On automatic robust and efficient mesh generation and adaptation for EDA

Hang Si

Tetrahedron VII Workshop
Barcelona, Spain
Oct. 09-11, 2023

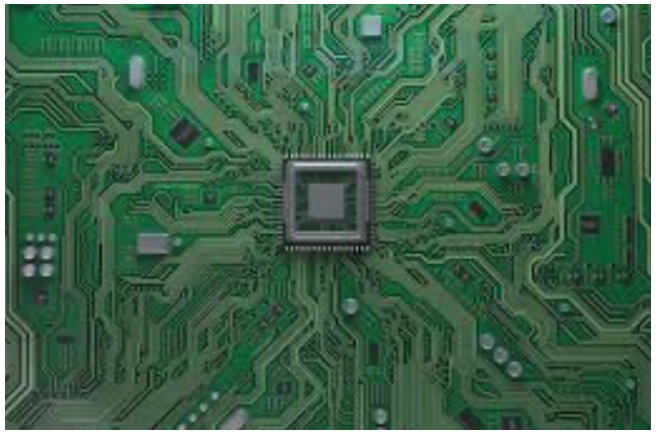
cadence

Meshing in EDA

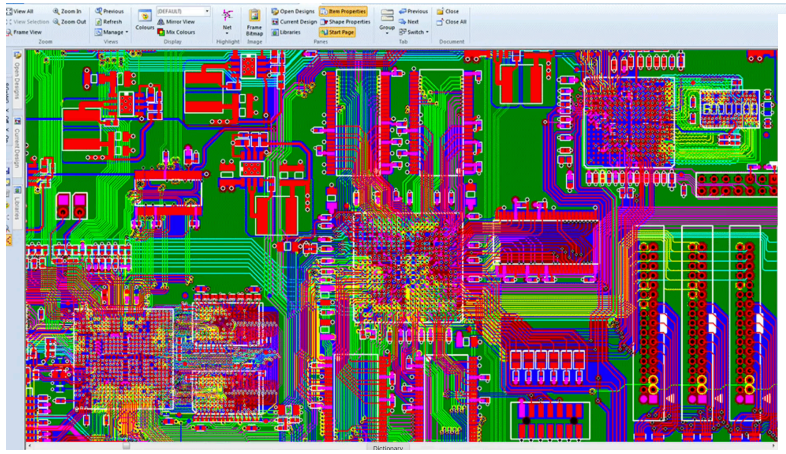
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

EDA (Electronic Design Automation) and EDA softwares

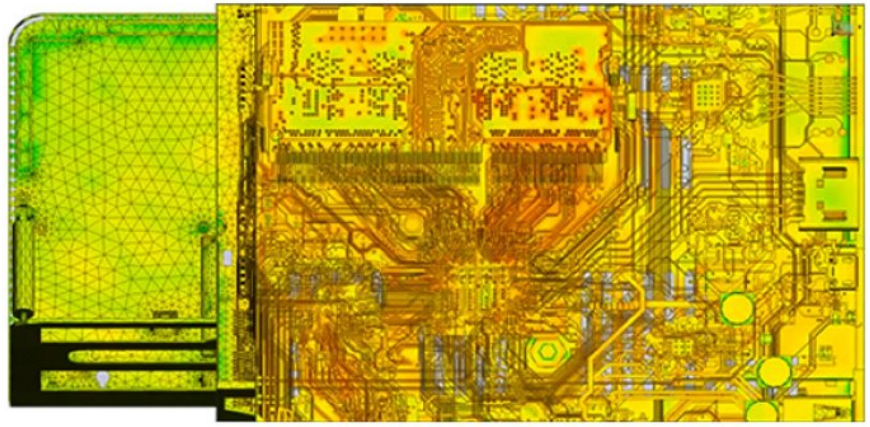
- EDA is a category of software tools for designing electronic systems such as integrated circuits (ICs) and printed circuit boards (PCBs).
- Cadence develops software, hardware and solutions for the computing, chip, 5G communications, automotive and aerospace industries.



Michael Green
What is chip design?
https://magreen.medium.com



Asianometry: EDA Software, designing billions of circuits with code
https://www.youtube.com/watch?v=ihz2WY-E2C8



Cadence, Clarity

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Outline

- Meshing in EDA
- Delaunay-based 2d/3d mesh generation
- Surface mesh generation
- Remarks and outlook

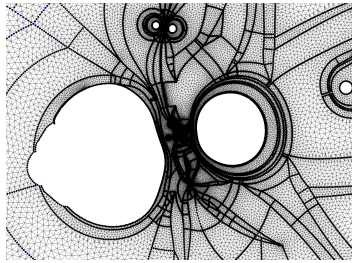
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

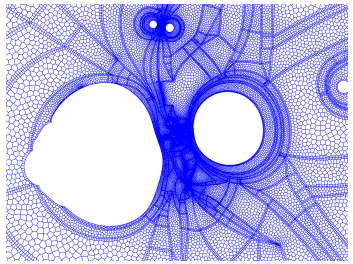
Delaunay-based 2d/3d mesh generation

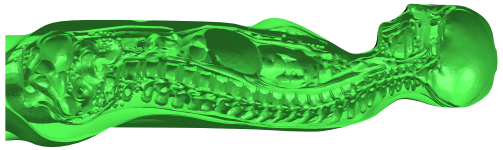
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

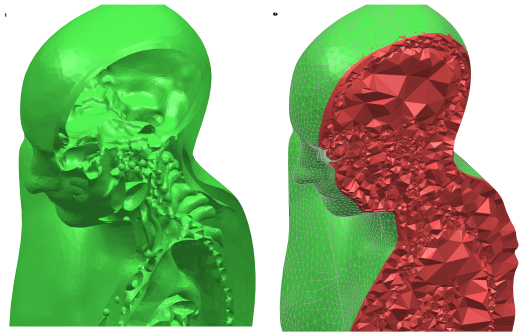
Delaunay and Voronoi meshes

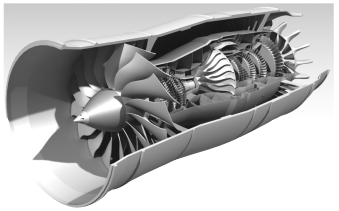
- Nice geometric properties: nearest neighbors, MaxMin angle, in relation to convex polytope theory
- Dual of Voronoi diagrams (crucial property in FEM and FVM methods)
- High mesh quality (good angles) and regularity.
- Fast algorithms have been developed
- Robust (open source) softwares: Triangle, Detri2, TetGen

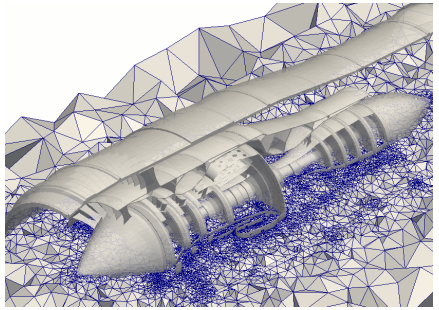










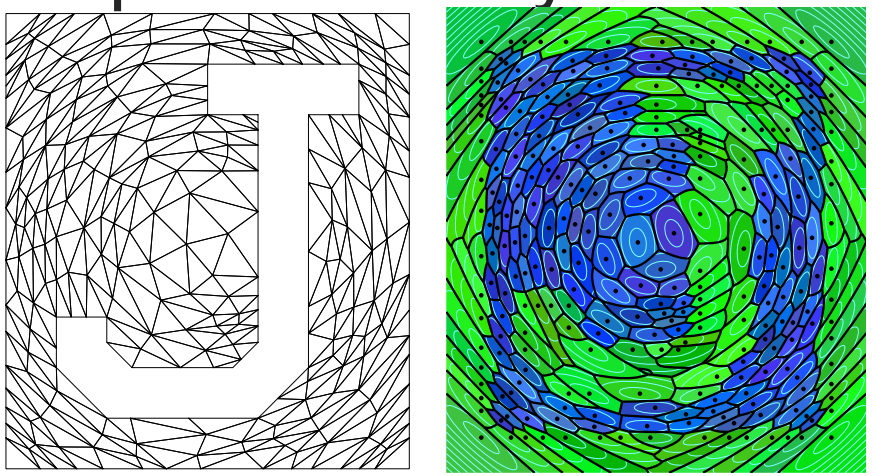


Georgy F. Voronoy (1868-1908)

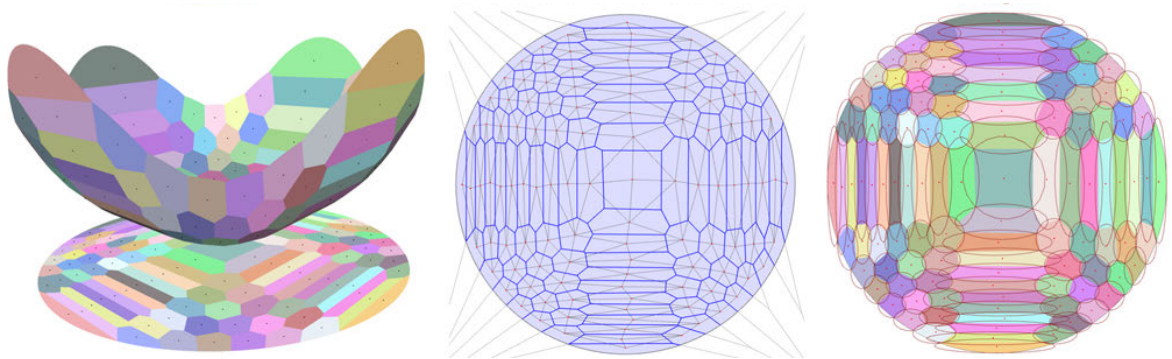
Boris N. Delaunay (1890-1980)

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

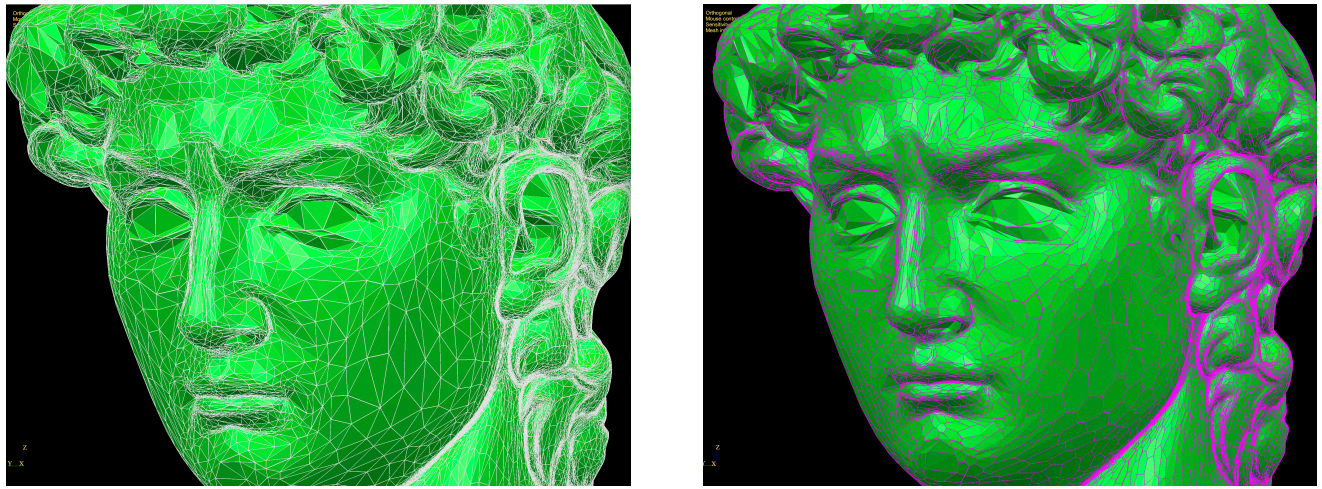
Anisotropic Delaunay and Voronoi meshes



An anisotropic Delaunay mesh wrt a metric field
Labelle & Shewchuk (not always exists)



Optimal Voronoi tessellation, Budninskiy et al, TOG 2016
(Only valid for convex functions)



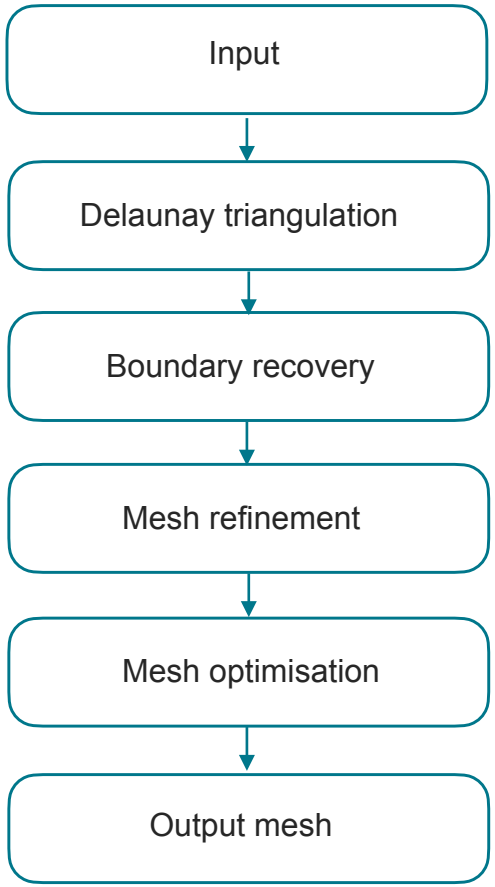
Anisotropic Delaunay surface mesh

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

A Delaunay-based Mesh Generation Framework (2d/3d)

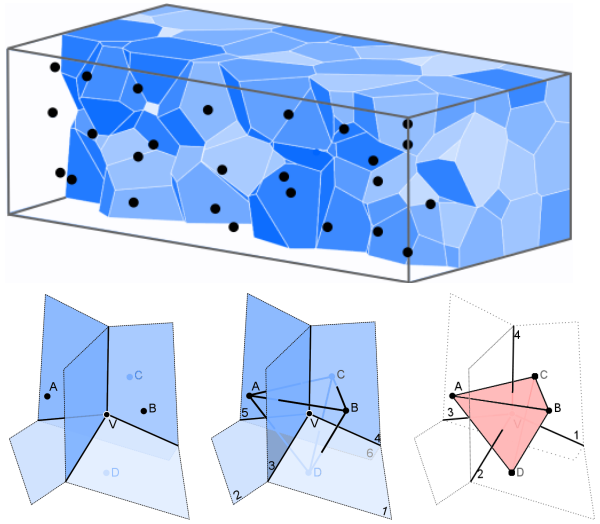
Input: a set of points, line segments, facets

1. Delaunay triangulation: create the DT from input points.
2. Boundary Recovery: insert all input segments and facets to create a constrained (Delaunay) triangulation (CDT).
3. Mesh refinement: insert new points into the CDT according to (user-defined) element shape and size requirements.
4. Mesh optimisation: improve the mesh quality by vertex smoothing, edge/face swaps, and vertex insertions/deletions

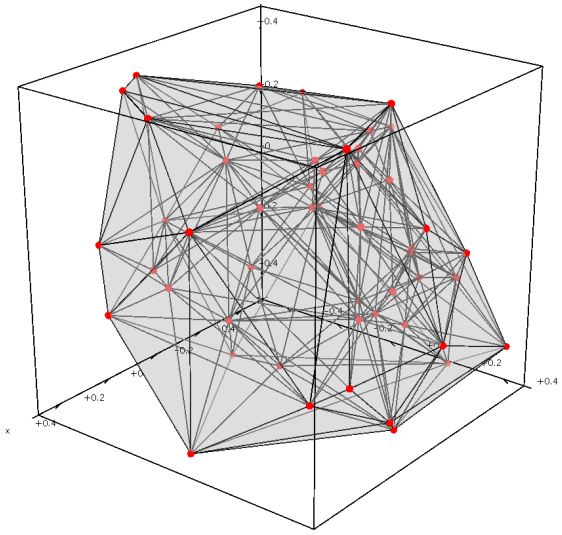


© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

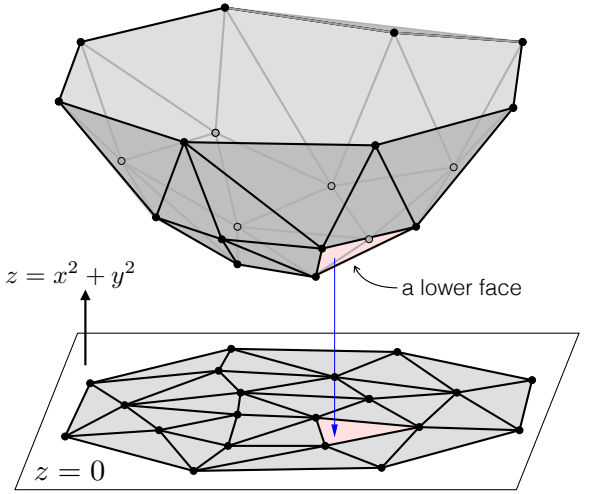
Delaunay triangulations



Dual of Voronoi diagrams



empty circumsphere property



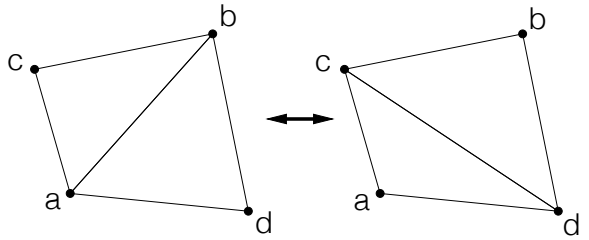
projections of convex hulls

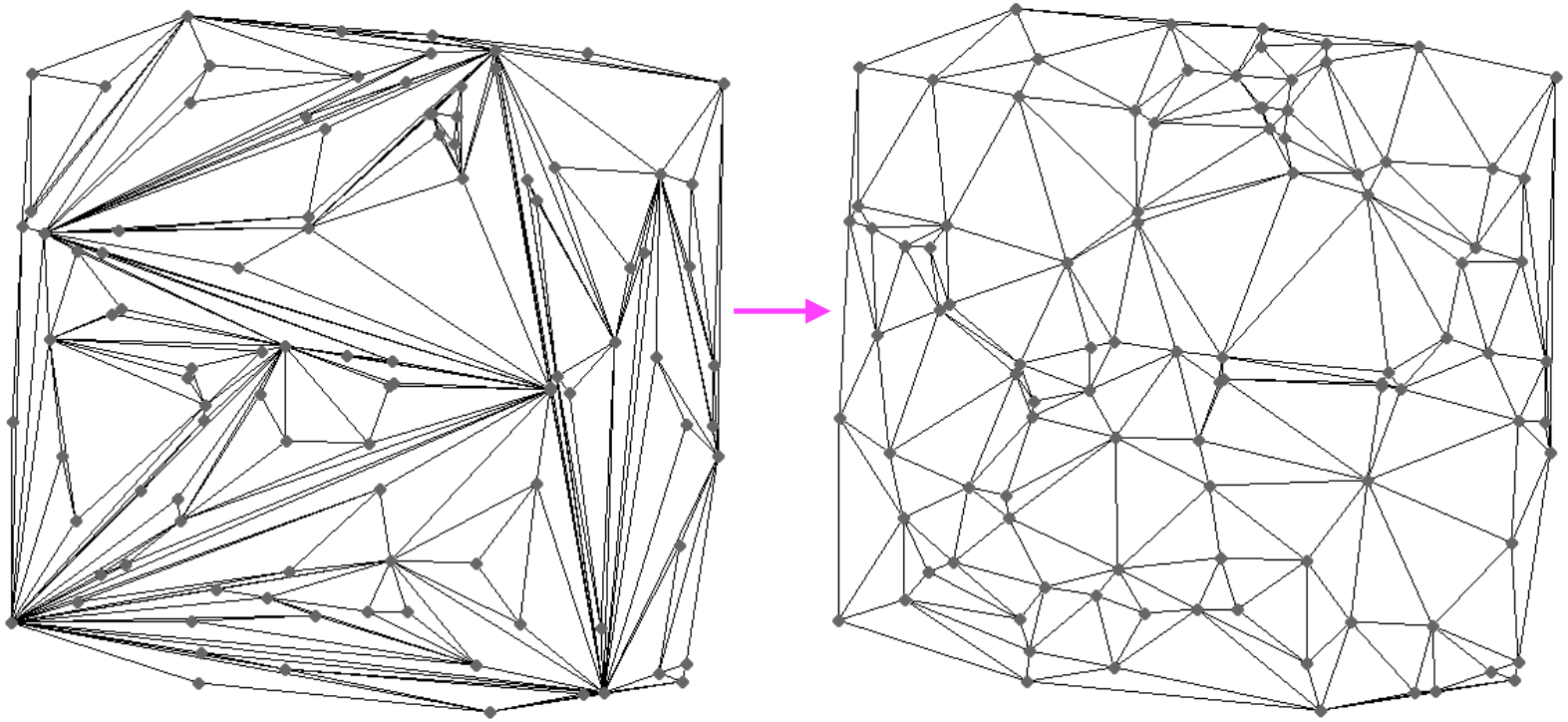
Figure from J. Pellerin's thesis

Lawson's Flip Algorithm [1972, 1977]

