \frac{(A - C, B - C)_{6d}}{\|A - C\|_{6d} \|B - C\|_{6d}}$$

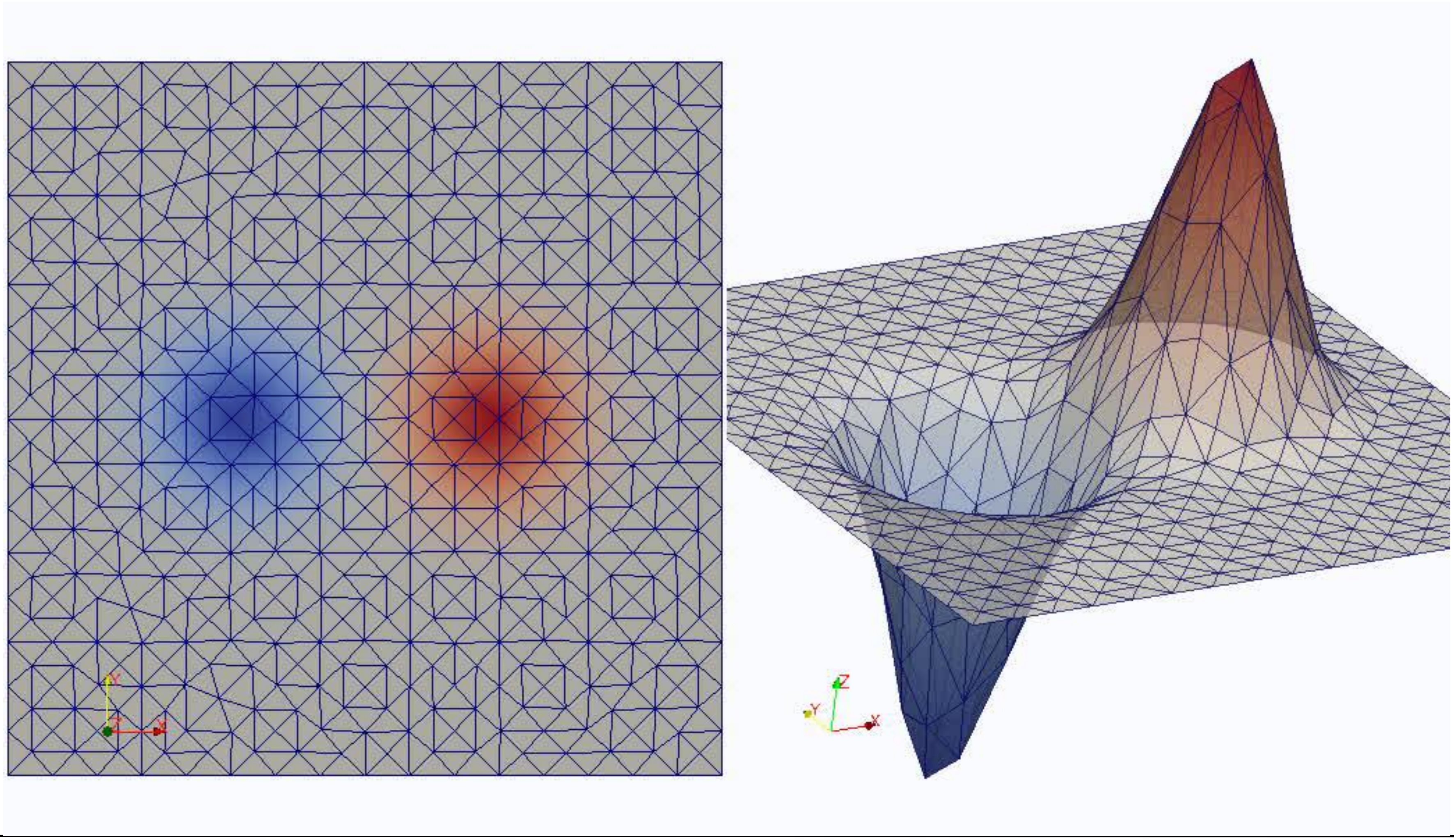
Mesh adaptation (using 6d lengths and angles)

- Starting from an initial mesh of a surface $\Gamma \subset \mathbb{R}^3$
- Evaluate the lengths of the angles of the triangles in $\mathbb{R}^6$.
- Perform the standard local mesh adaptation operations to make the mesh as uniform as possible in $\mathbb{R}^6$.