- 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 **ab** $\in \mathcal{T}$
    flip **ab**;
end while





© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Correctness and runtime

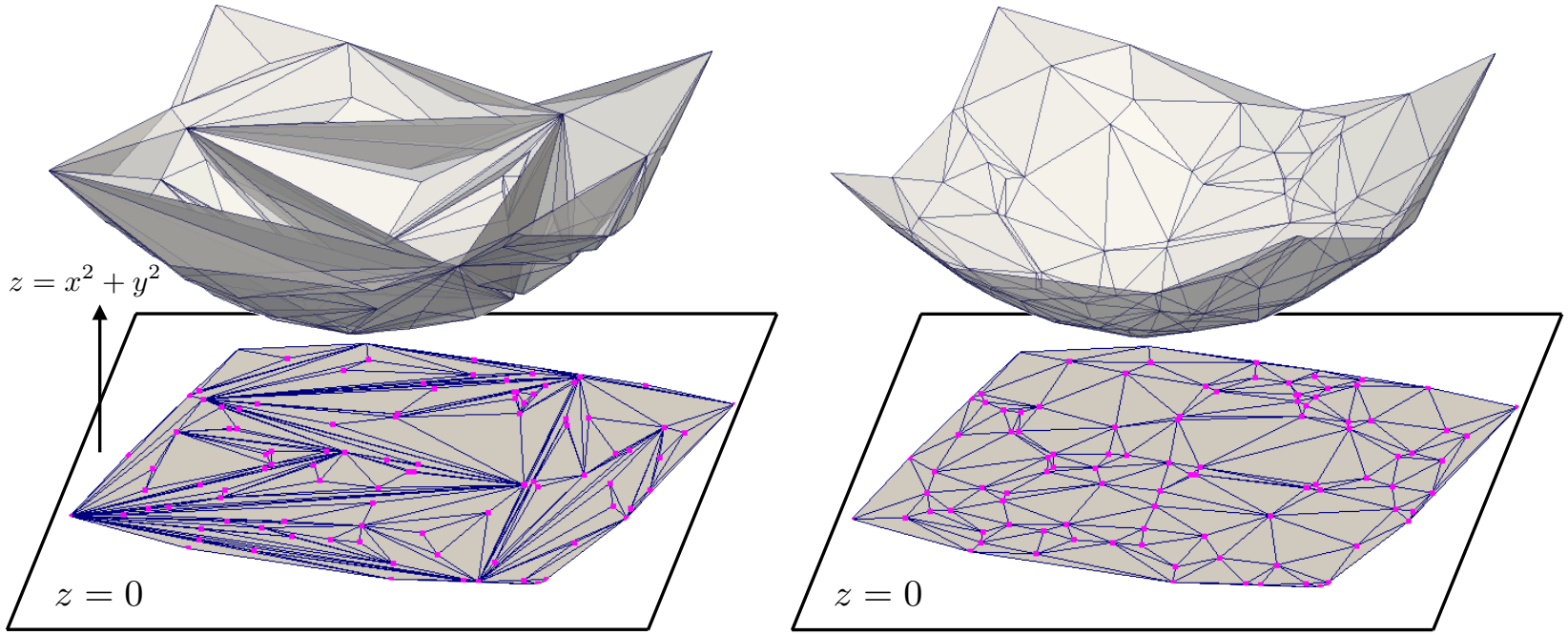


FIGURE 17. Top: Bottom: The lifted view of the Lawson's flip algorithm which transforms a non-convex surface (left) in 3d into a convex one (right).

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Flip graph

2.4. **The (undirected) flip graph.** One can use flips to traverse the set of all triangulations of a point set S. We can form a *flip-graph* $\mathcal{G}$ of S. Each triangulation is a node of $\mathcal{G}$, and each edge of $\mathcal{G}$ between two nodes u and v means there is a flip that changes the triangulation u to v. Figure 19 shows an example. The termination of Lawson's flip algorithm implies that the flip-graph for any point set in the plane is connected, i.e., one can go from any triangulation of S to any other triangulation.

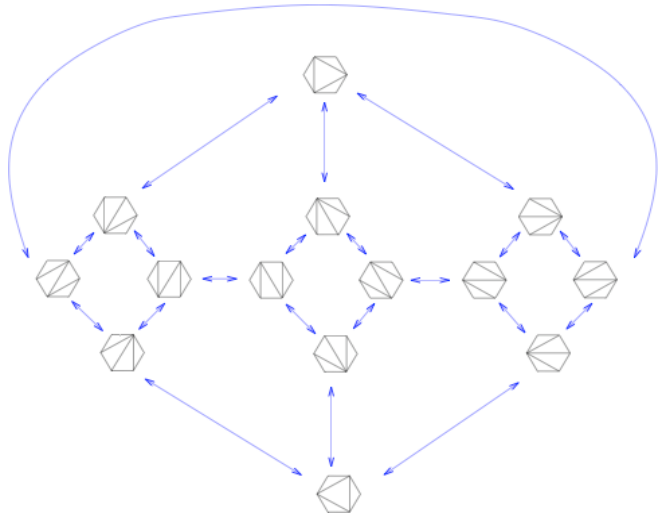


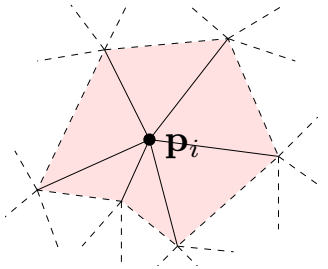
FIGURE 19. The flip graph of a set of the vertex set of a convex 6-gon.

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

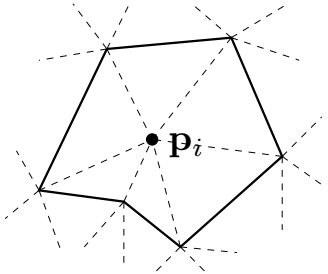
Incremental flip algorithm

Algorithm: IncrementalFlip($S = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$)
Input: a sequence S of n points in $\mathbb{R}^2$;
Output: the Delaunay triangulation $\mathcal{D}$ of S;
1 initialize $\mathcal{D}_0$ with only one larger triangle $t_{\mathbf{xyz}}$;
2 **for** $i = 1$ to n **do**
3 find the triangle $\tau \in \mathcal{D}_{i-1}$ containing $\mathbf{p}_i$;
4 insert $\mathbf{p}_i$ by a 1-3 flip;
5 initial the stack L with link edges of $\mathbf{p}_i$;
6 LawsonFlip(L);
7 **endfor**
8 remove all triangles containing $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ from $\mathcal{D}_n$;

FIGURE 22. The incremental-flip algorithm.



the star of $\mathbf{p}_i$



the link of $\mathbf{p}_i$

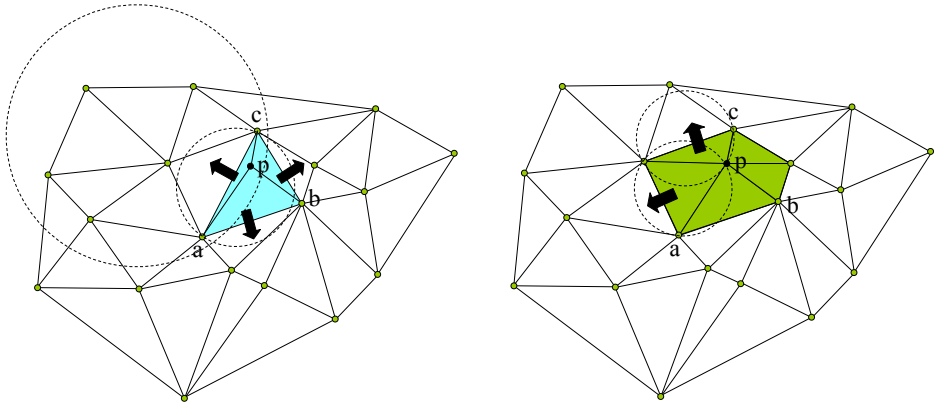
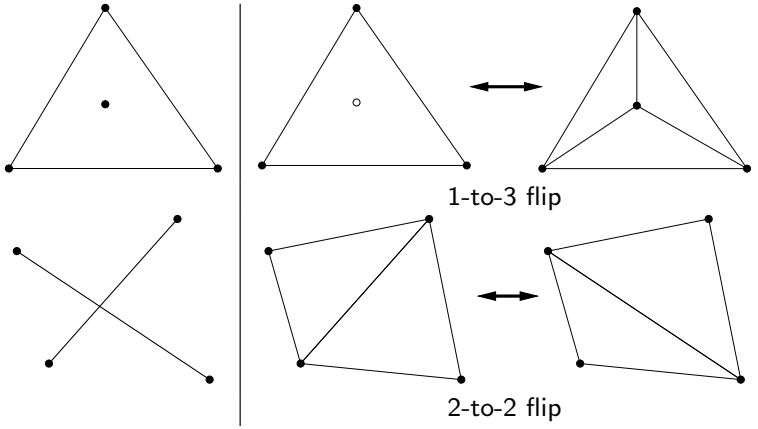


FIGURE 23. Recovery Delaunay property by the Lawson's flip algorithm.

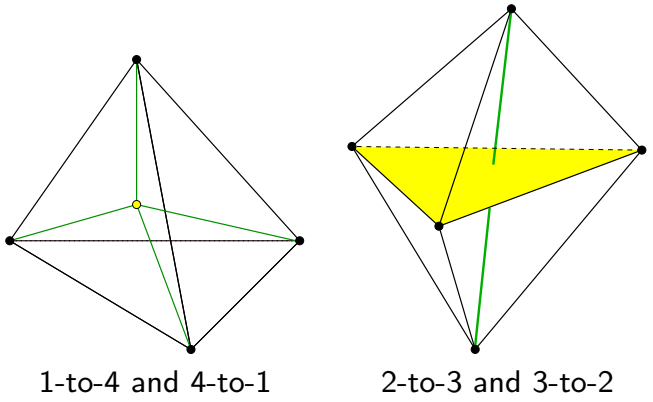
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Flips in any dimension

- **Radon's partitions[Radon 1921]**: Any set of $d+2$ points in $\mathbb{R}^d$ can be partitioned into two disjoint sets whose convex hulls intersect.



2d

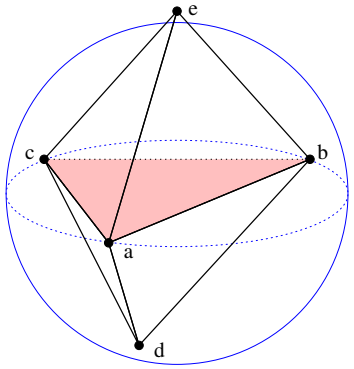


3d

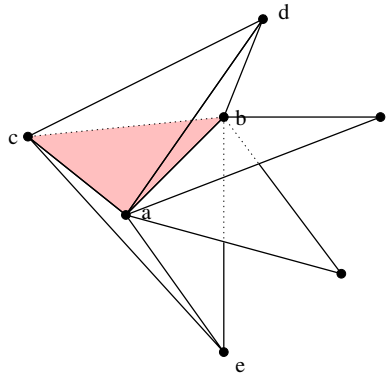
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Flips in 3d

Lemma [Joe 1989]: There exists a 3d non-Delaunay triangulation which contains a cycle of non-locally Delaunay and unflippable faces.



a locally Delaunay face (shaded)



face (a,b,c) is unflippable

TABLE 1
Vertex coordinates.

Index	x	y	z
1	0.054	0.099	0.993
2	0.066	0.756	0.910
3	0.076	0.578	0.408
4	0.081	0.036	0.954
5	0.082	0.600	0.726
6	0.085	0.327	0.731
7	0.123	0.666	0.842
8	0.161	0.303	0.975

TABLE 2
Pseudo-locally optimal non-Delaunay triangulation (left) and Delaunay triangulation (right). A tetrahedron is described by its four vertex indices.

1	2	3	5	1	2	3	5
1	2	4	6	1	2	5	6
1	2	4	7	1	2	6	8
1	2	5	6	1	3	4	6
1	2	7	8	1	3	5	6
1	3	4	6	1	4	6	8
1	3	5	6	2	3	5	7
1	4	7	8	2	5	6	8
2	3	5	7	2	5	7	8
2	4	5	6	3	4	6	8
2	4	5	7	3	5	6	7
3	4	6	8	3	6	7	8
3	5	6	7	5	6	7	8
3	6	7	8				
4	5	6	8				
4	5	7	8				
5	6	7	8				

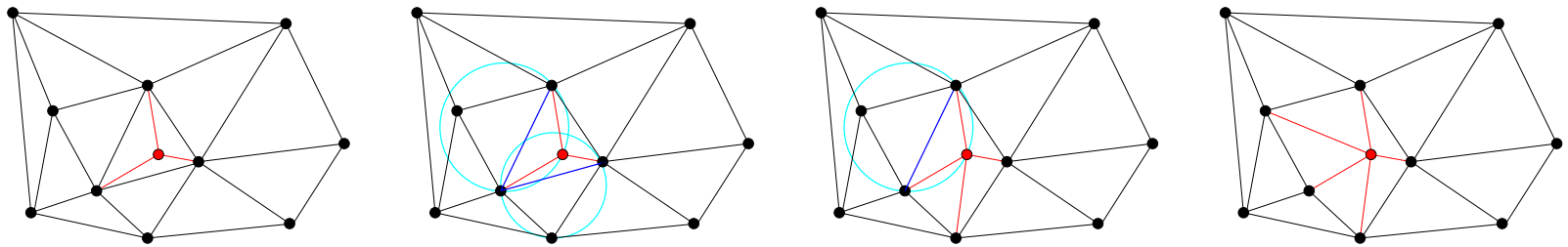
Joe's example [Joe 1989]

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Incremental flip in 3d

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

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



Edelsbrunner, H. & Shah, N. R. Incremental topological flipping works for regular triangulations, Algorithmica, 1996, 15, 223-241

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Implementation choices

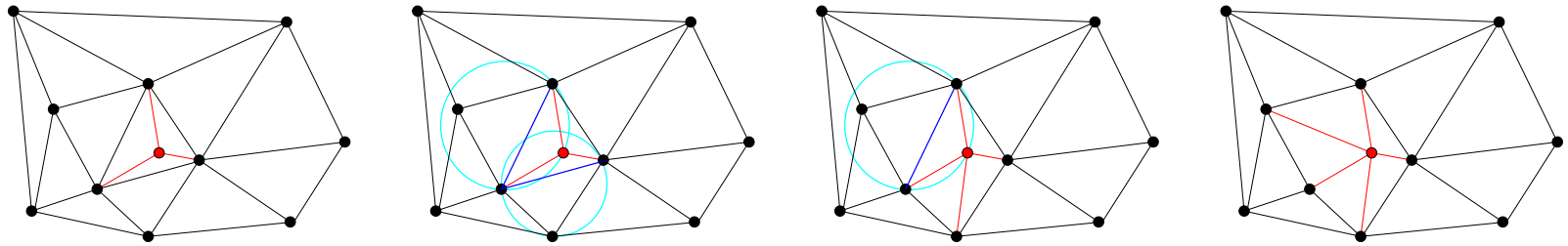
- Let $S = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ be a finite set of points in $\mathbb{R}^3$. And we assume S is in general position. ← **sort the input points by BRIO + Hilbert order**
- Let $[\mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}]$ be a sufficiently large tetrahedron that contains all points of S.

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

← no need of this step when using an infinite vertex

← **Use Bowyer-Watson cavity algorithm**

Use filtered robust predicates to do point orientation3d and point-in-sphere tests

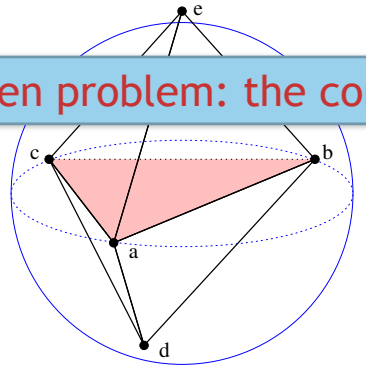


Célestin Marot, Jeanne Pellerin, and Jean-François Remacle. One machine, one minute, three billion tetrahedra. *International Journal for Numerical Methods in Engineering*, 117(9):967–990, 2019.

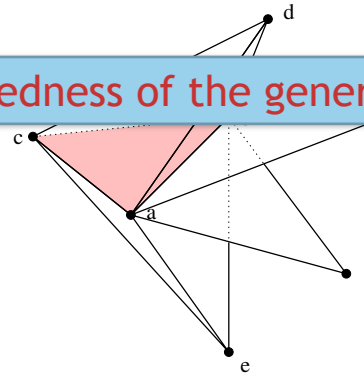
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Flips in 3d

Lemma [Joe 1989]: There exists a 3d non-Delaunay triangulation which contains a cycle of non-locally Delaunay and unflippable faces.



a locally Delaunay face (shaded)



face (a,b,c) is unflippable

Open problem: the connectedness of the general flip graphs in 3d and 4d.

TABLE 1
Vertex coordinates.

Index	*x*	*y*	*z*
1	0.054	0.099	0.993
2	0.066	0.756	0.910
3	0.076	0.578	0.408
4	0.081	0.036	0.954
5	0.082	0.600	0.726
6	0.085	0.327	0.731
7	0.123	0.666	0.842
8	0.161	0.303	0.975

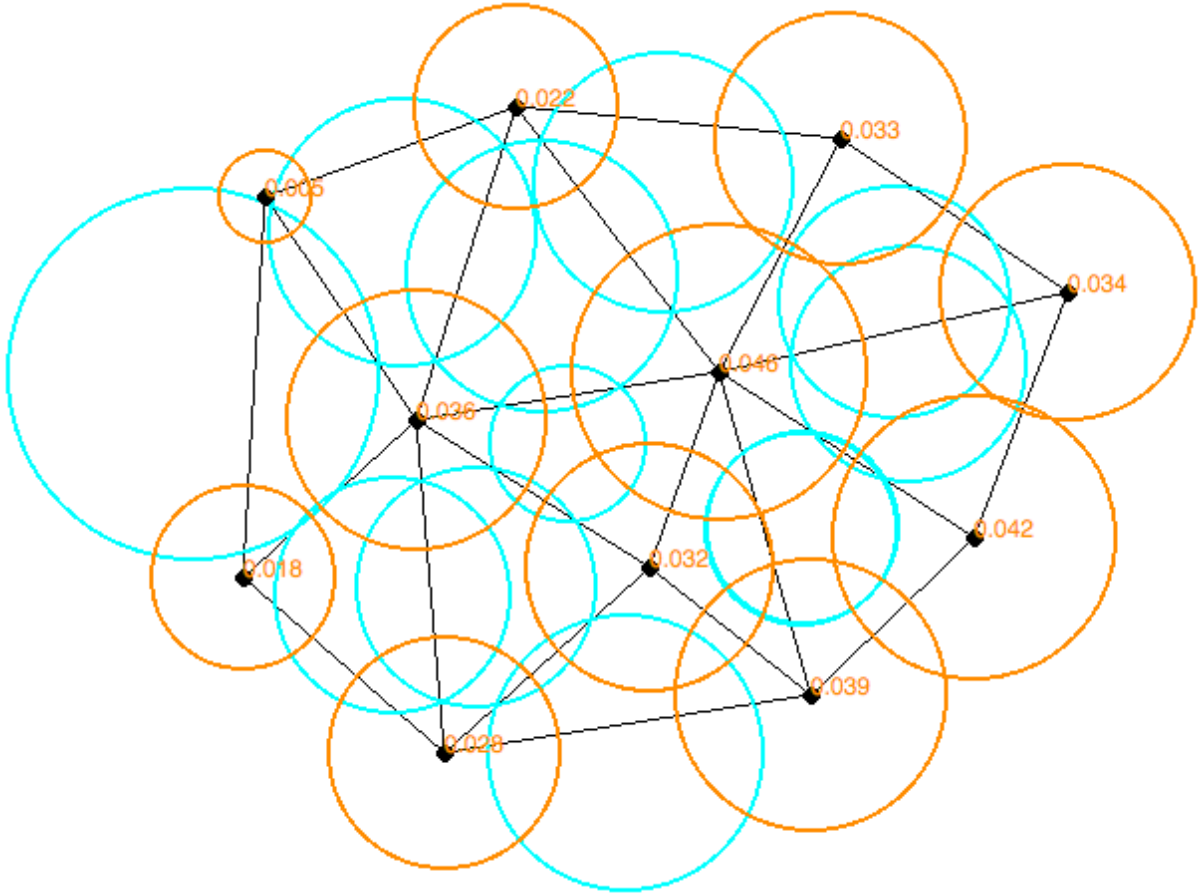
Delaunay triangulation (right). A tetrahedron is described by its four vertex indices.

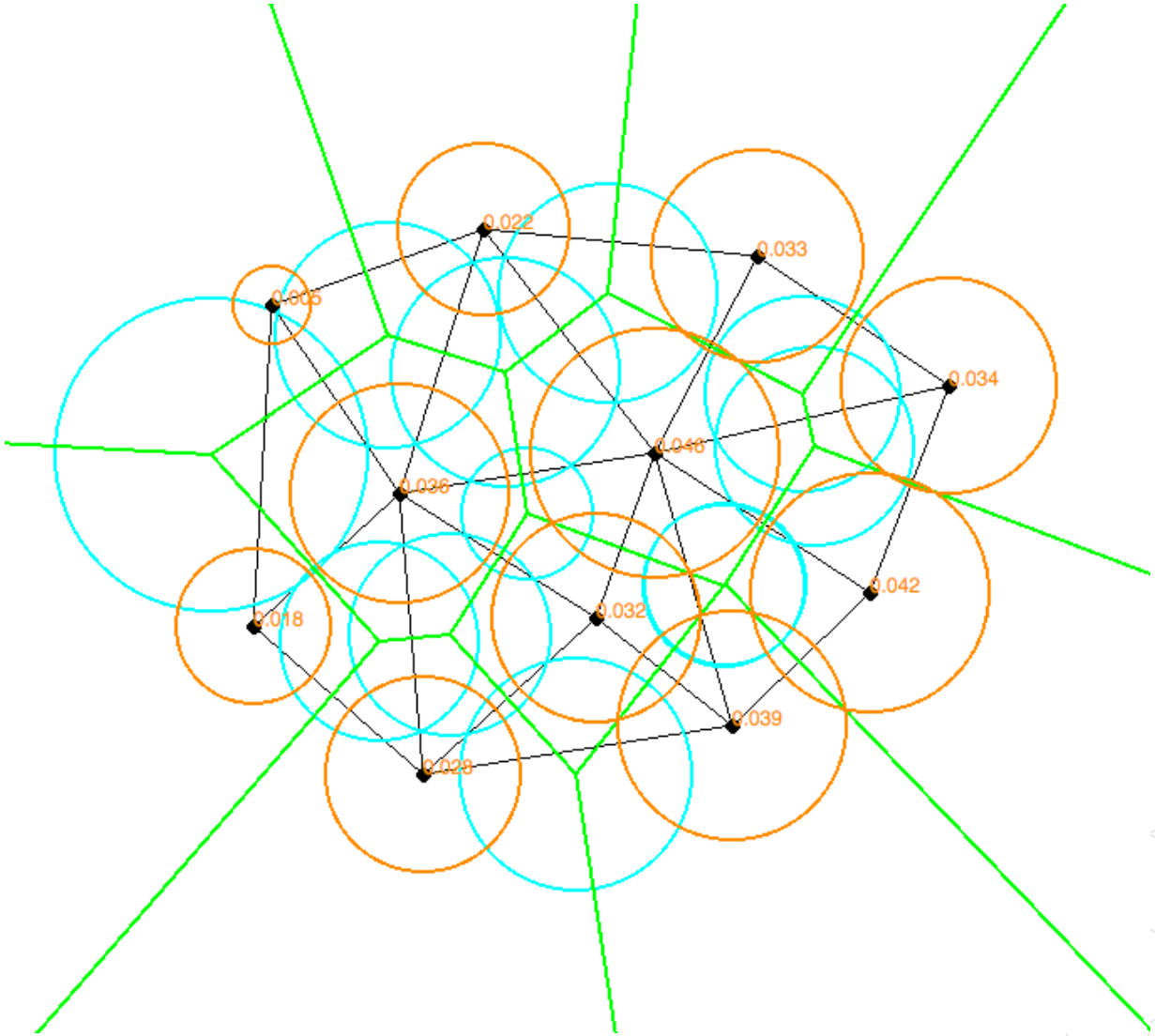
1	2	3	5	1	2	3	5
1	2	4	6	1	2	5	6
1	2	4	7	1	2	6	8
1	2	5	6	1	3	4	6
1	2	7	8	1	3	5	6
1	3	4	6	1	4	6	8
1	3	5	6	2	3	5	7
1	4	7	8	2	5	6	8
2	3	5	7	2	5	7	8
2	4	5	6	3	4	6	8
2	4	5	7	3	5	6	7
3	4	6	8	3	6	7	8
3	5	6	7	5	6	7	8
3	6	7	8				
4	5	6	8				
4	5	7	8				
5	6	7	8				

Joe's example [Joe 1989]

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Weighted Delaunay Triangulations

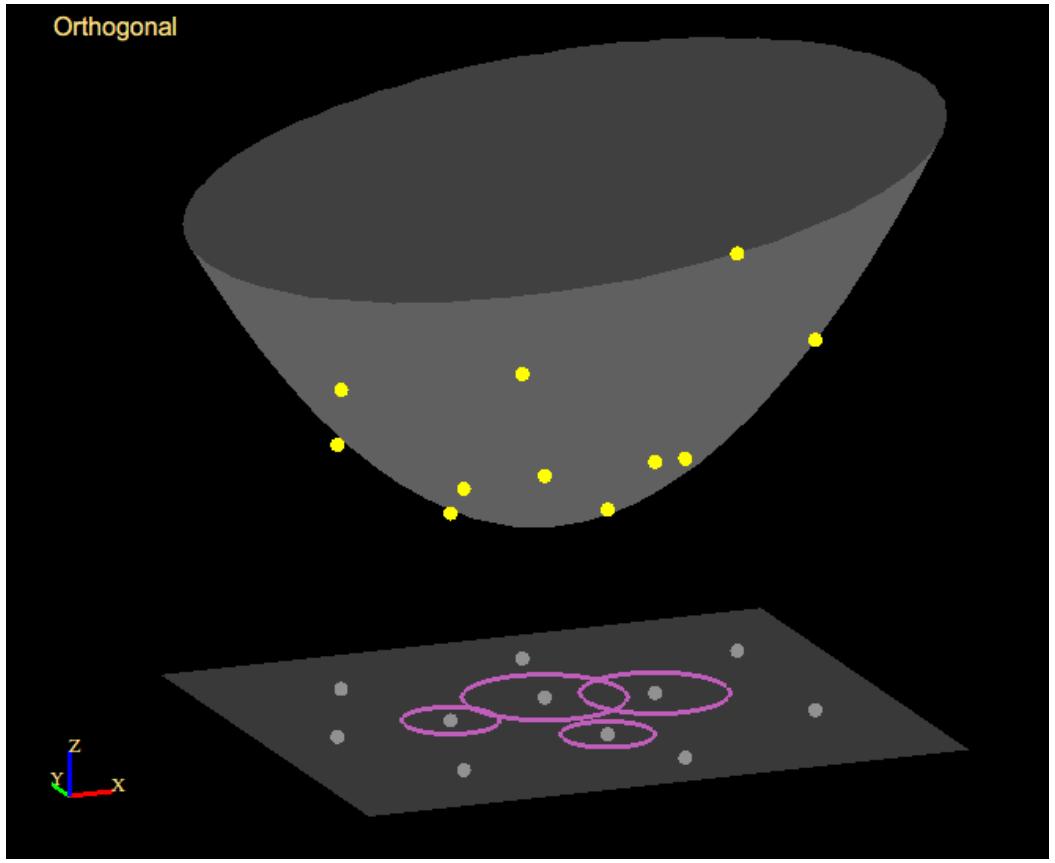


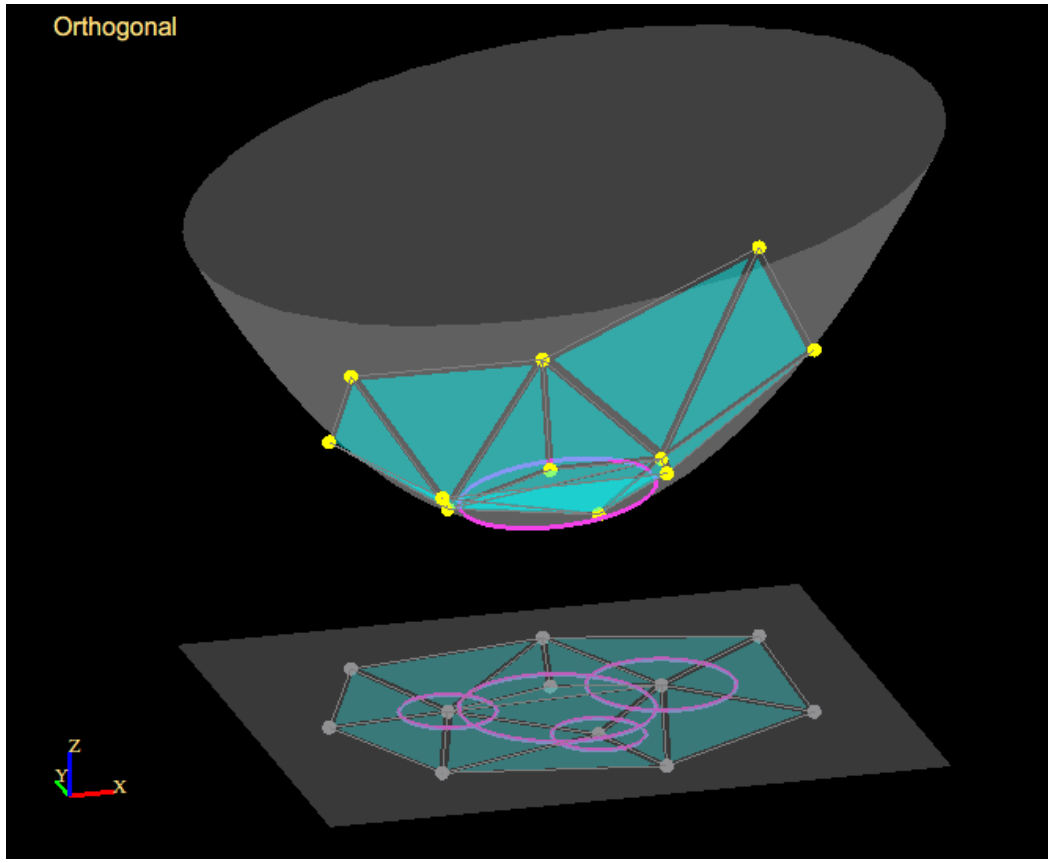


© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

cadence

Weighted Delaunay triangulations



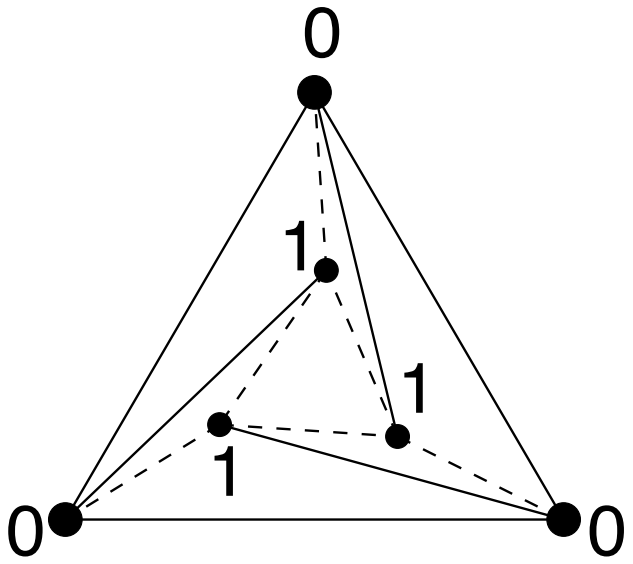


© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

cadence

Non-weighted Delaunay 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**].



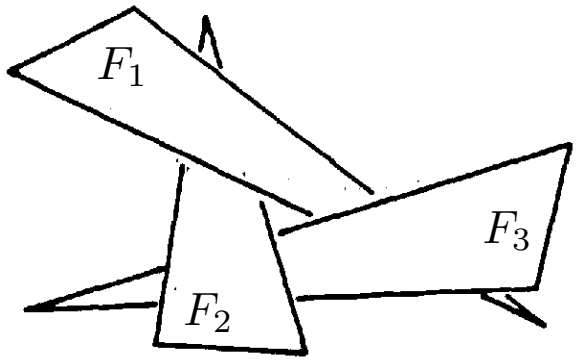
A non-regular triangulation

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

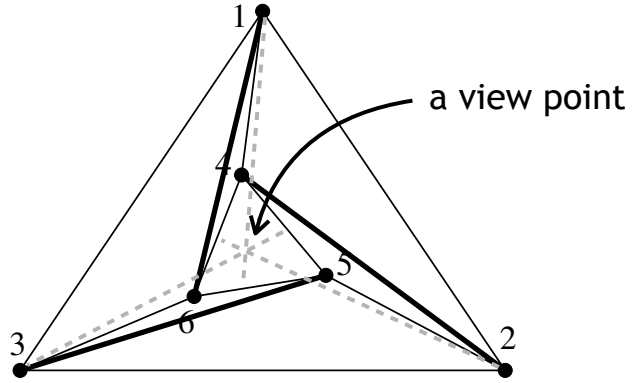
cadence

The acyclic theorem [Edelsbrunner 1990]

- The in_front/behind relation: Let **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* **x** if there is a ray L starting at **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 **x** in $\mathbb{R}^d$ is acyclic.



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



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

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

cadence

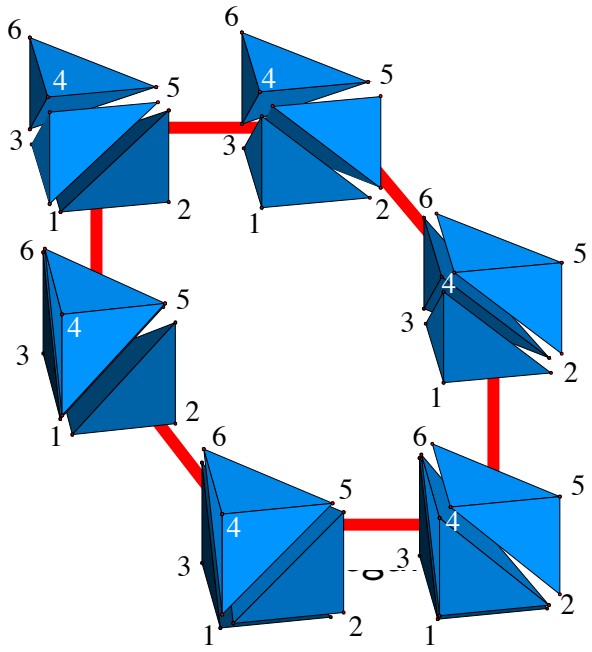
The Flip Graph of Regular Triangulations

- **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.



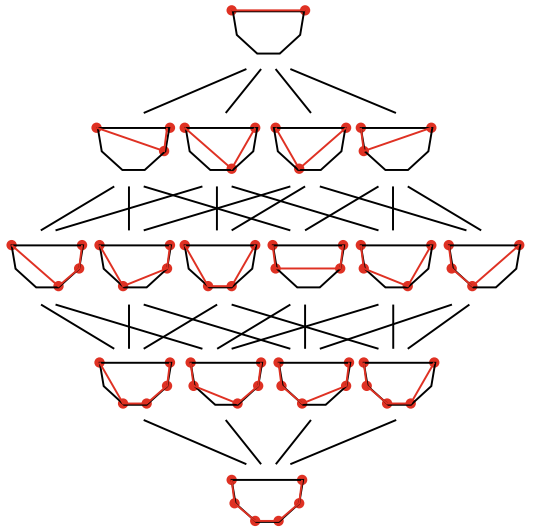
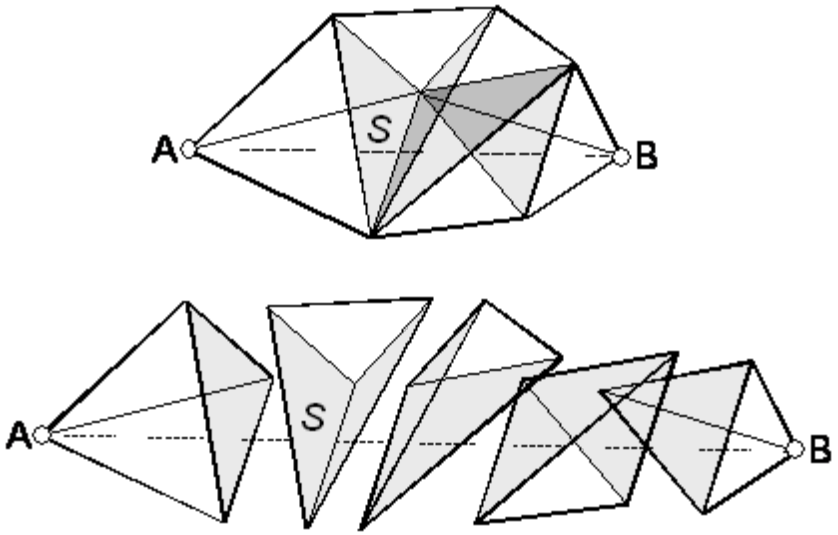


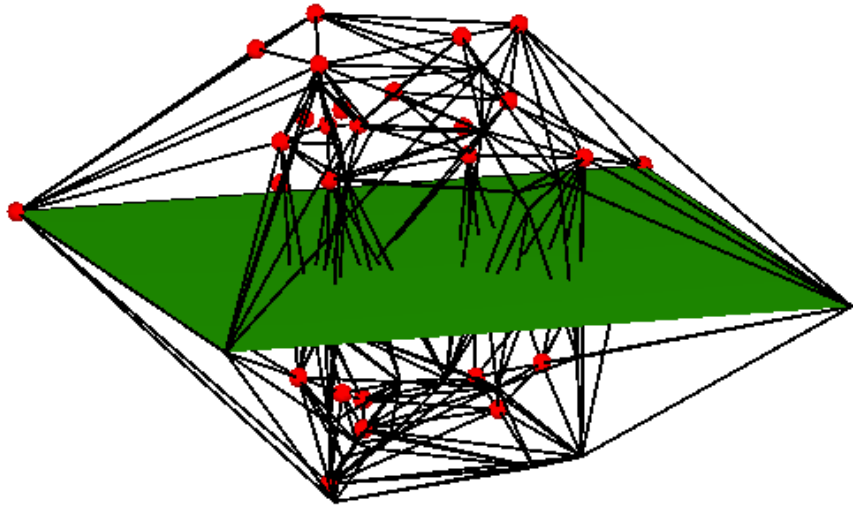
Figure 6.7: The height of a section defines a poset on all triangulations of $\mathbf{C}(6,1)$.