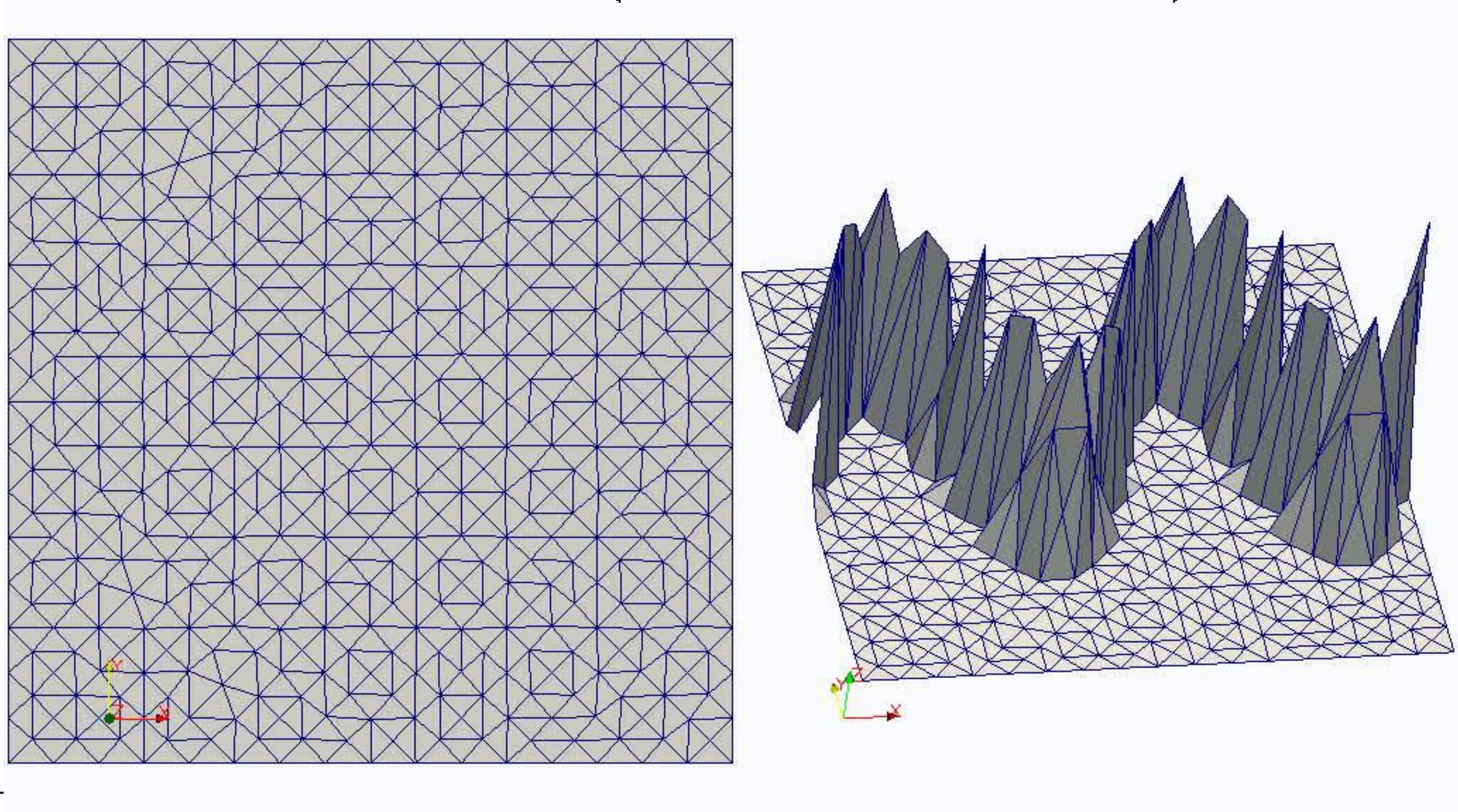


Examples

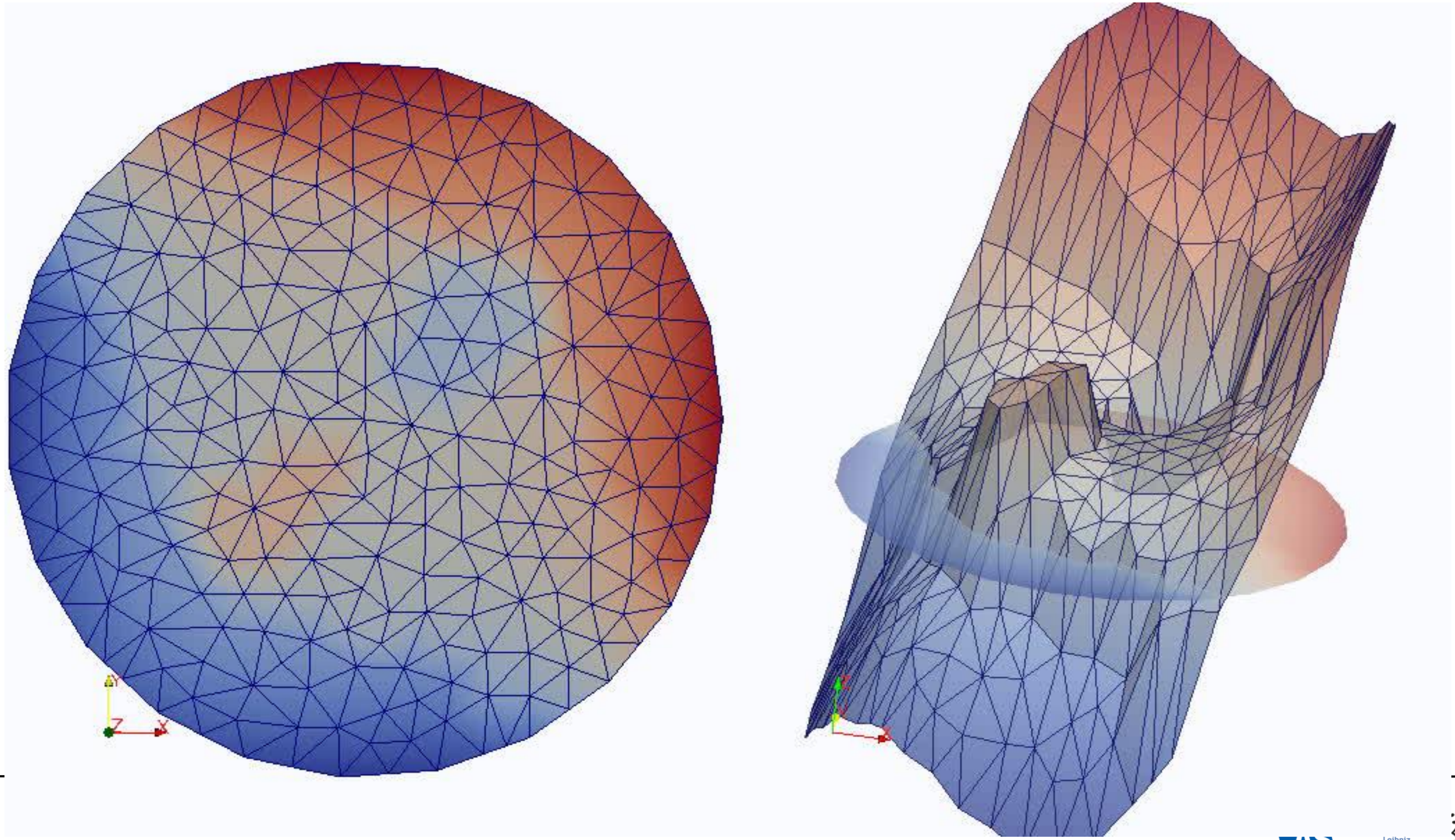
$$f = e^{-20((x-0.25)^2+y^2)} - e^{-20((x+0.25)^2+y^2)}$$



$$f(x,y) = \tanh\left(-100\left(y - \frac{1}{4}\sin(2\pi x)\right)^2\right)$$



$$f = (10x^3 + y^3) + atan2(0.001, \sin(5y) - 2x)$$
$$+(10y^3 + x^3) + atan2(0.01, \sin(5x) - 2y)$$



Mesh adaptation via HDE

Contrary to the classical mesh adaptation procedure, the proposed adaptation strategy in this paper does not involve both the estimation of an error and the construction of a metric field. In each iteration of the mesh adaptation, we use the following steps:

$$\text{SOLVE} \longrightarrow \text{RECOVER GRADIENT} \longrightarrow \text{ADAPT},$$

and this process stops when it converges or a desired maximum number of iterations is reached.