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Boundary Recovery



How to recover the edge AB?
images from [**Owen 1998**]

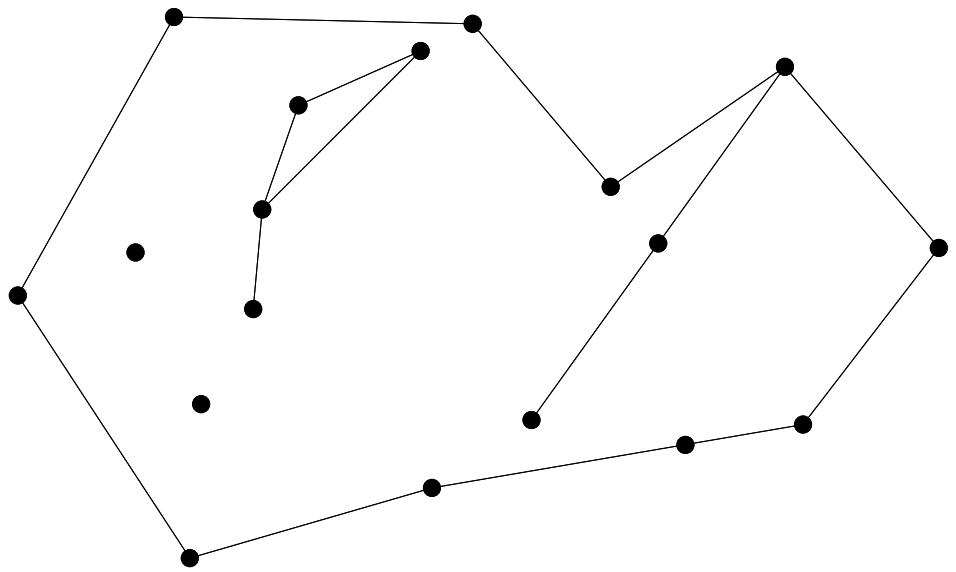


How to recover the rectangular face

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Planar straight-line graphs (PSLGs)

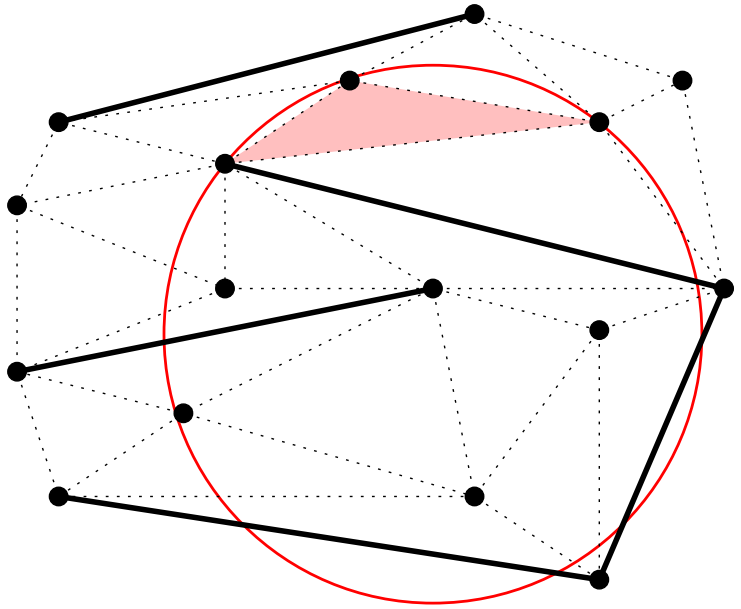
2.1. **Planar straight-line graphs.** Consider now the input is a finite set of points $S \subset \mathbb{R}^2$, together with a finite set of line segments, L, each connecting two points in S. We require any two line segments of L be disjointed or met at most in a common endpoint. We call $G = (S, L)$ a *planar straight-line graph* (PSLG), see Figure 2.1 Left for an example.



A PSLG (S, L)

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

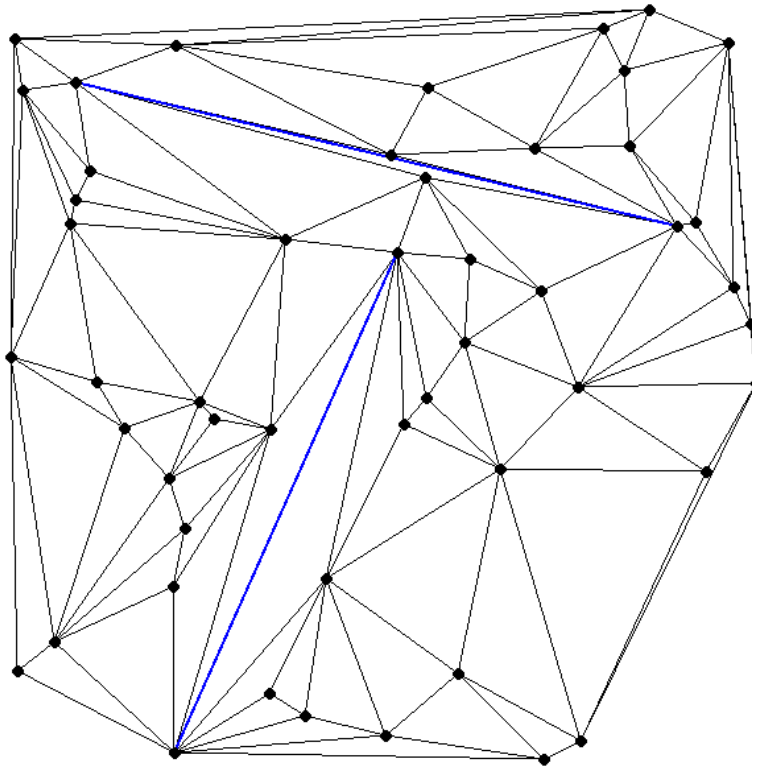
A constrained Delaunay triangulation (CDT) is a triangulation whose triangles are all constrained Delaunay [**Lee & Lin 1986**].

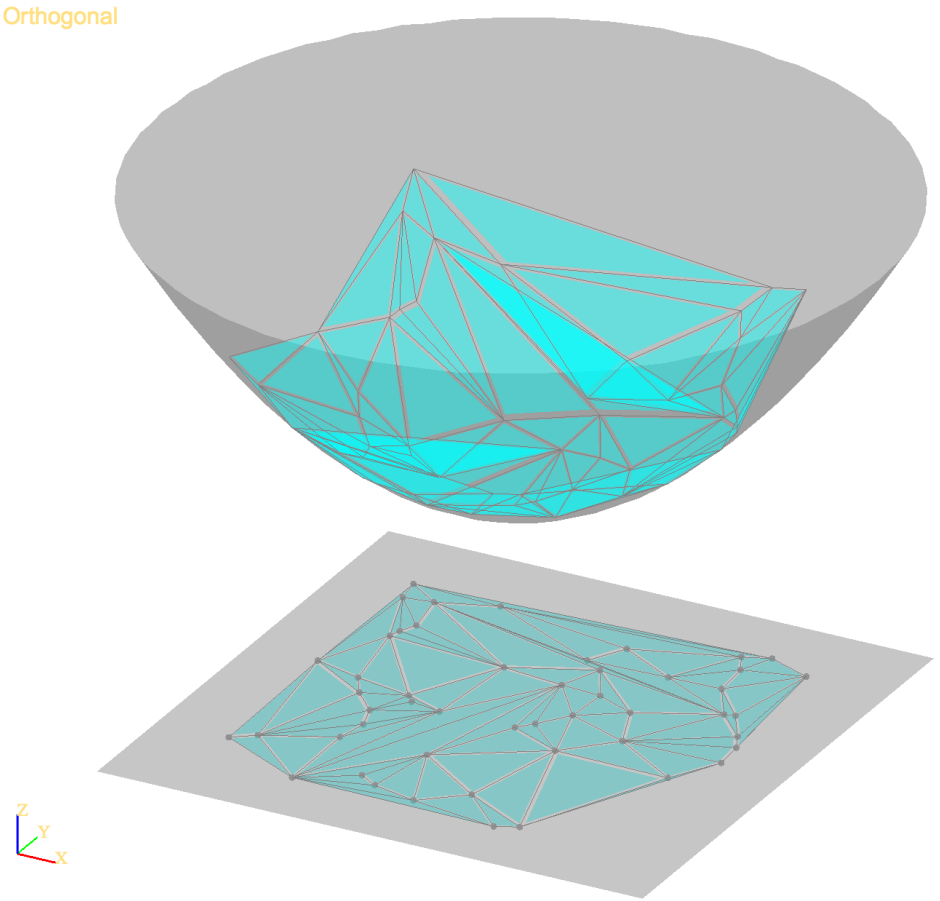


a constrained Delaunay triangulation

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

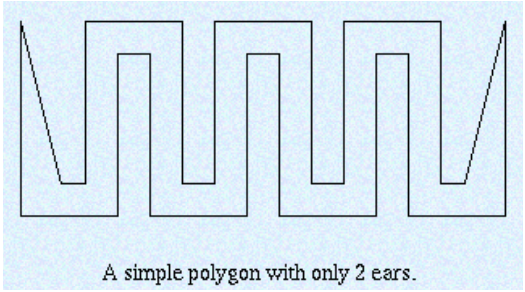
The lifting transformation of CDTs





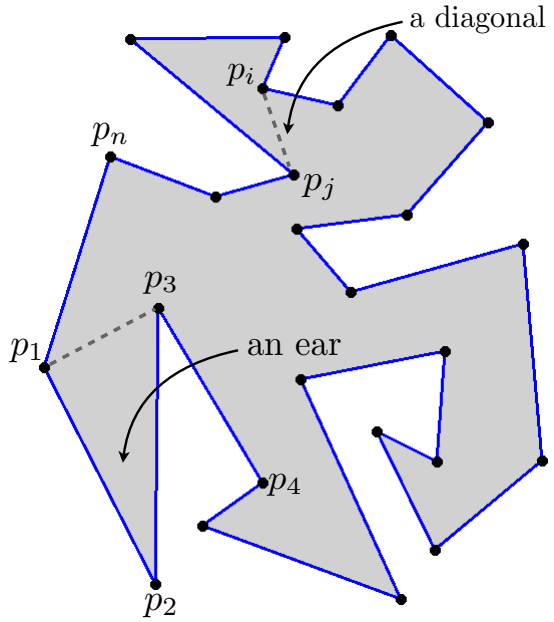
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

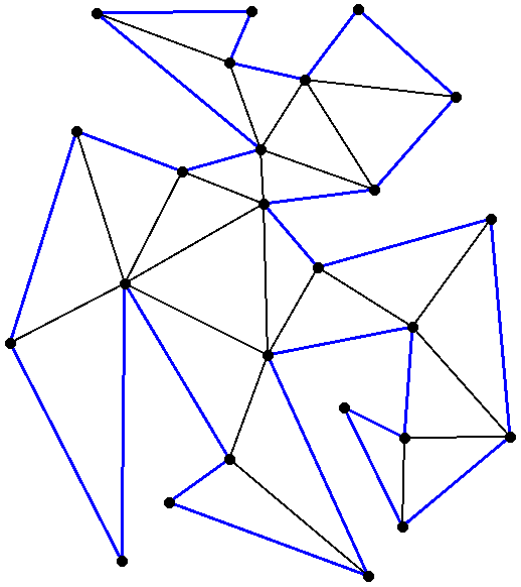
Theorem 1.1 (Two-Ear theorem [8]). *Every simple polygon with more than* 3 *vertices has at least two ears.*



The Ear-Clipping algorithm to triangulate a simple polygon.

1 **while** P contains more than 3 vertices **do**
2 Find an ear $\{a_{i-1}, a_i, a_{i+1}\}$ of P;
3 Output a triangle $\{a_{i-1}, a_i, a_{i+1}\}$;
4 Update $P := P \setminus \text{conv}\{a_{i-1}, a_i, a_{i+1}\}\}$;
5 **endwhile**





From this algorithm, it is easy to see that every triangulation of a simple polygon with n vertices has exactly $n-2$ triangles and $n-3$ diagonals. These counts can also be derived from the Euler's formula (we leave it as an exercise).

the worst case of finding one ear takes $O(n^2)$ time.

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Algorithm: IncrementalCDT(S, L)
Input: A PSLG (S, L), $k := |L|$;
Output: the CDT $\mathcal{T}$ of (S, L);

```
1  construct an initial CDT T_0 of S;
2  for i = 1 to k do
3      if s_i ∈ L and s_i ∉ T_{i-1} then
4          RecoverEdge(s_i, T_{i-1});
5          Let E be the set of new edges in T_{i-1};
6          ConstrainedLawsonFlip(E);
7      endif
8  endfor
```

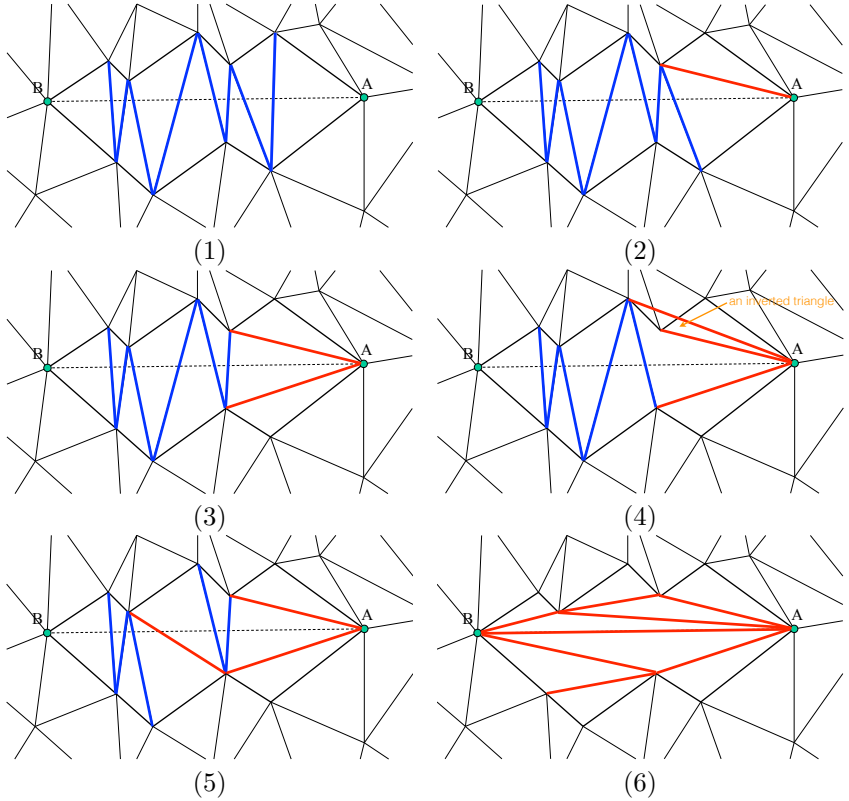
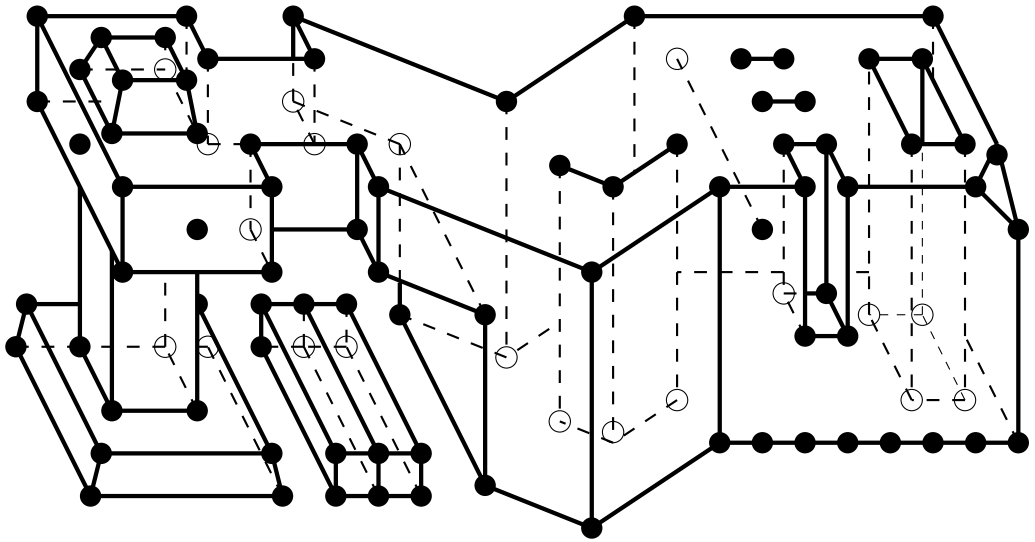


FIGURE 15. Recover an edge by flips. (Figures from S. Owen).

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Piecewise linear complexes

- A 3D *piecewise linear complex* (PLC) [**Miller et al'96**] is a collection $\mathscr{X}$ of vertices, edges, polygons, and polyhedra, collectively called *cells*, such that
  - **(1)** the boundary of each cell in $\mathscr{X}$ is also cells in $\mathscr{X}$; and
  - **(2)** if $f, g \in \mathscr{X}$ and $f \cap g \neq \emptyset$, then $f \cap g$ is a union of cells in $\mathscr{X}$.

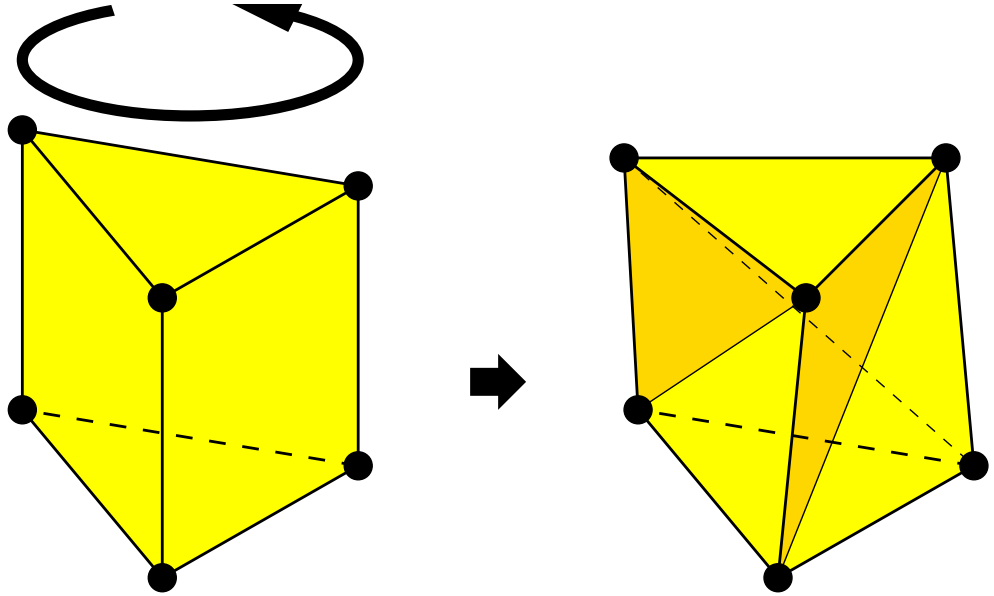


— figure by J. Shewchuk

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Some polyhedra need Steiner points (additional points)

- 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**].

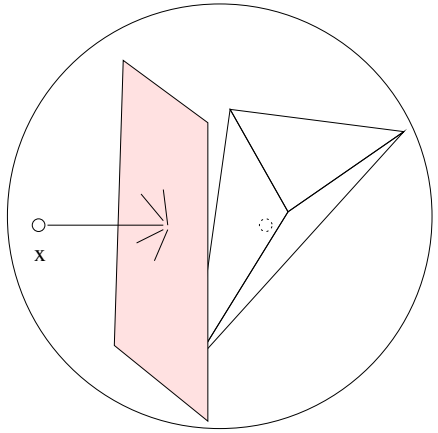


The Schönhardt Polyhedron [**1928**]

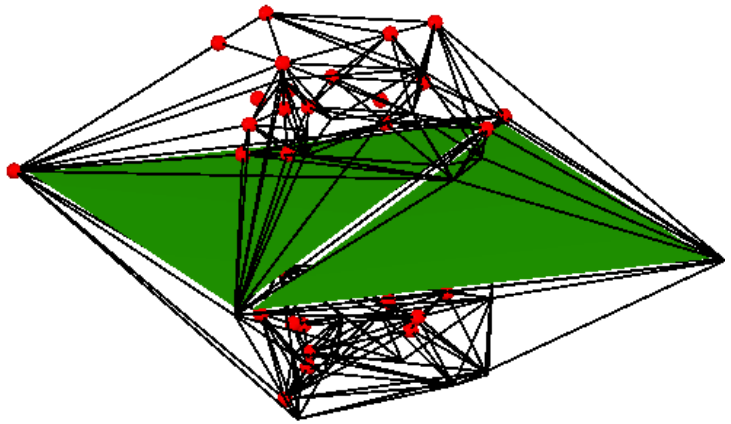
(1) Jakob Steiner (1796 — 1863), a Switzerland native and a geometer from Berlin.

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

- We generalize the definition of 2d CDT of [**Lee & Lin 1986**] into 3D.
- A tetrahedron is constrained Delaunay if: (i) it does not intersect any constraint in its interior; and (ii) its circumsphere contains no vertex that is visible from its interior.
- A tetrahedralization $\mathcal{T}$ of a PLC $\mathcal{X}$ is a constrained Delaunay tetrahedralization (CDT) if every tetrajedron in $\mathcal{T}$ is constrained Delaunay [**Shewchuk 1998**].



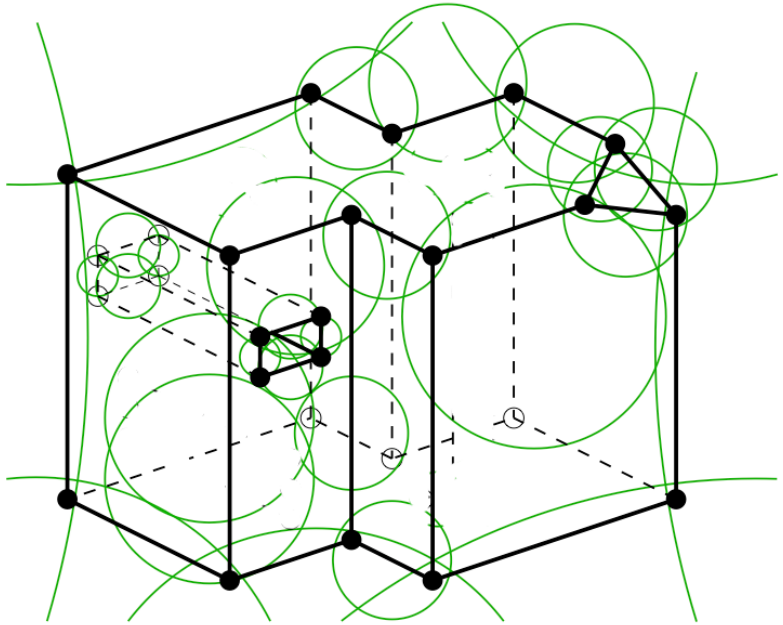
a constrained Delaunay tetrahedron



a constrained Delaunay tetrahedralization (CDT)

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

- 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

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

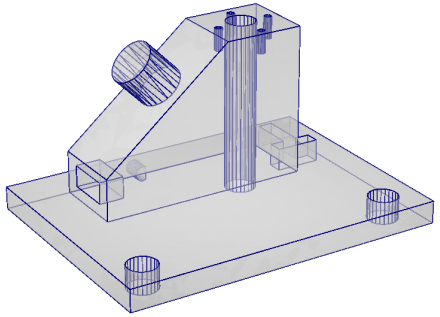
A CDT algorithm

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

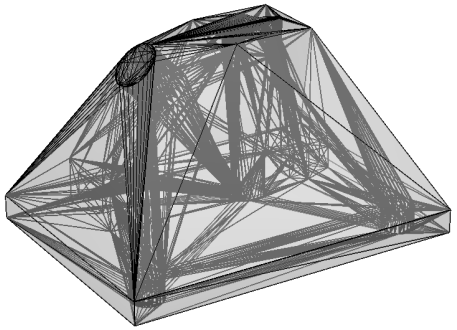
(1) **Initialization**: Creating a Delaunay tetrahedralization of the vertices of $\mathcal{X}$;

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

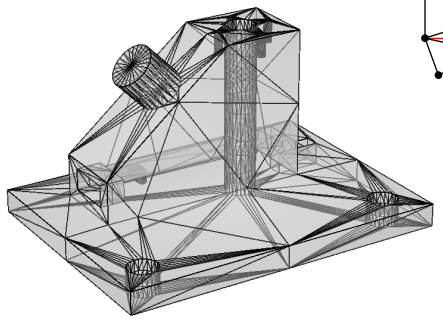
(3) **Polygon insertion**: Generating the Steiner CDT of $\mathcal{X}$.

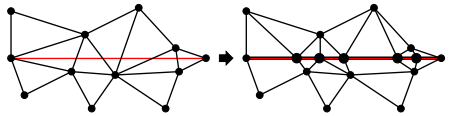


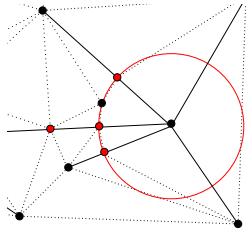
An input PLC $\mathcal{X}$



(1) Initialization

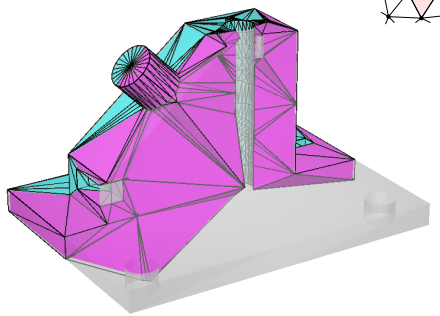


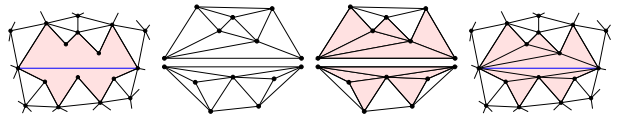


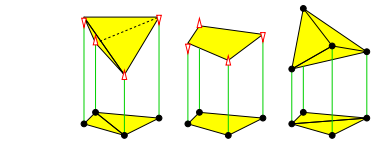


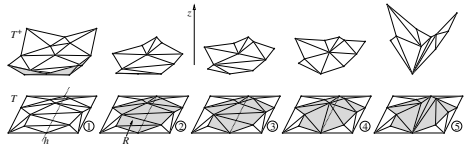
(2) Segment insertion

[Si & Gärtner 2005]







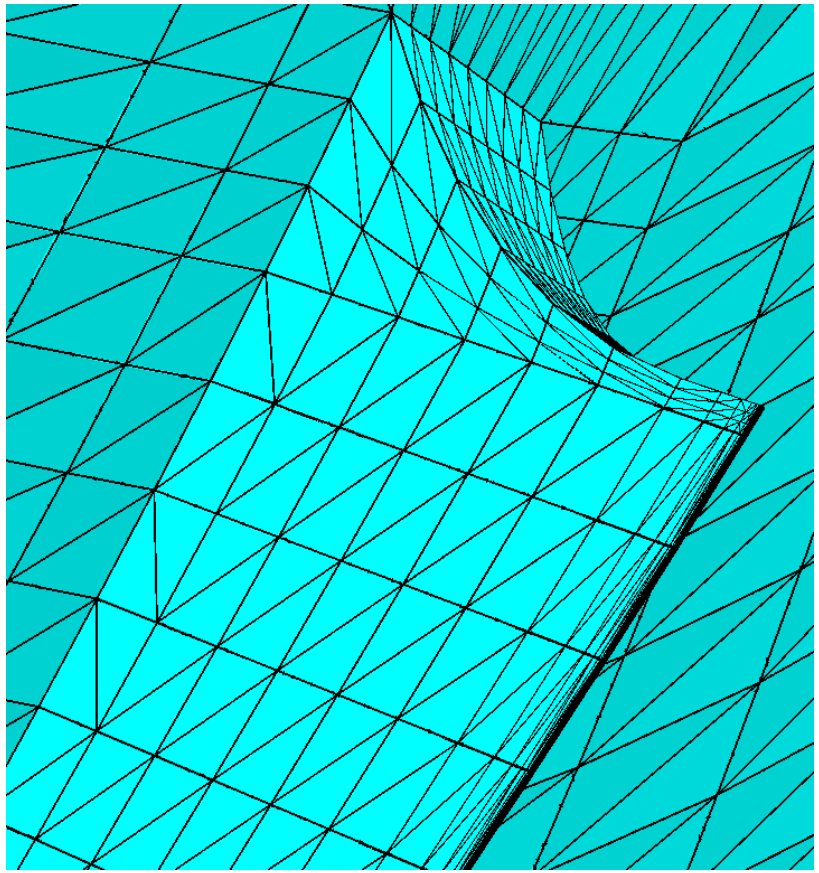


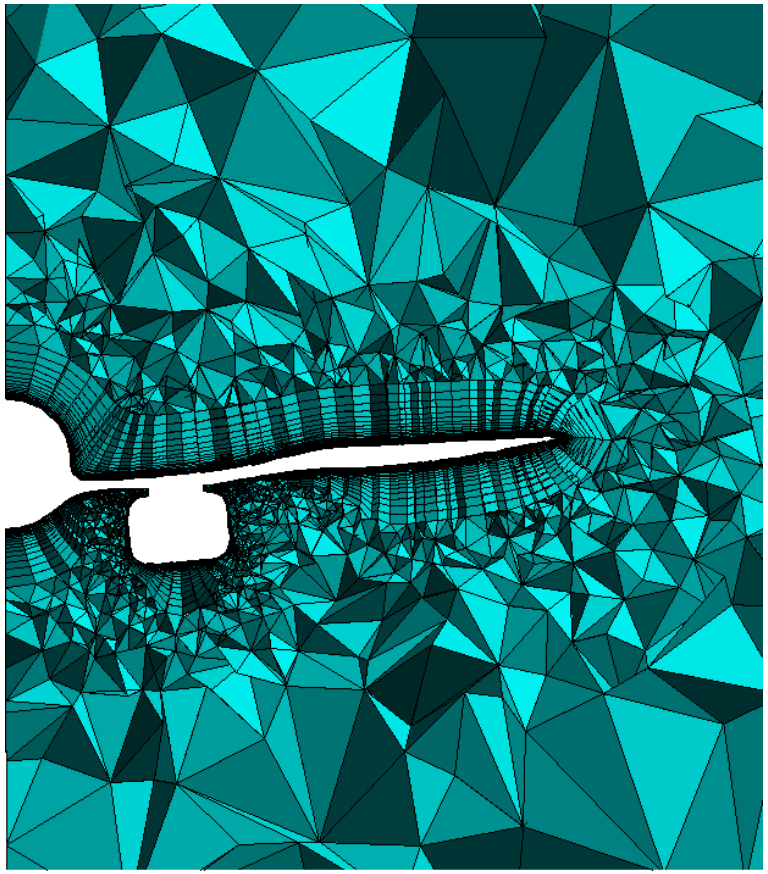
(3) Polygon insertion

[Shewchuk 2003] [Si & Shewchuk 2014]

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

- 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

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Three classical 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`).

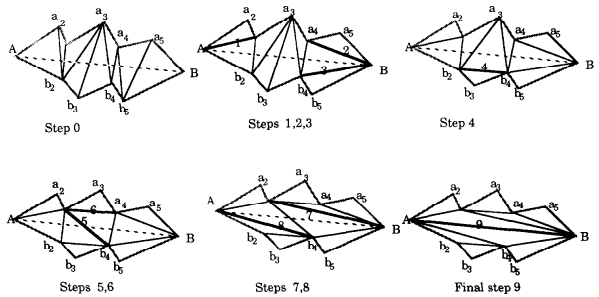


Fig. 11. Step 0, steps 1, 2, 3, step 4, steps 5, 6, steps 7, 8, and step 9.

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

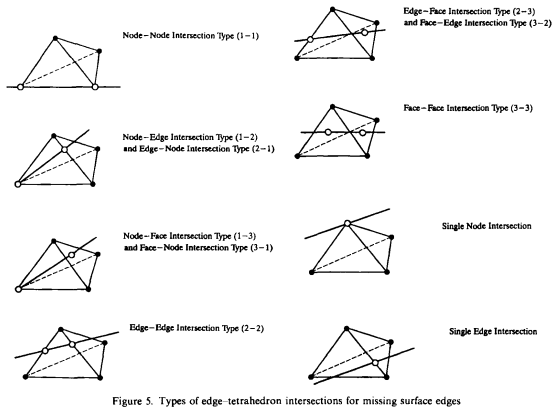


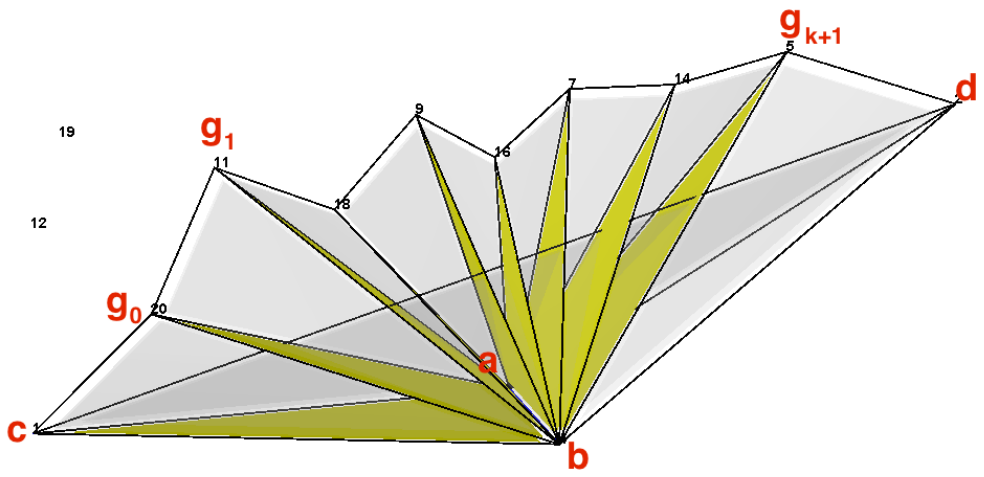
Figure 5. Types of edge–tetrahedron intersections for missing surface edges

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

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

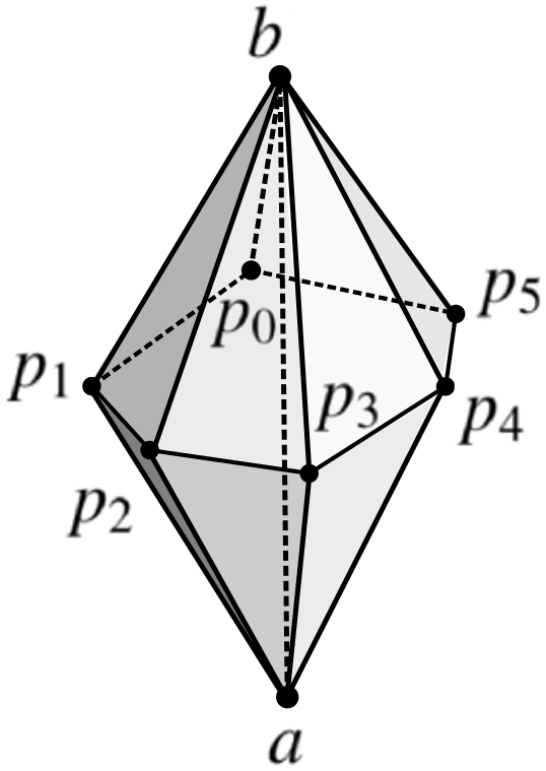
Recover edge (face) by flips

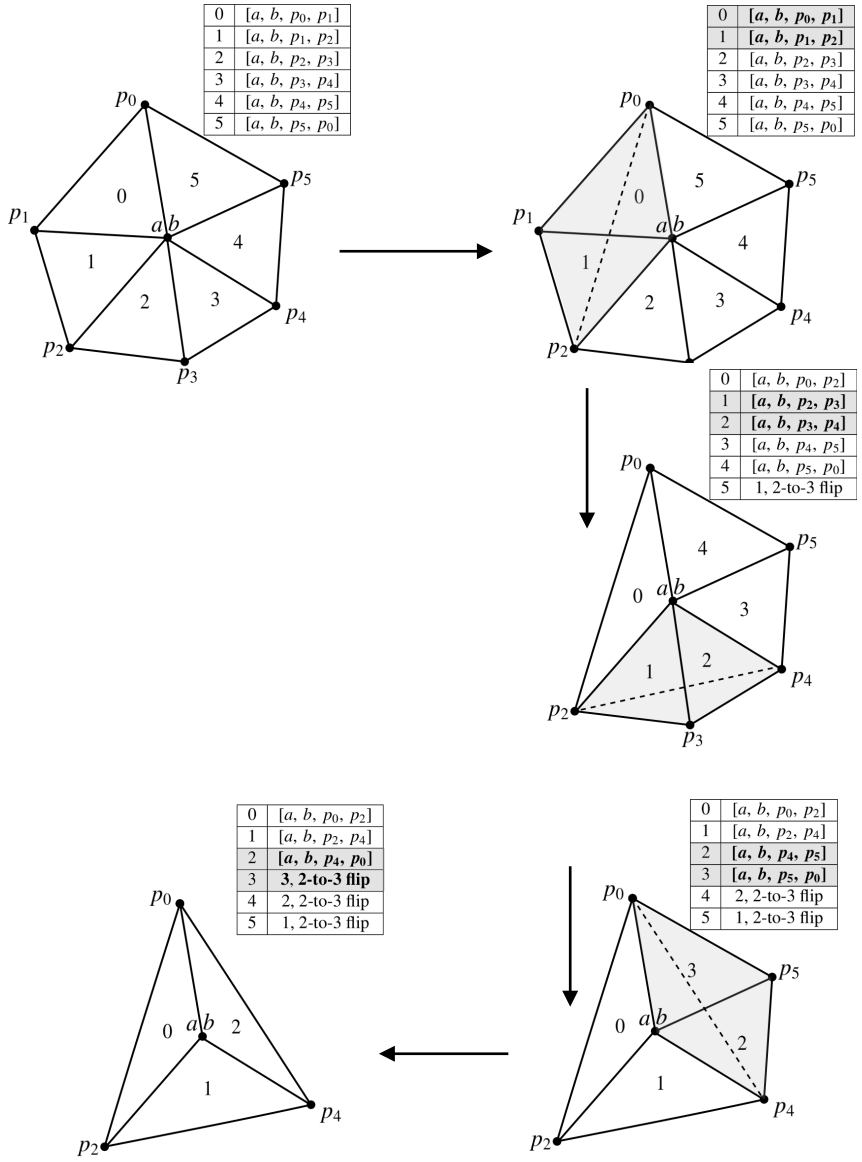
- Edge (and face) recovery by flips:
  - Maintain a list L of all faces that are intersecting an edge (or a face);
  - Remove each face in L by either a 2-to-3 flip or by the edge removal algorithm;
  - Stop either (i) L is empty, or (ii) no face in L can be removed;



© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

The general n-m-flip





© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Where to insert interior Steiner points?