The embedding map

Let the piecewise polynomial solution of the PDE is u_h, we define the following embedding: $\Phi_{u_h} : \mathbb{R}^2 \to \mathbb{R}^5$ defined as

$$\Phi_{u_h}(\mathbf{x}) := (x,\, y,\, s\, u_h(x,\, y),\, s\, g_x(x,\, y),\, s\, g_y(x,\, y))^t\,,$$

where s is a user-specified parameter as before and

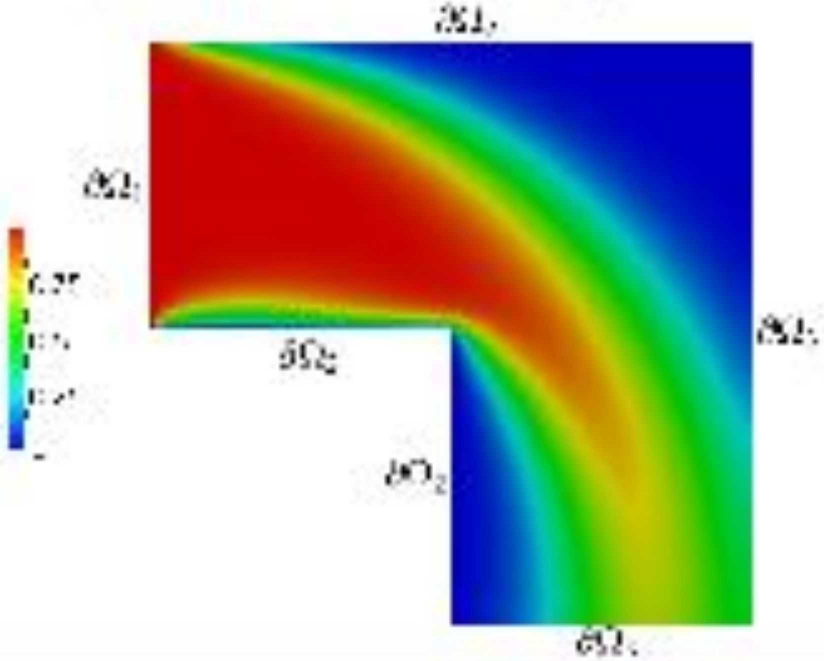
$$g_x(x,\, y) := [\nabla u_h\,(x,\, y)]_x\,, \qquad g_y(x,\, y) := [\nabla u_h\,(x,\, y)]_y\,,$$

here $[\nabla u_h\,(x,\, y)]_x$ and $[\nabla u_h\,(x,\, y)]_y$ are the x and y components of the gradient of the discrete solution u_h, respectively.

An example

$$
\begin{cases}
-\mu \Delta u + \vec{\beta} \cdot \nabla u &= 0 \quad \text{in } \Omega\,, \\
u &= 1 \quad \text{in } \partial\Omega_1\,, \\
u &= 0 \quad \text{in } \partial\Omega_2\,, \\
\mu \frac{\partial u}{\partial n} &= 0 \quad \text{in } \partial\Omega_3\,,
\end{cases}
$$

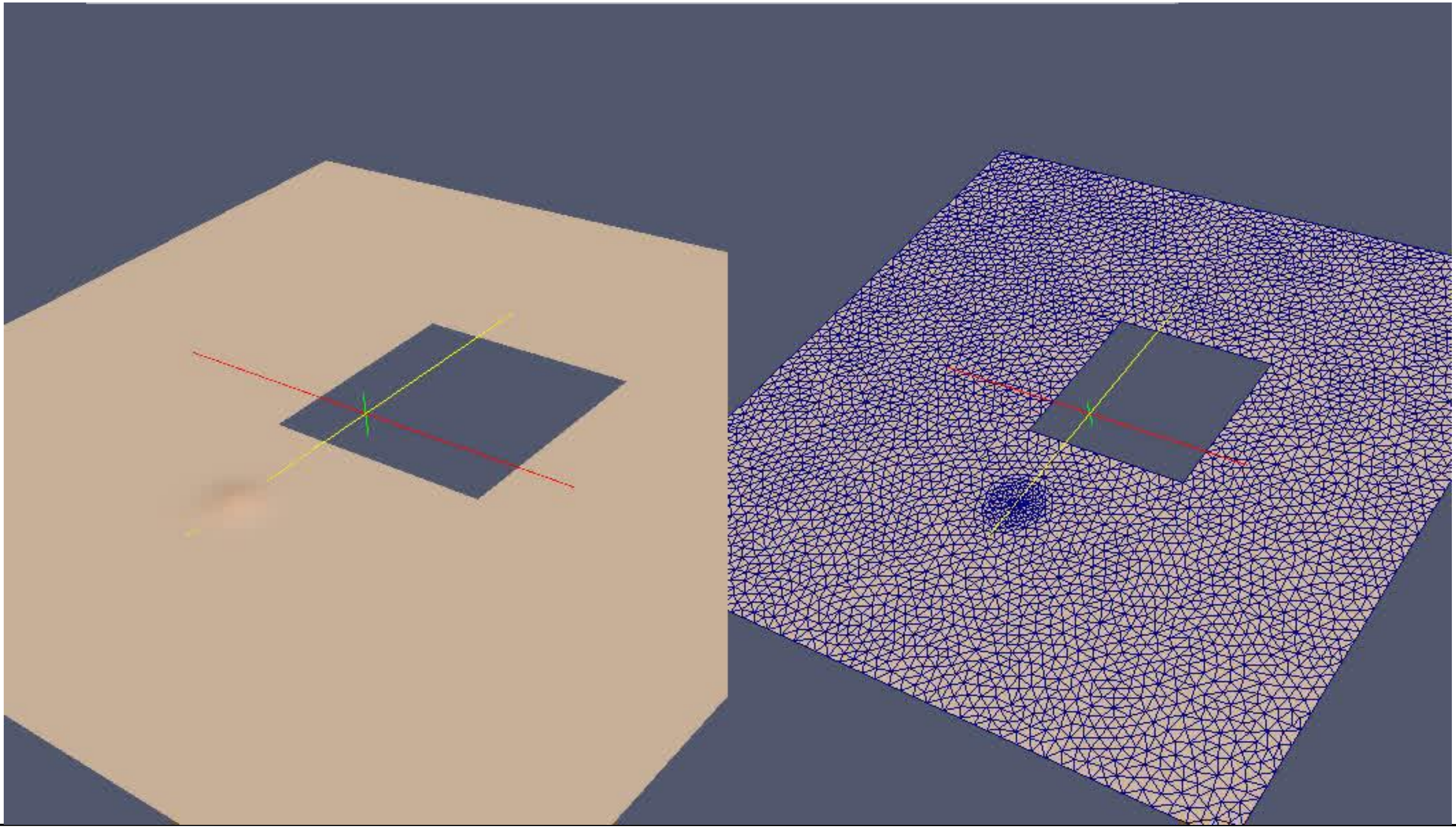
here $\mu = 0.05$, $\vec{\beta} = (x, -y)^t$.