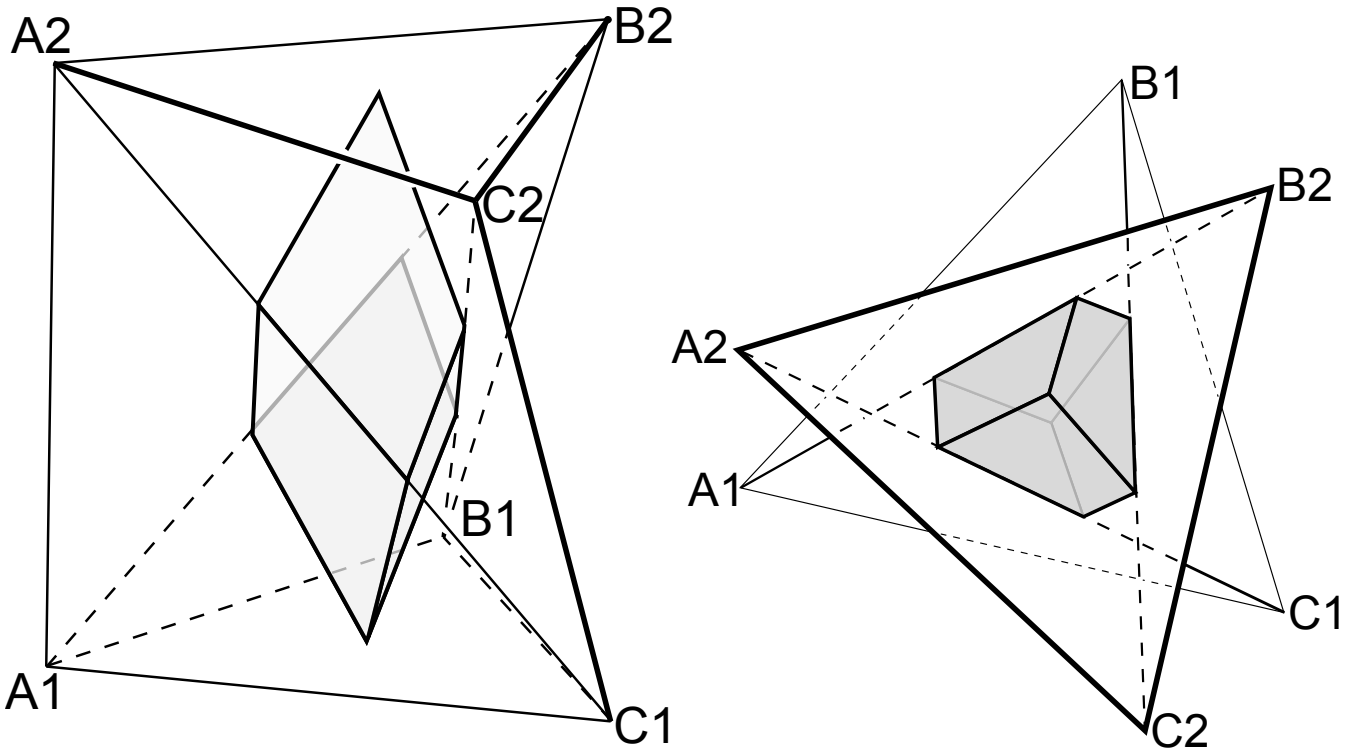


Figure: The (open) valid domain for placing Steiner points inside the Schönhardt polyhedron. A side view (left) and a top view (right) are shown.

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Where to insert interior Steiner points (cont'd)?

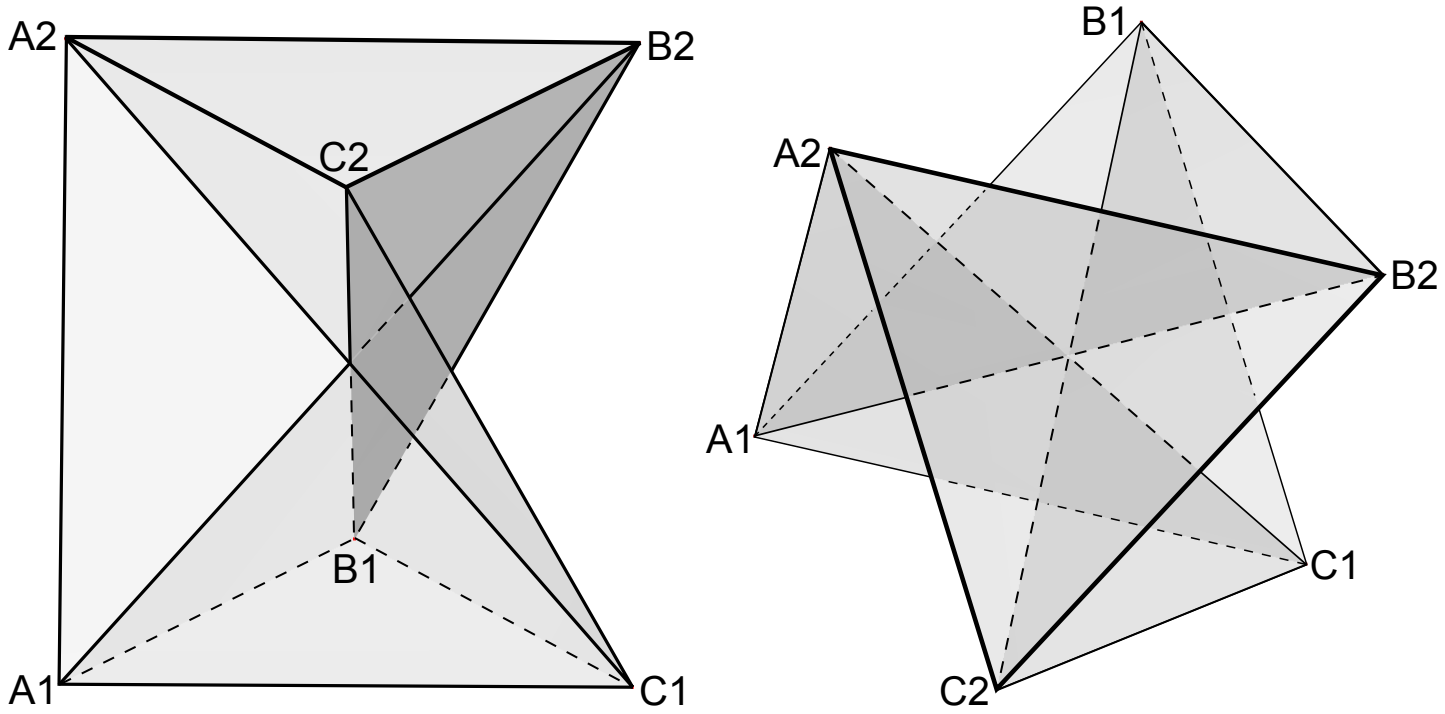


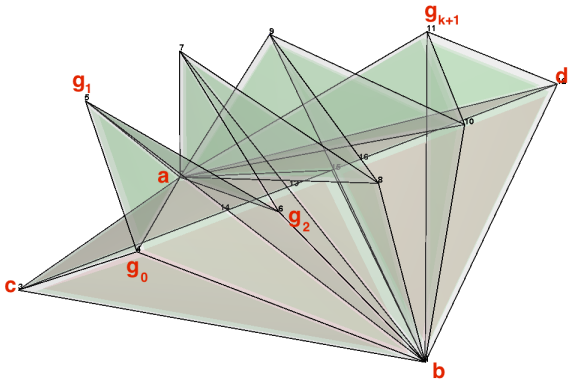
Fig. 9 The Schönhardt polyhedron with a rotation angle of $\vartheta = 60°$. The edges A_1B_2 and A_2C_1, A_1B_2 and B_1C_2, and A_2C_1 and B_1C_2 are coplanar since the dihedral angles between some faces is $0°$. A side view (left) and a top view (right) are shown.

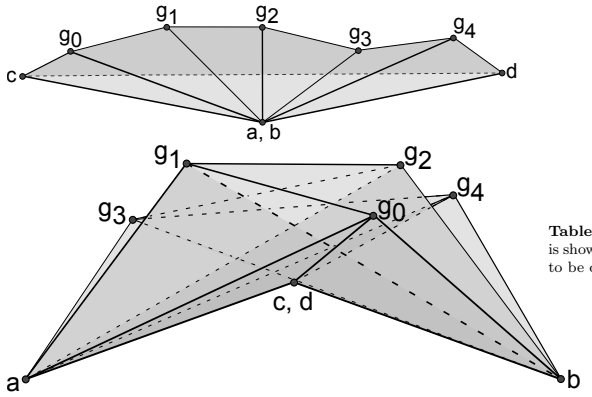
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Add interior Steiner points

Now the polyhedron σ_n, $n = 6 + k$, where $k \geq 0$, is constructed by choosing the boundary faces listed in Table.

(1)	$(\mathbf{a}, \mathbf{c}, \mathbf{d}), (\mathbf{b}, \mathbf{c}, \mathbf{d})$
(2)	$(\mathbf{a}, \mathbf{c}, \mathbf{g}_0), (\mathbf{b}, \mathbf{c}, \mathbf{g_0}), (\mathbf{a}, \mathbf{d}, \mathbf{g}_{k+1}), (\mathbf{b}, \mathbf{d}, \mathbf{g}_{k+1})$
(3)	$(\mathbf{a}, \mathbf{g}_i, \mathbf{g}_{i+1}), (\mathbf{b}, \mathbf{g}_i, \mathbf{g}_{i+1})$, where $i = 0, \ldots, k$





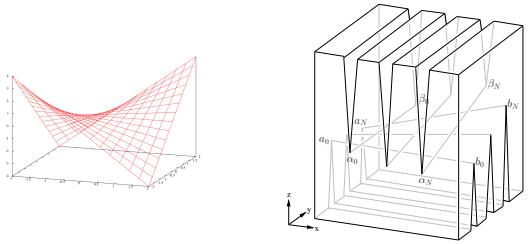
	a	b	c	d	$\mathbf{g}_0$	$\mathbf{g}_1$	$\mathbf{g}_2$	$\mathbf{g}_3$
x	-1.294	4.830	4.830	-3.536	4.253	-0.301	3.117	-2.183
y	10	0	10	0	6.532	9.760	2.999	8.657
z	4.830	1.294	-1.294	3.536	-2.426	0	-2.571	0.646
	$\mathbf{g}_4$	$\mathbf{g}_5$	$\mathbf{g}_6$	$\mathbf{g}_7$				
x	1.874	-3.330	0.163	-4.051				
y	1.002	6.864	-0.105	3.184				
z	-1.808	1.350	-0.366	2.242				

Table 4 A choice of the coordinates of the vertices of a σ_{12}. The geometry of this polyhedron is shown in Figure 14. With these coordinates, this polyhedron needs at least 4 Steiner points to be decomposed.

Theorem 4 *Given $n \in \mathbb{N}_{\geq 6}$, one can construct an a 3d polyhedron σ_n with n vertices which has the property that one needs exactly $\left\lceil \frac{n-5}{2} \right\rceil$ interior Steiner points to decompose it.*

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Add interior Steiner points



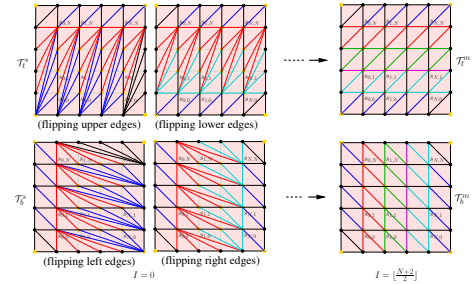
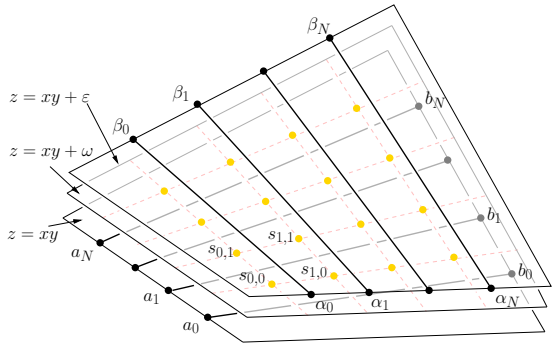


Figure: Left: A saddle surface (a hyperbolic paraboloid). Right: The Chazelle polyhedron with three notches, i.e., $N = 2$, on the top and the bottom faces, respectively.



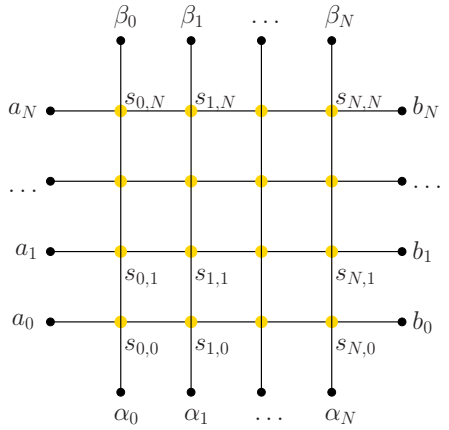
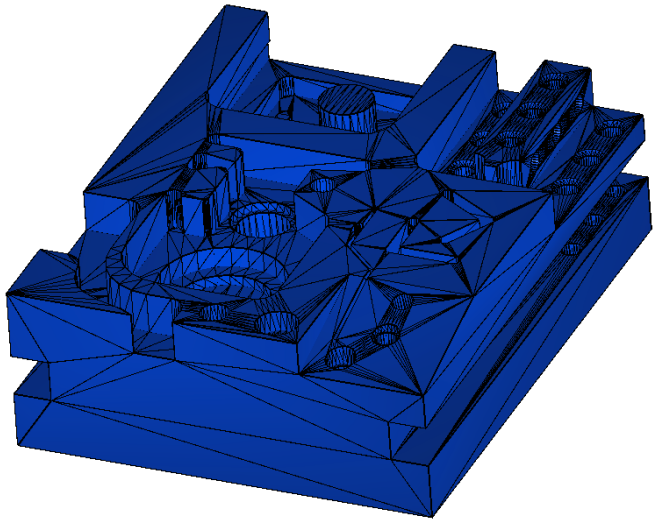
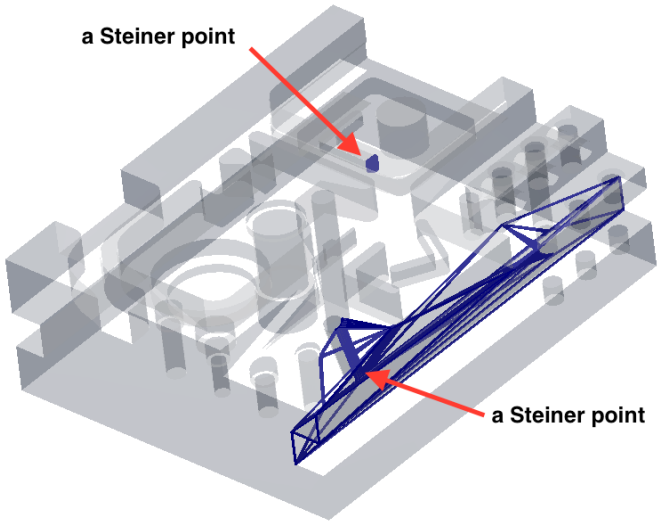


Figure: The interior Steiner points, $\{s_{i,j} \mid i, j = 0, \ldots, N\}$, are placed directly at the intersections of the two set of lines in the xy-plane and all lie on the saddle surface $z = xy + \omega$, where $0 < \omega < \varepsilon$.

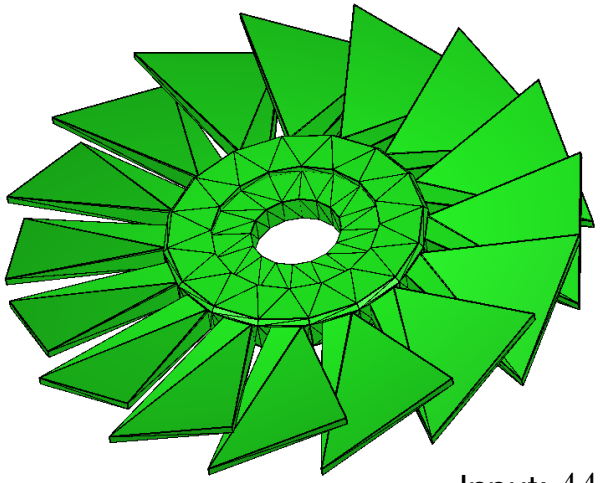
Theorem The reduced Chazelle polyhedron $\Phi^s_{N,\varepsilon}$ needs $(N+1)^2$ interior Steiner points as $\varepsilon \to 0$.

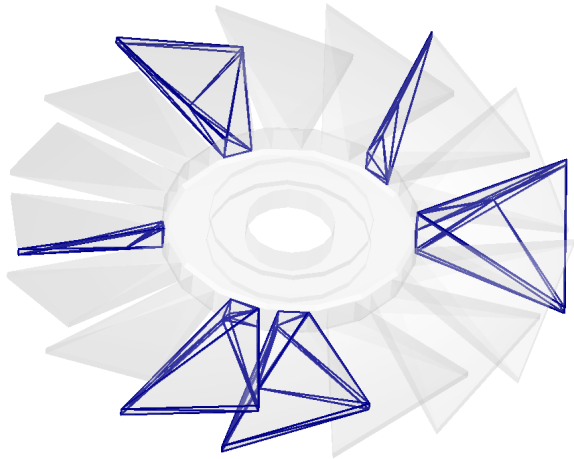
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.





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





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

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

How Meshes Affect Solution

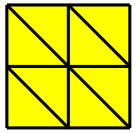
Skinny elements cause problems.

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.

J. Shewchuk, IMR, 2005

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

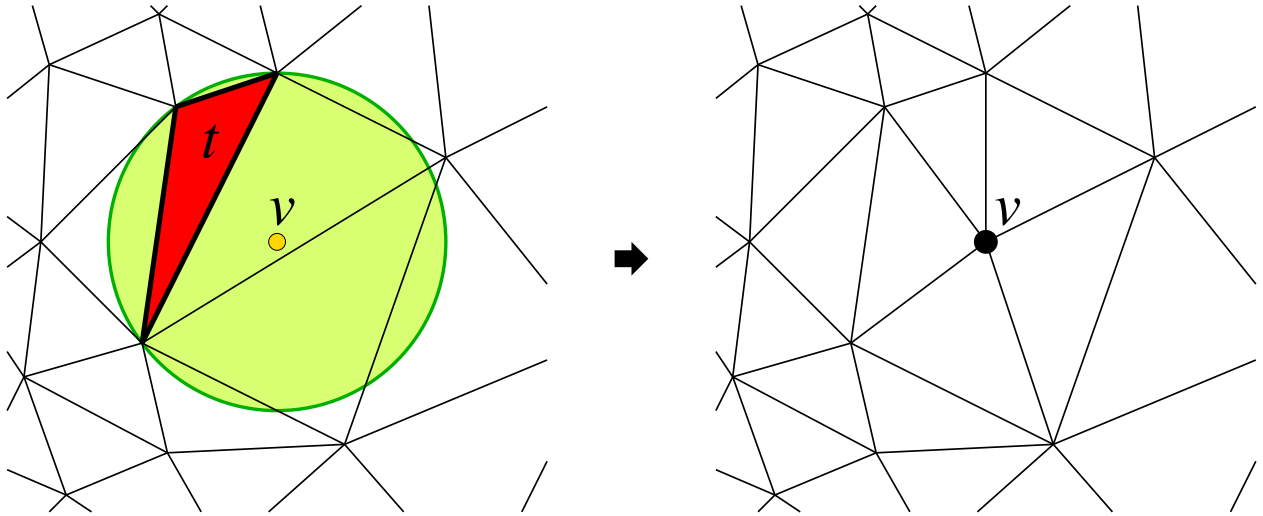
Steiner point generation

- 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**];

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Delaunay Refinement

[Alert: here comes the MAIN IDEA behind all Delaunay refinement algorithms



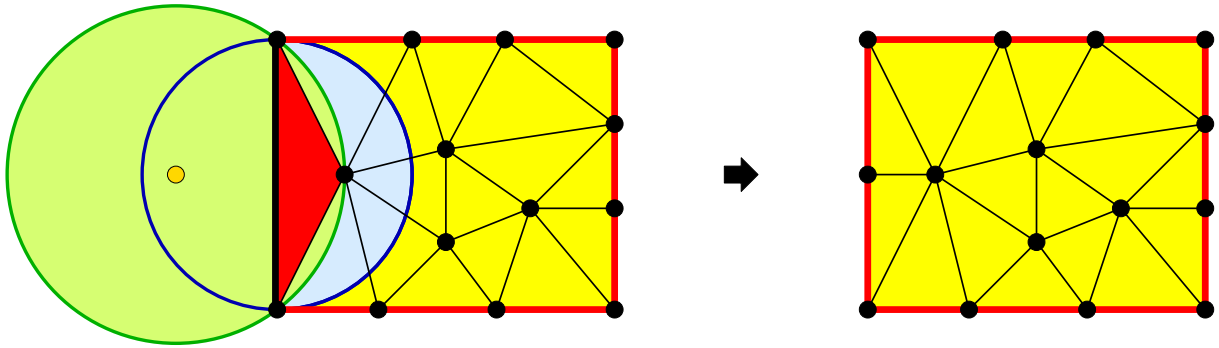
Kill each skinny triangle by inserting vertex at circumcenter. (Bowyer–Watson algorithm.)