Example: Wave

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

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



Outline

1. Introduction
2. Triangular Mesh Generation
3. Tetrahedral Mesh Generation
4. Mesh Adaptation
5. Further Topics

Hybrid Methods

CFD Meshing

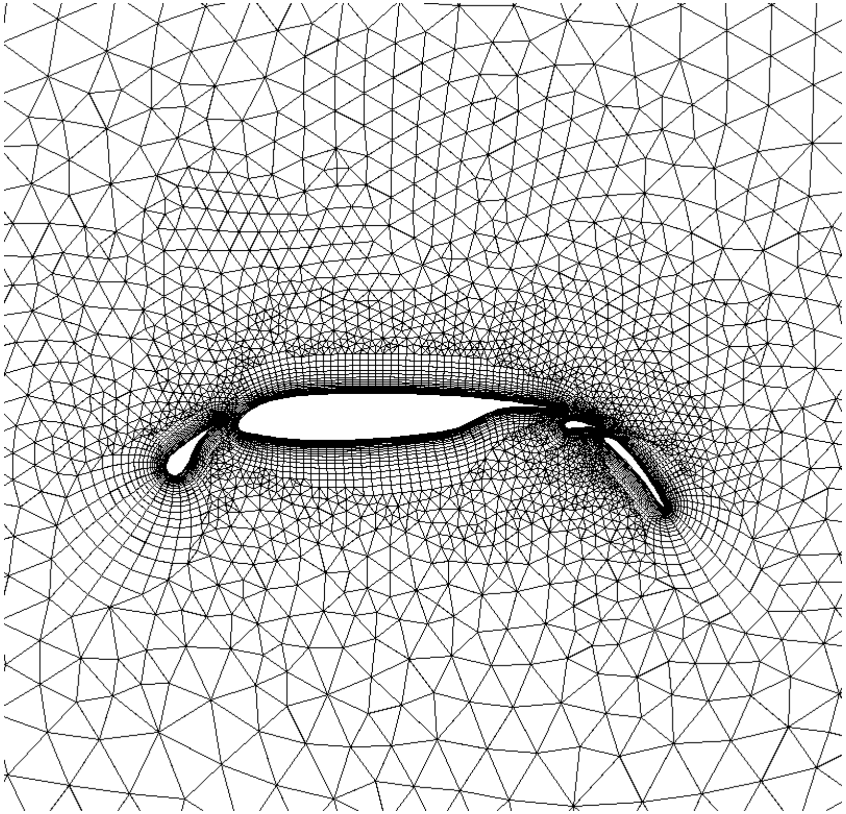


Image courtesy of Roy P. Koomullil, Engineering Research Center, Mississippi State University, http://www.erc.msstate.edu/~roy/

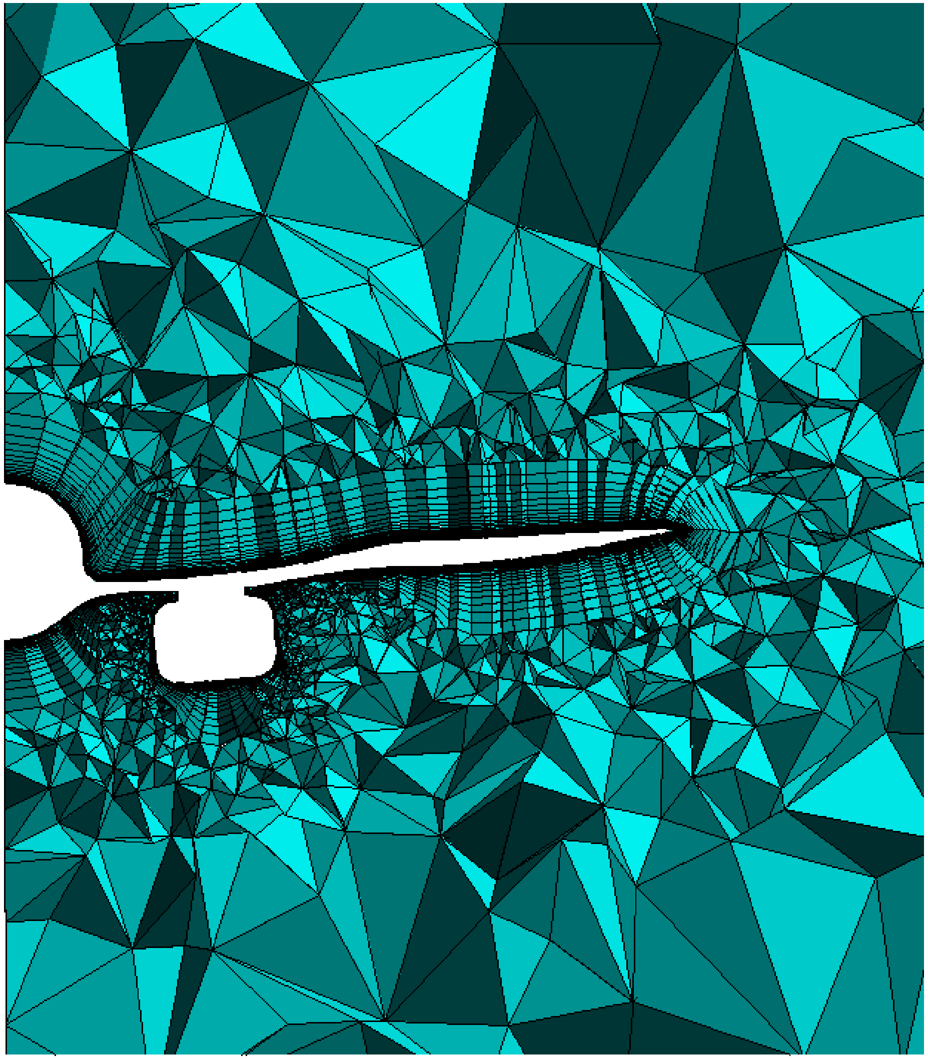
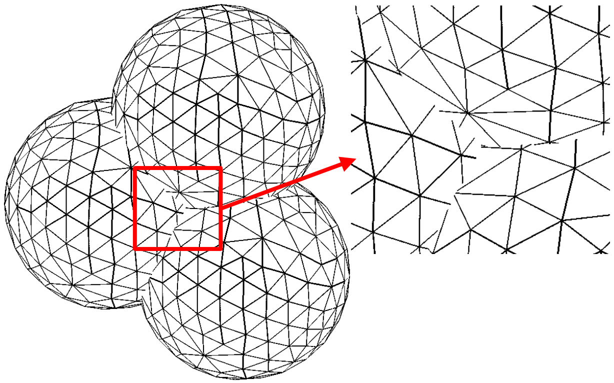
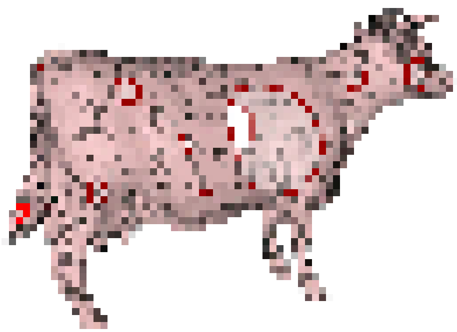


Image courtesy of acelab, University of Texas, Austin, http://acelab.ae.utexas.edu

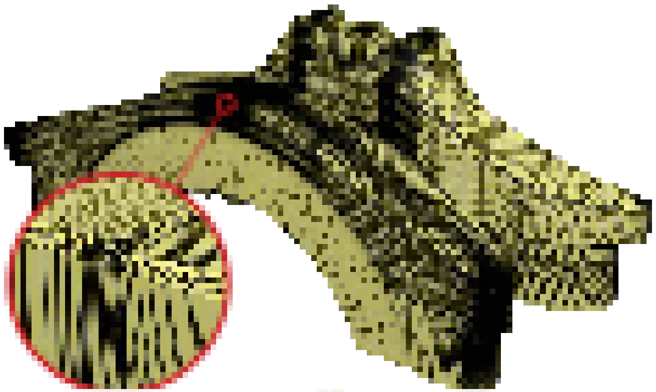
Automatically repairing geometric issues has proven to be a complicate



self-intersections



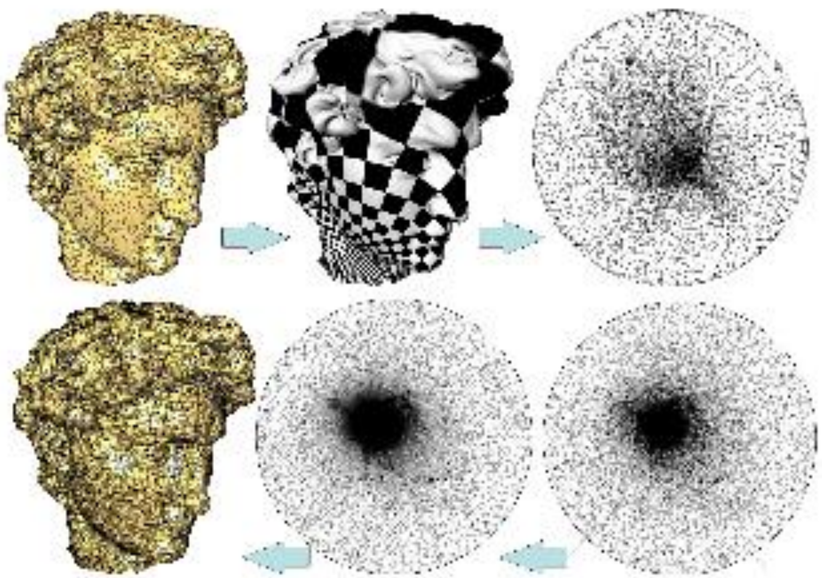
holes



bad quality

Surface meshing and remeshing

Figure 10.1: Meshes: Irregular, semi-regular and regular.



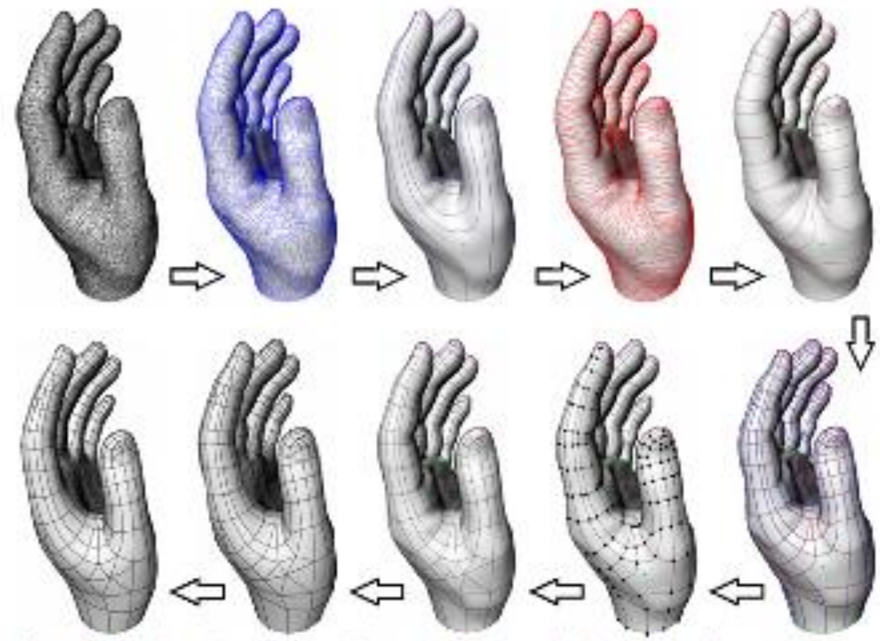


Figure 10.9: Anisotropic remeshing: From an input triangulated geometry, the curvature tensor field is estimated, then smoothed, and its umbilics are deduced (colored dots). Lines of curvatures (following the principal directions) are then traced on the surface, with a local density guided by the principal curvatures, while usual point-sampling is used near umbilic points (spherical regions). The final mesh is extracted by subsampling, and conforming-edge insertion. The result is an anisotropic mesh, with elongated quads aligned to the original principal directions, and triangles in isotropic regions.