All new edges are at least as long as circumradius of t (because v is at center of empty circumcircle).

J. Shewchuk, IMR, 2005

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

What if a circumcenter is outside the domain?

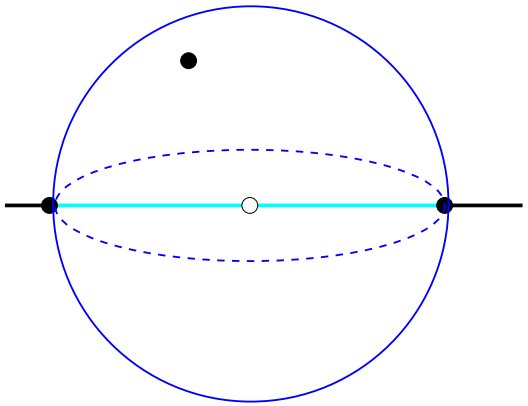


Then a boundary segment is encroached. Split it.

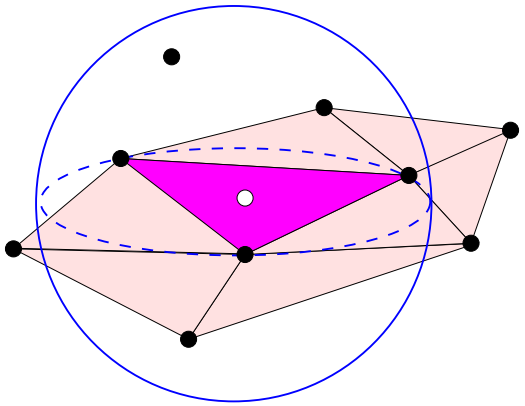
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Steiner points insertion rules

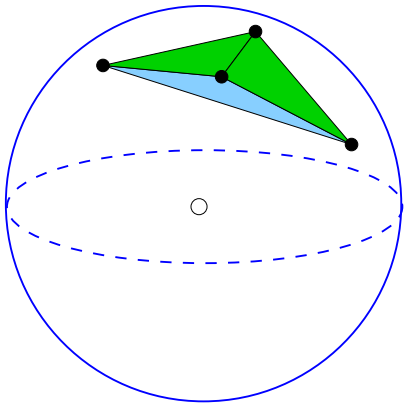
- **Rule 1**: Split a segment if it is encroached.
- **Rule 2**: Split a subface if it is encroached. However, if the new vertex would encroaches upon a segment, reject the vertex. Split the encroached segment(s) instead.
- **Rule 3**: Split a badly-shaped tetrahedron. However, if the new vertex would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



Rule 1



Rule 2

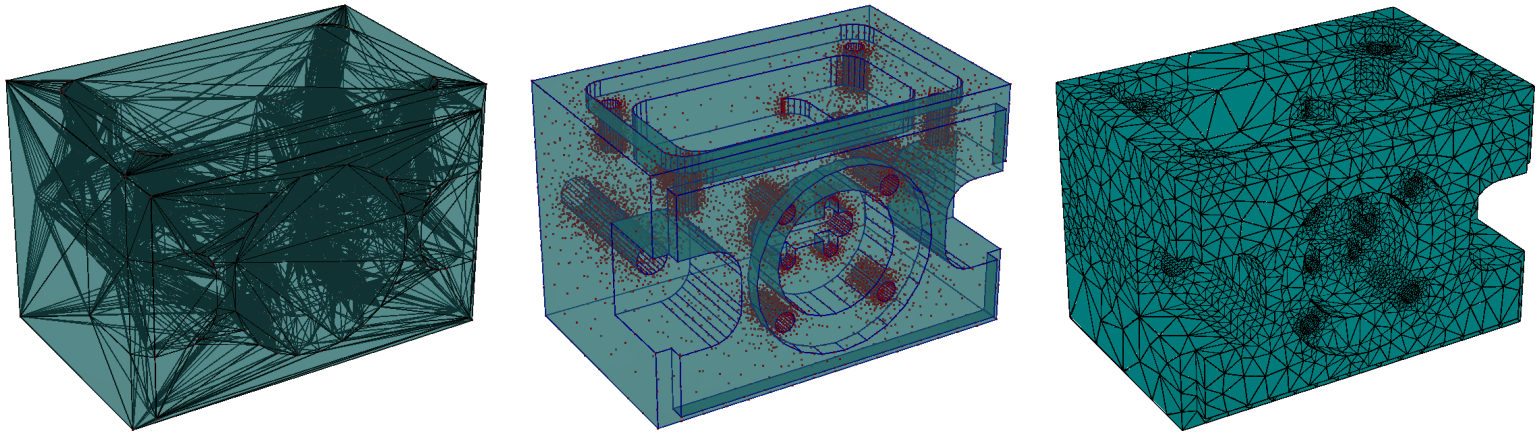


Rule 3

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Delaunay refinement

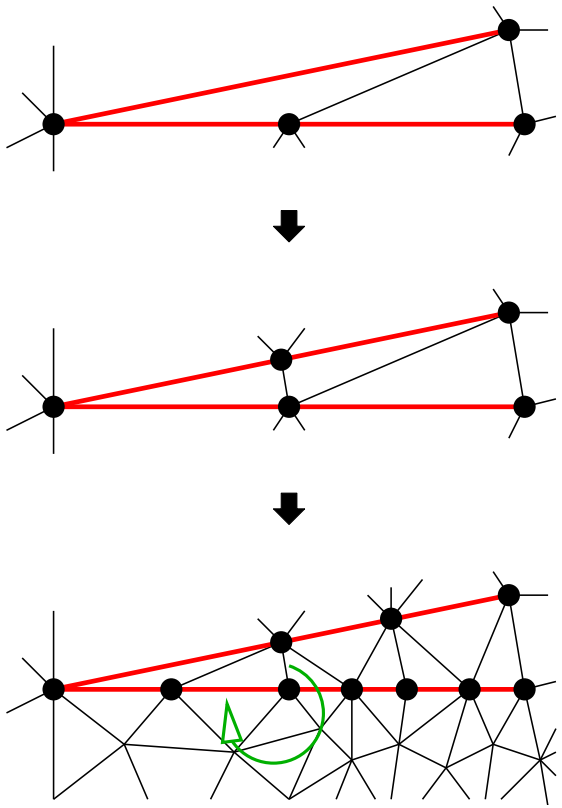
DelaunayRefinement $(\mathcal{X}, \rho_0)$
// $\mathcal{X}$ is a PLC; ρ_0 is a radius-edge ratio bound.
1 Initialize a set V of the vertices of $\mathcal{X}$;
2 Initialize a Delaunay tetrahedralization $\mathcal{D}$ of V;
3 **repeat**:
4 Create a new point by rule i, $i \in \{1, 2, 3\}$;
5 Add **v** to V, update $\mathcal{D}$ of V;
6 **until** {no new point can be generated};
7 **return** $\mathcal{D}$ of V;



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

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

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



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

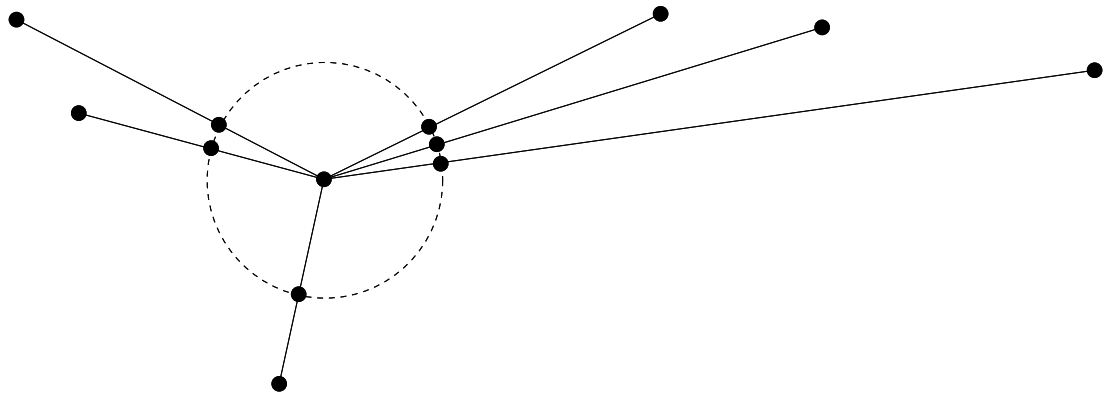
Another vertex is inserted, creating a very short edge.
Oops!

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

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Fixing the Algorithm

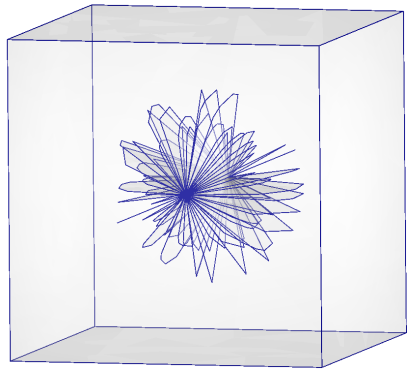
- “Groom” input by splitting segments with augmenting points.
- All segments at same length?



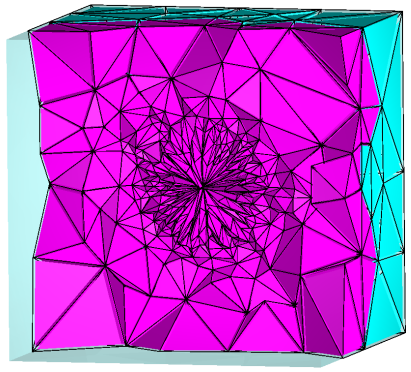
Sangria seminar, 03.10.03 – p.9/18

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

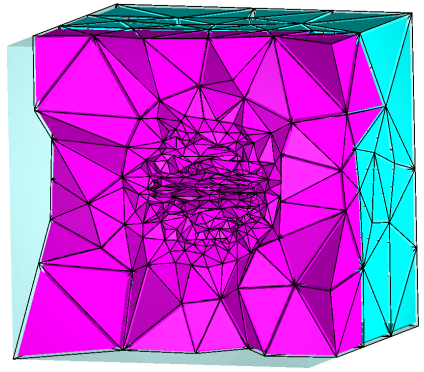
Constrained Delaunay refinement



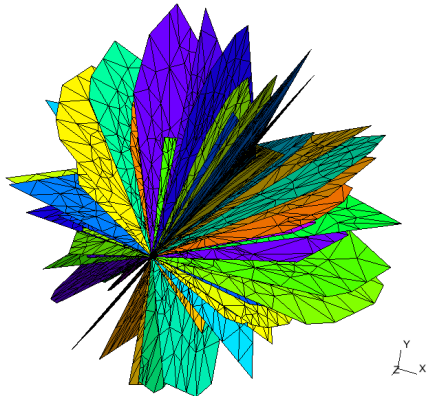
Input `test-64-6`
161 vertices, 70 polygons



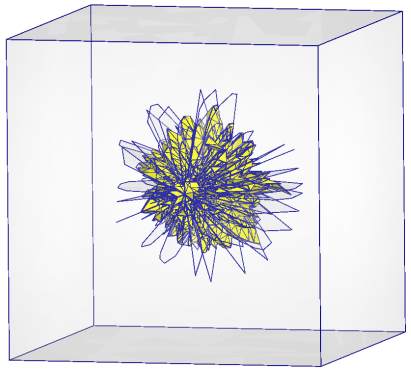
Tet mesh, 3,733 vertices
(cut along the Z-axis)



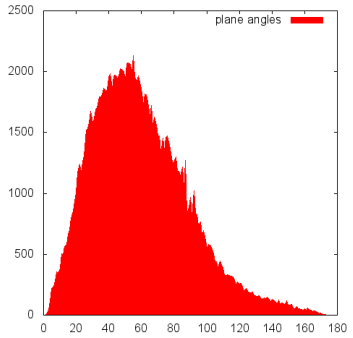
Tet mesh, 23,727 tets
(cut along the Y-axis)



refined "fan blades"



remaining skinny tetrahedra
(radius-edge ratios > 2)



plane angles

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

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Centroid Voronoi Tessellation

$$\inf_{\mathbf{x}_1,\ldots,\mathbf{x}_N, D_1,\ldots,D_N} \sum_{i=1}^{N} \int_{D_i} \|\boldsymbol{x} - \boldsymbol{x_i}\|^2 \rho(\mathbf{x}) d\mathbf{x}.$$

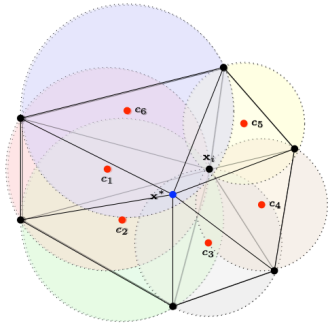
Two necessary conditions for the minimizer of (1)[Lloyd 1982]
1. D_i is the Voronoi region V_i of $\mathbf{x}_i$.
2. $\mathbf{x}_i$ is the centroid of the V_i, namely

$$\mathbf{x}_i = \frac{\int_{V_i} \mathbf{x}\rho(\mathbf{x})d\mathbf{x}}{\int_{V_i} \rho(\mathbf{x})d\mathbf{x}}.$$

Therefore the minizer is called Centroid Voronoi Tessellation(CVT).

CVT and its application: Du, Faber and Gunzburger 1999, Du, Gunzburger and Ju 2003, Du and Wang 2002, 2003, Huang, Qing and Wang 2008.

ODT Smoothing

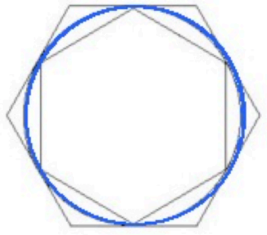


Let $\mathbf{c}_j$ be the center of the circumsphere of τ_j. Then the optimal location $\mathbf{x}^*$ can be written as

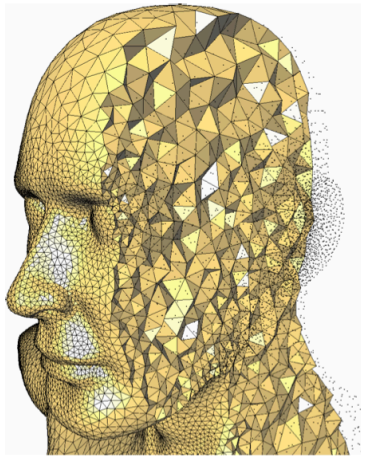
$$\mathbf{x}^* = \frac{\sum_{\tau_j \in \Omega_i} |\tau_j| \mathbf{c}_j}{\Omega_i}.$$

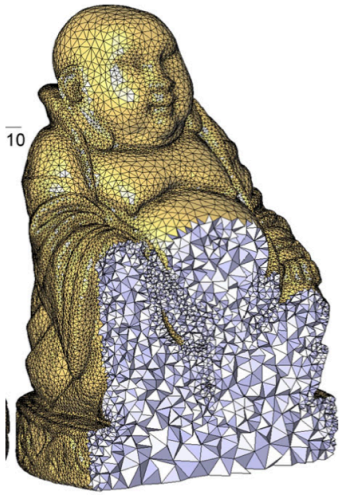
- Chen. *Mesh smoothing schemes based on optimal Delaunay triangulations*. 13th International Meshing Roundtable. 109-120, 2004.
- Alliez, Cohen-Steiner, Yvinec and Desbrun. *Variational tetrahedral meshing*. ACM SIGGRAPH 2005 Courses. 2005.

Duality between ODT and CVT



Inscribe approximation vs. Circumscribe approximation





From Alliez et al ACM Transactions on Graphics. 2005 and Tournois et al. ACM Trans. Graph.2009.

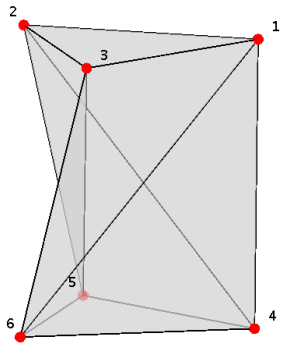
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Slides by Long Chen, UCI

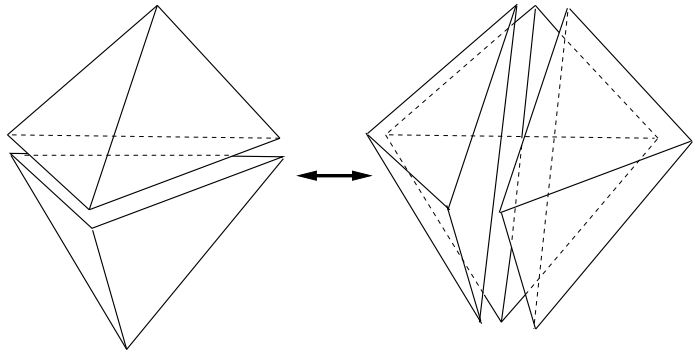
cadence

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

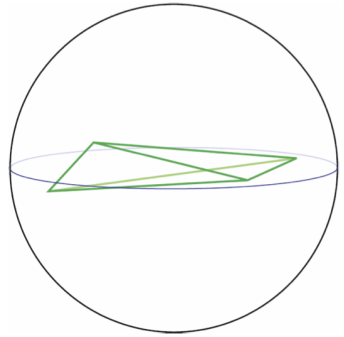
> *In three dimensions the theory is far from developed. The main difficulties are the following:* **(1)** *there exist configurations of boundary points and faces for which no conforming grid of tetrahedra exists unless extra points are inserted,* **(2)** *although 3D analogues of diagonal swapping 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