Courtesy P. Alleiz

Mesh quality 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.

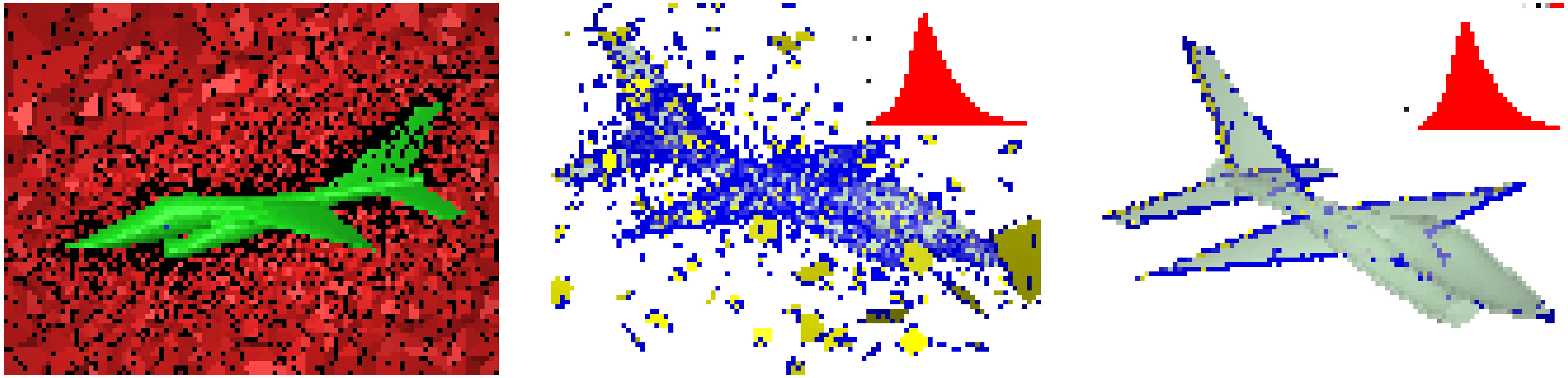
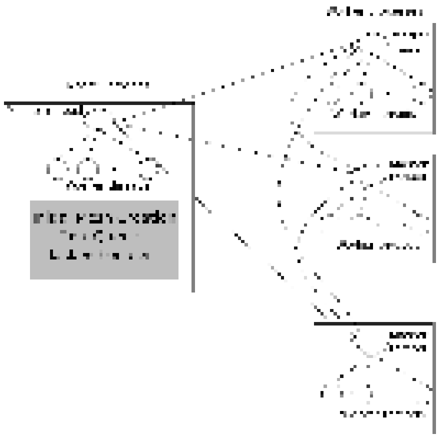
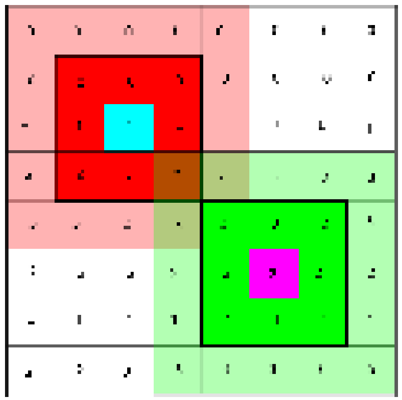
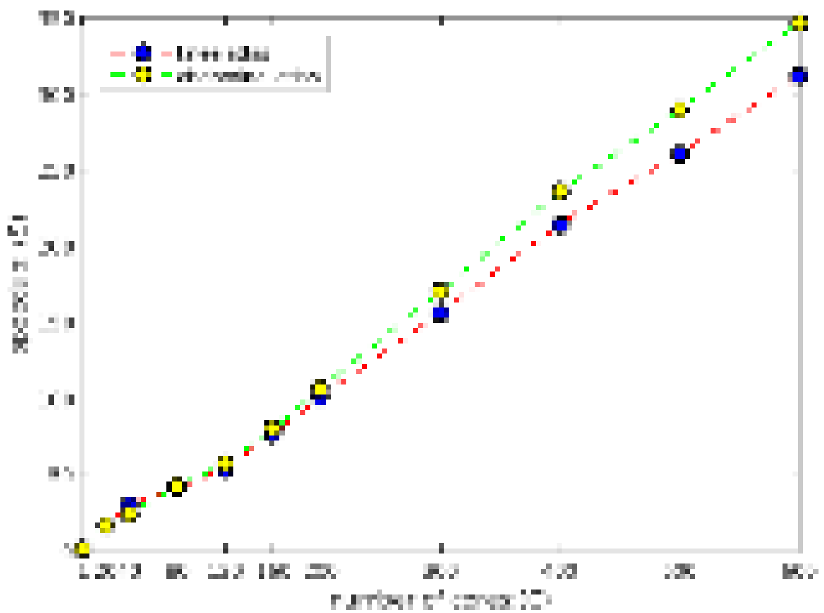


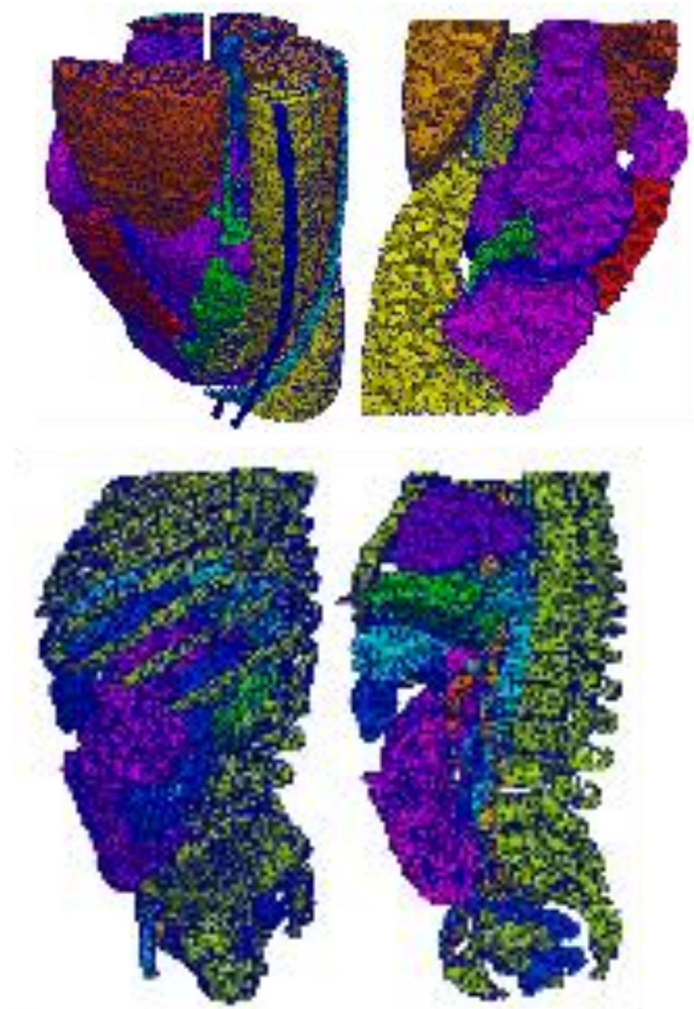
Figure: Left: A tet mesh from constrained Delaunay refinement. Middle: A highlight of the bad quality tets from the left mesh. Right: A highlight of the bad quality tets after mesh improvement.

Parallel mesh generation







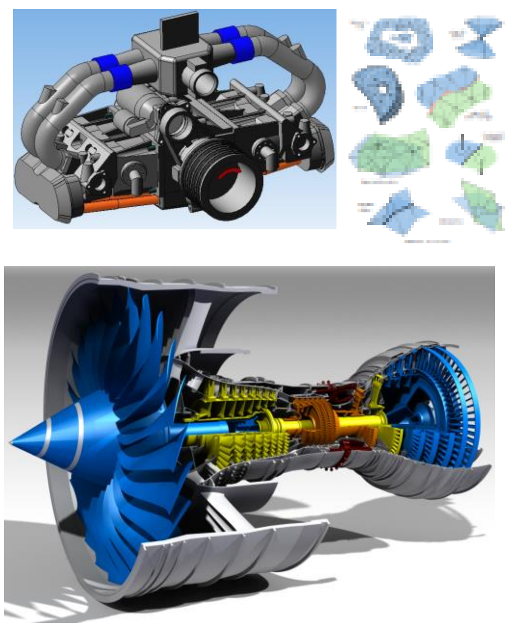


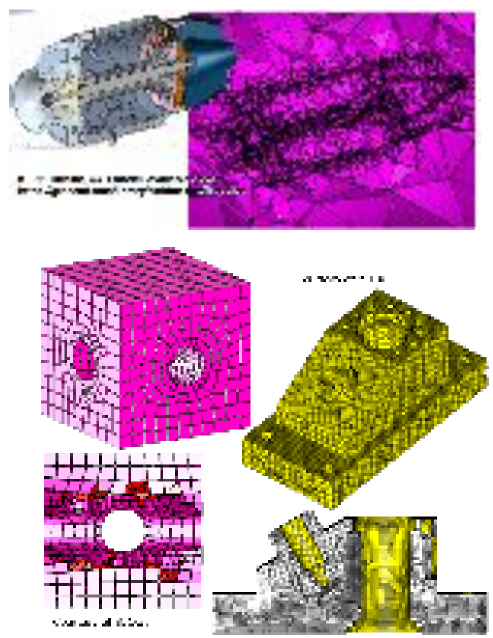
Courtesy Daming Feng et al 2016

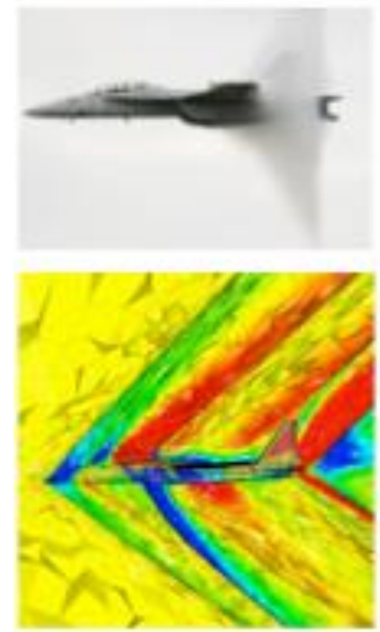
The Challenges

1. CAD geometry preparation, cleaning.
2. 3d surface and volume mesh generation.
3. Mesh adaptation, anisotropic meshes.

Automation, Robustness, Efficiency, …



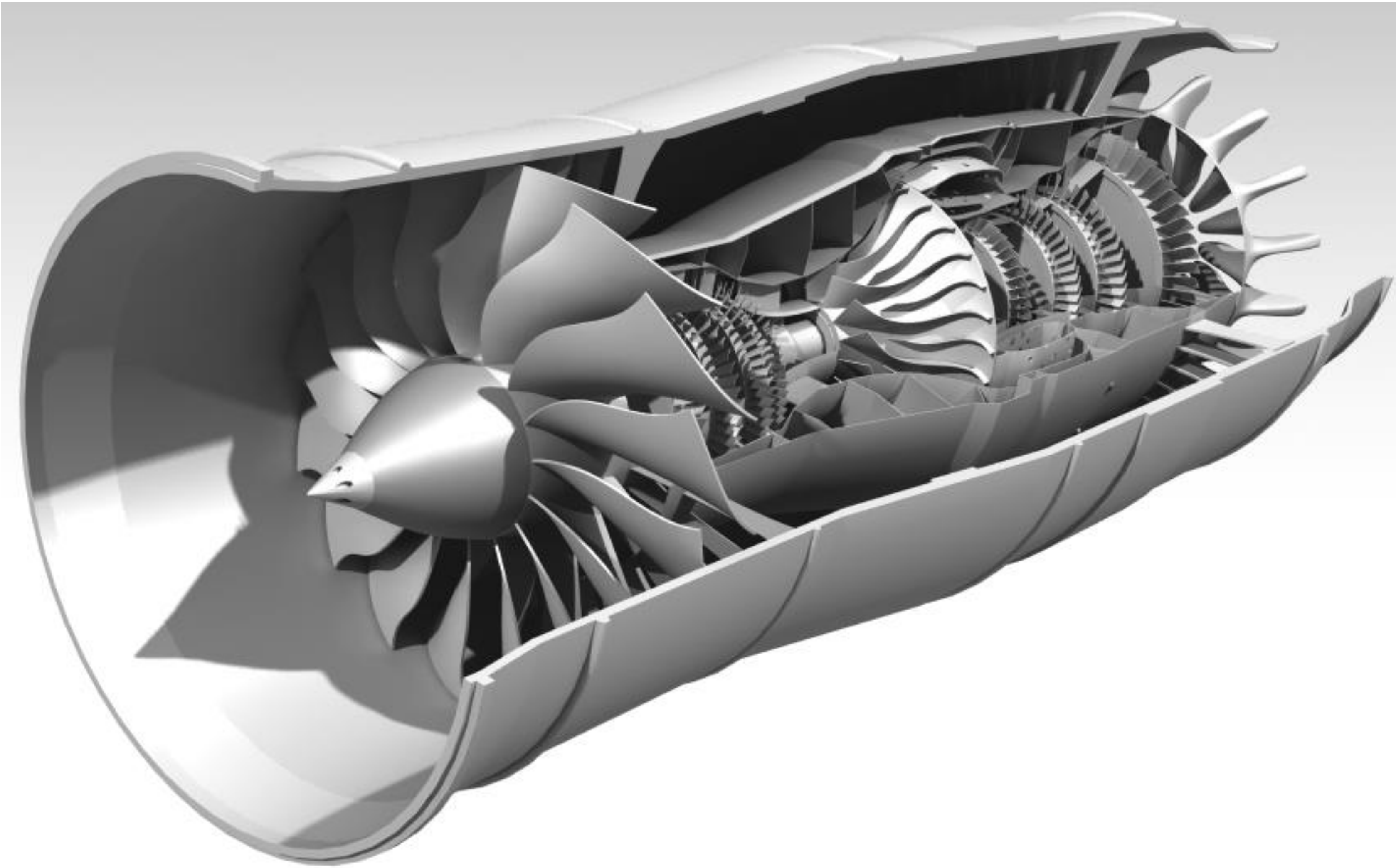




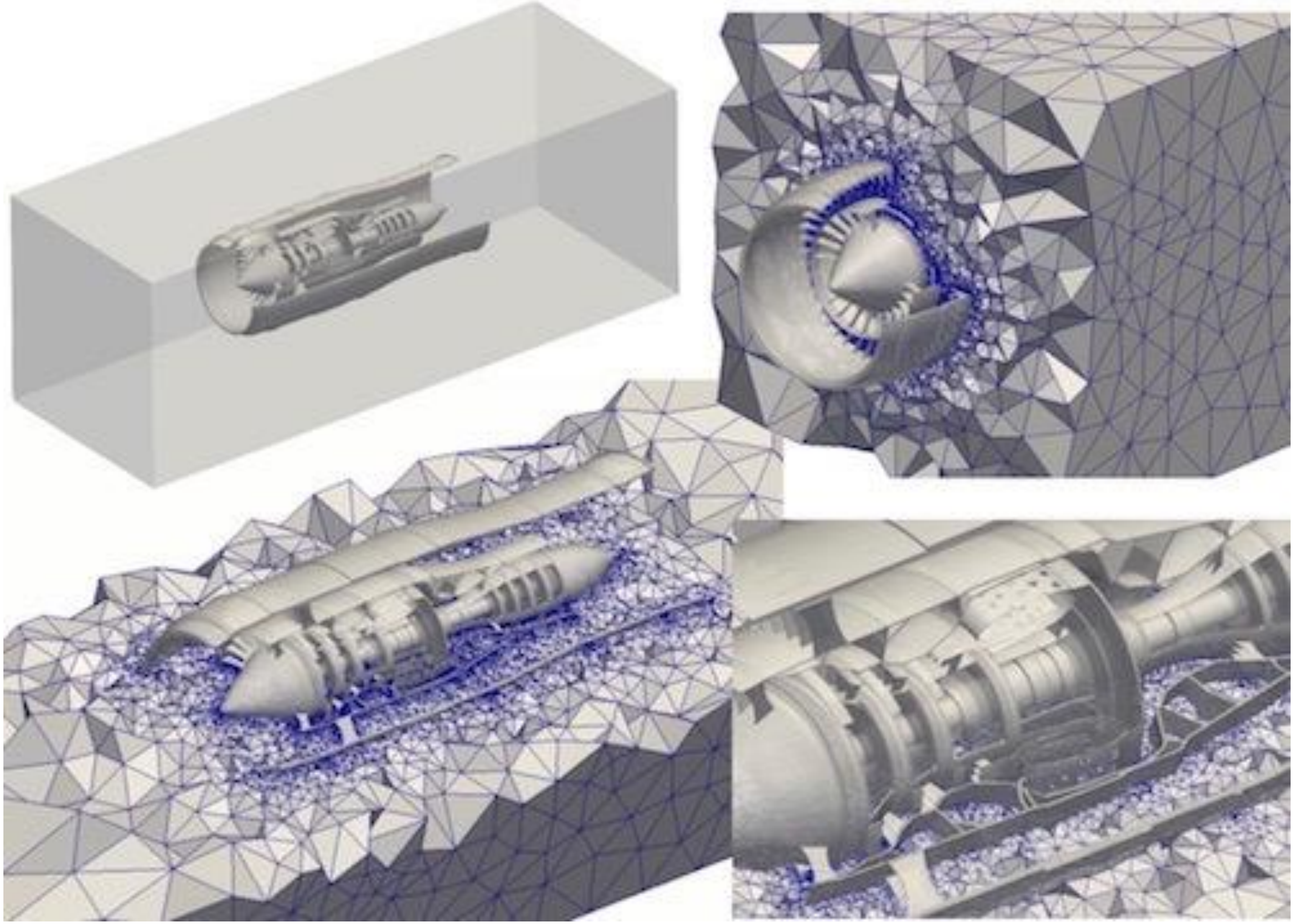
Images from Adrien Loiselle s Phd

Thank you for your attention

A CAD model of turbine



A 3d mesh of the turbine model



A numerical solution of the compressible Navier-Stoke equation