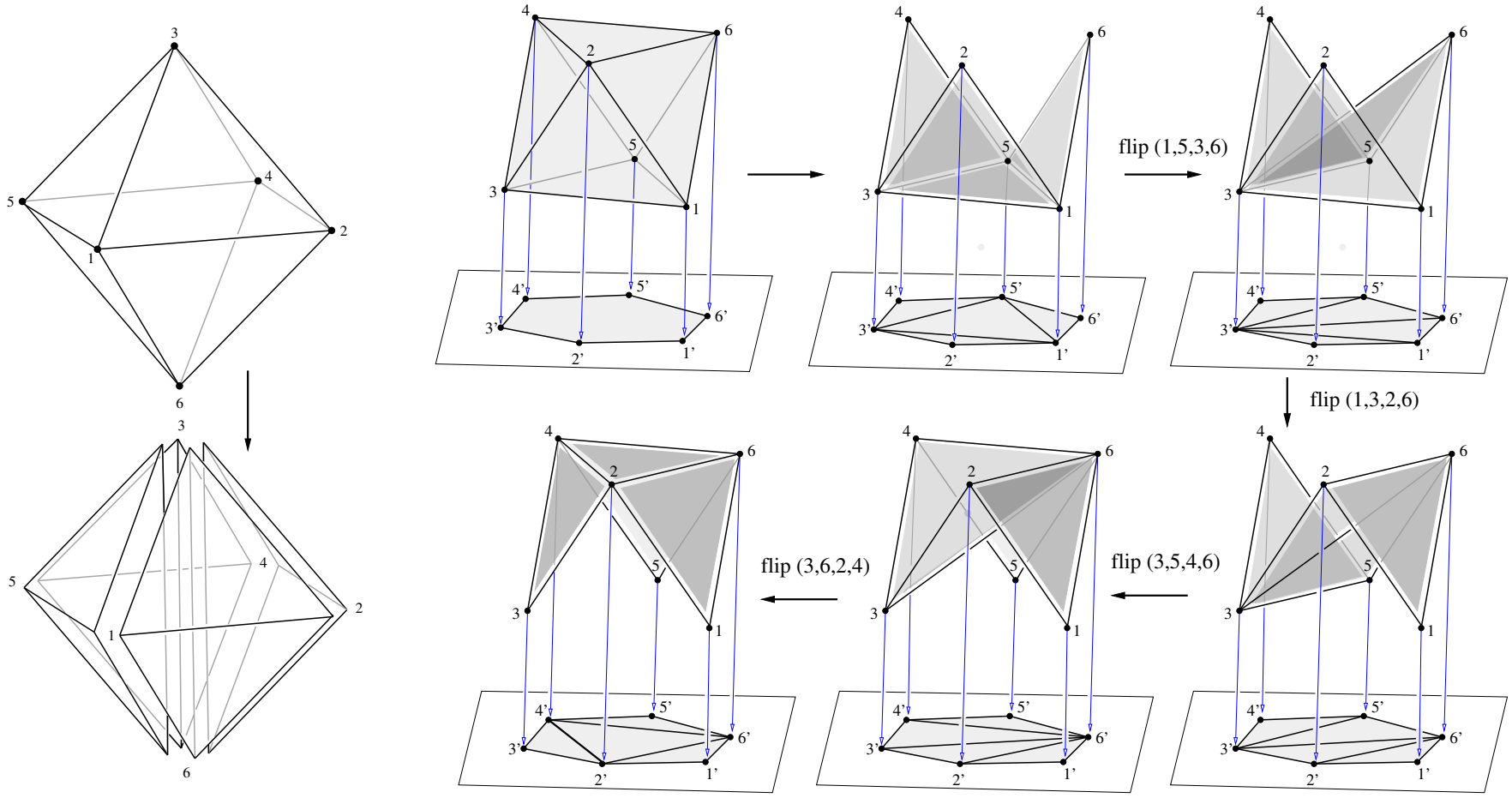
flip 2-to-3 3-to-2



Sliver

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

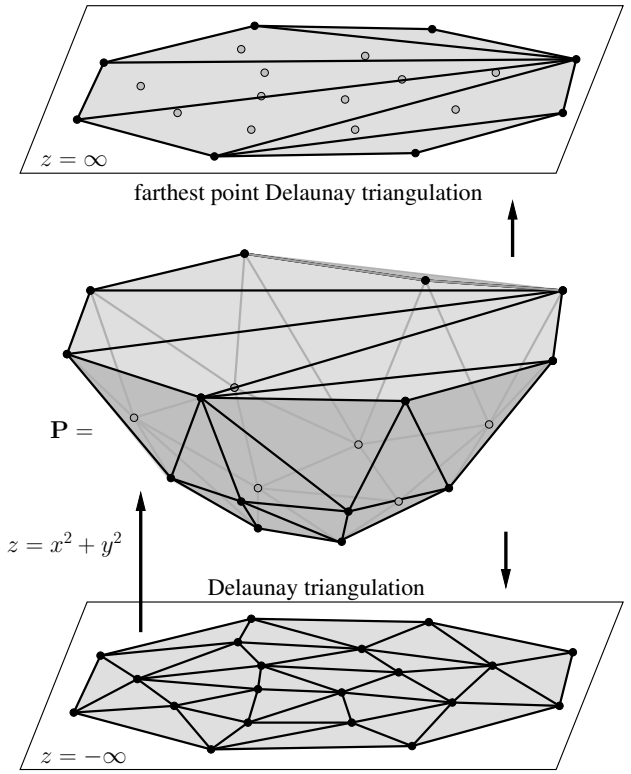
Monotone sequences of flips and triangulations



Sleator, Thurston, and Tarjan, *Rotation distance, triangulations, and hyperbolic geometry*, J. Amer. Math. Soc. 1988

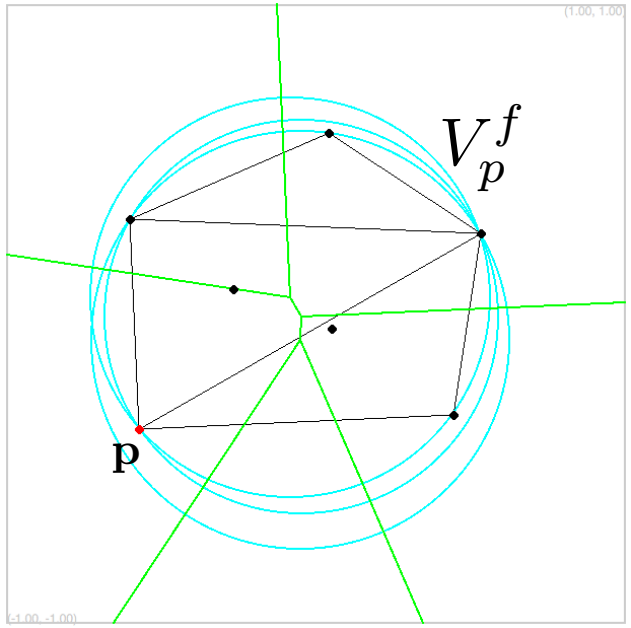
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Transforming triangulations via flips



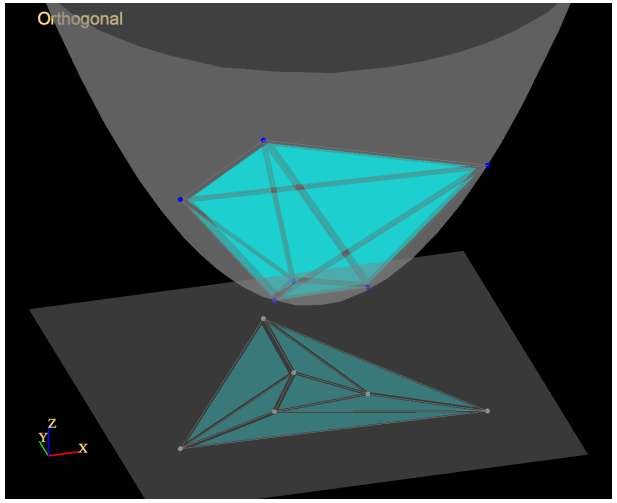
The furtherest point Voronoi Diagram

$$V_p^f = \{\mathbf{x} \in \mathbb{R}^d,\ \|\mathbf{x} - \mathbf{p}\| \geq \|\mathbf{x} - \mathbf{q}\|,\ \forall \mathbf{q} \in S\}$$

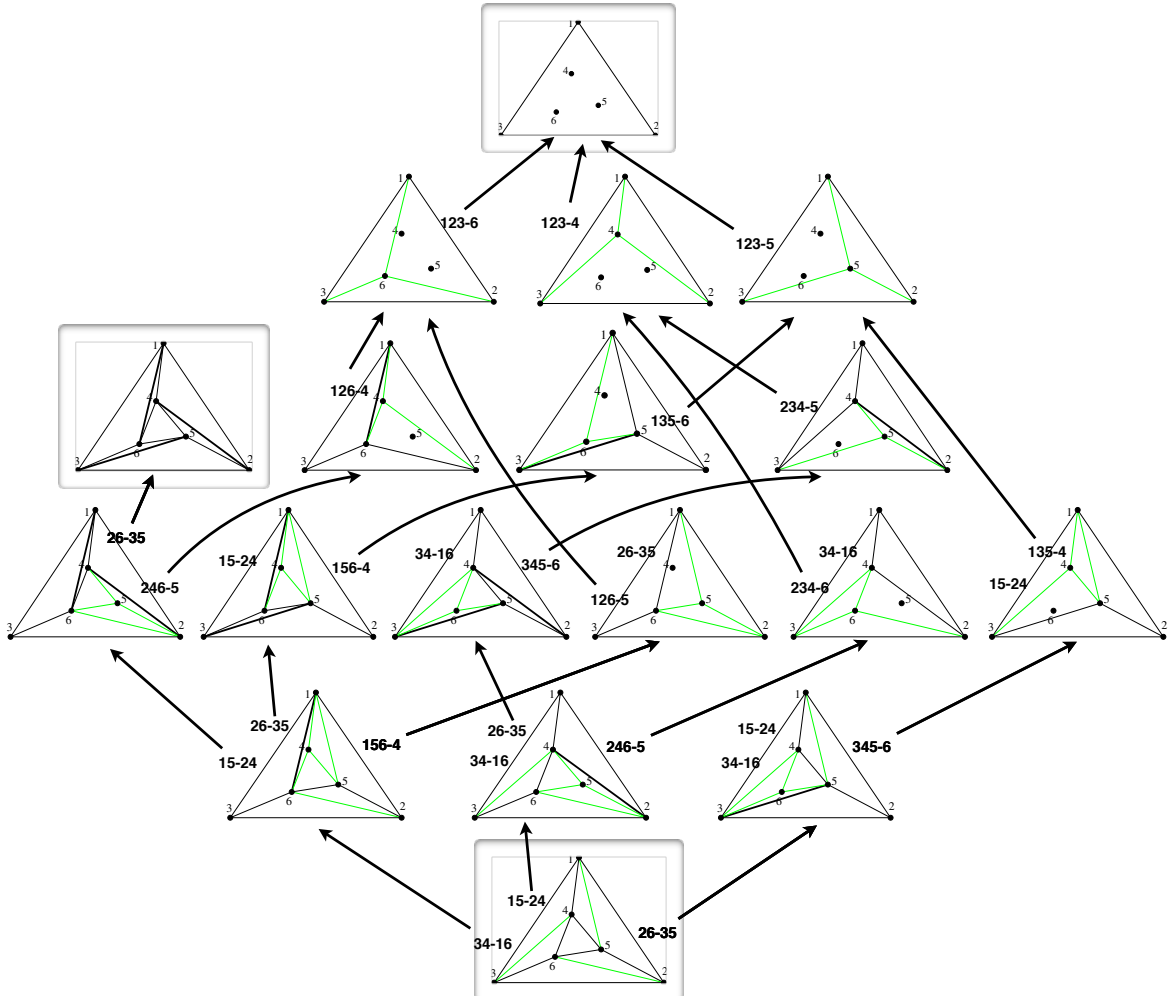


© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

The directed flip graph

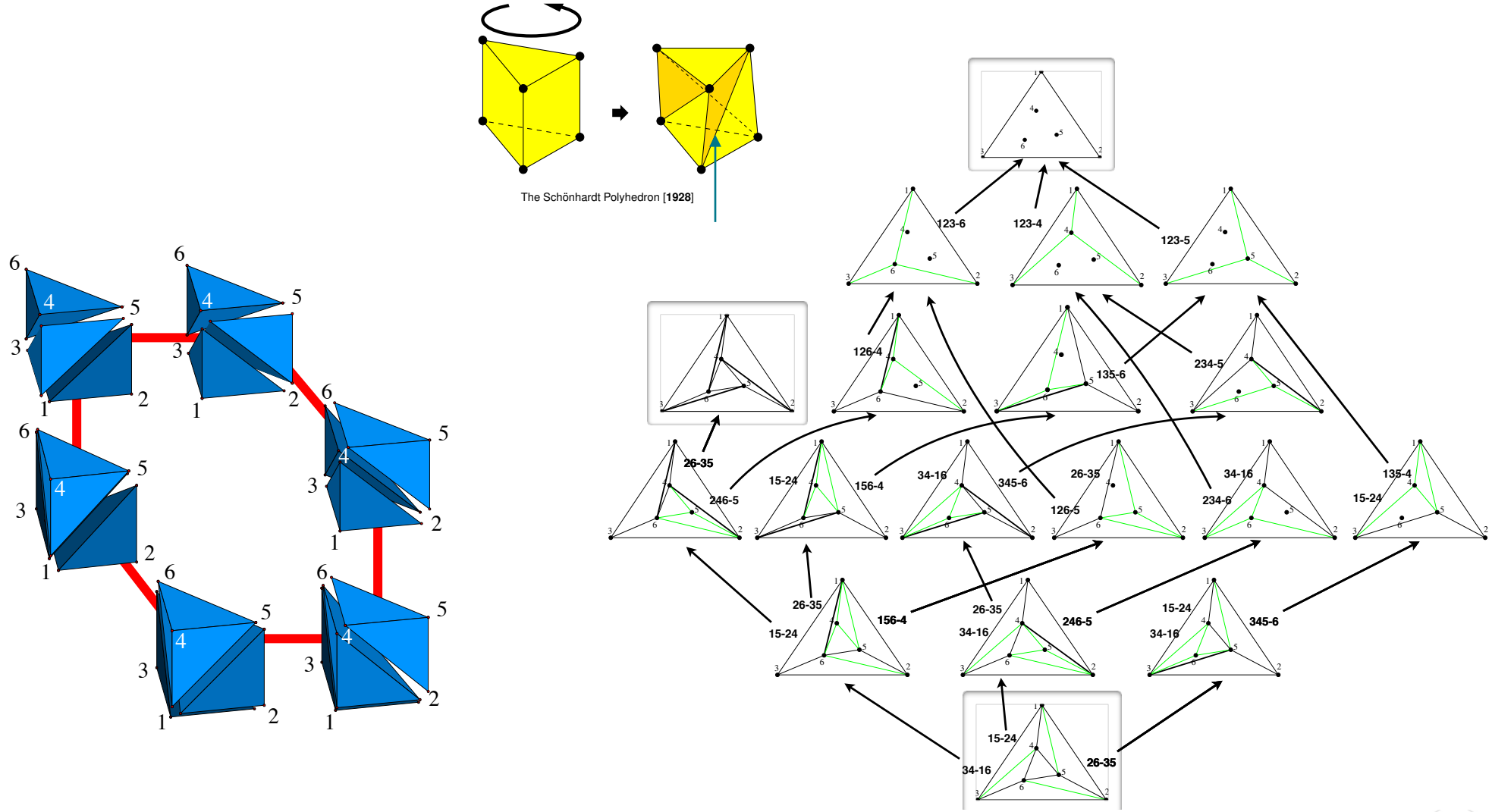


2d_6_points.node



© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

The directed flip graph



© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

The directed flip graph (continued)

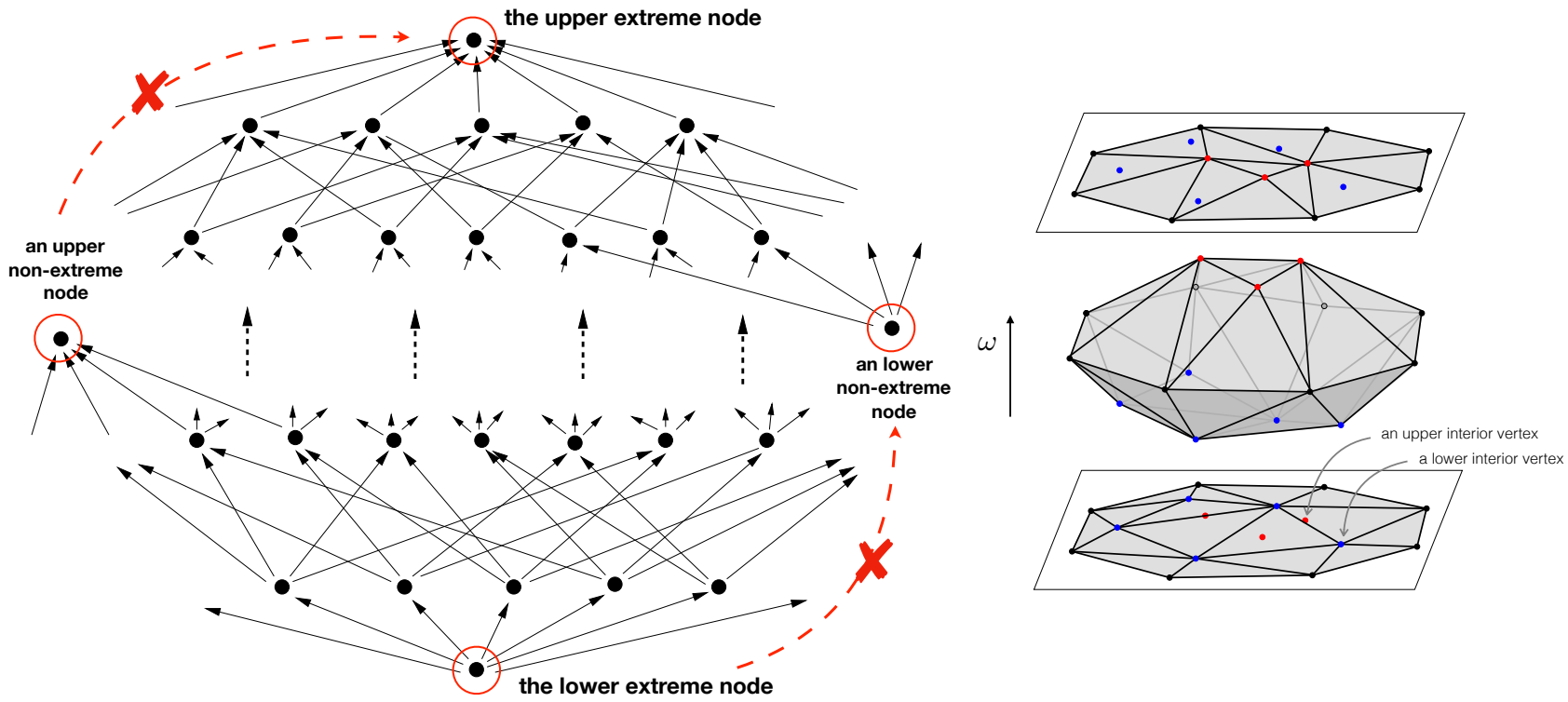


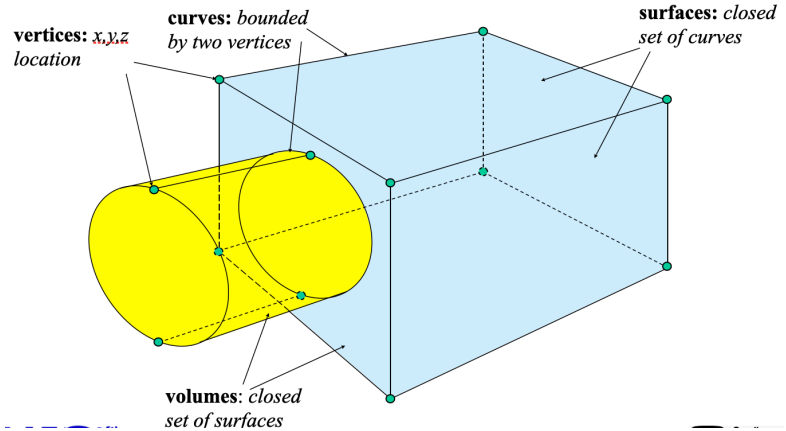
Figure 27: Left: Assume ω is neither convex nor concave. An example of a poset of triangulations of $(\mathbf{A}, \omega)$. Extreme nodes, upper non-extreme nodes, and lower non-extreme nodes are shown. Right: The lower and upper interior vertices of a point set $(\mathbf{A}, \omega)$.

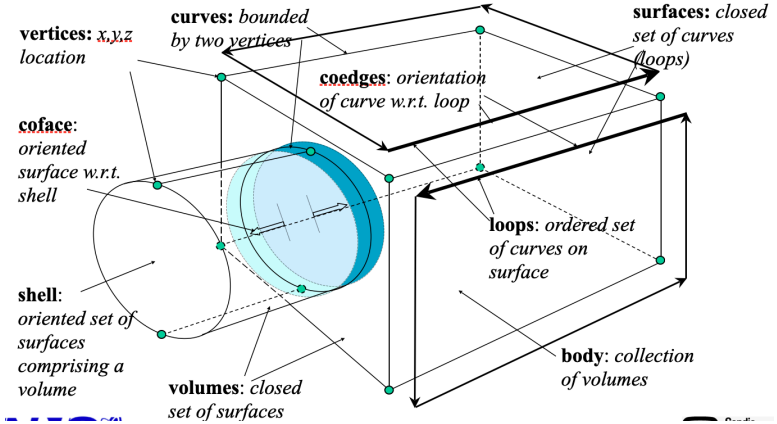
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

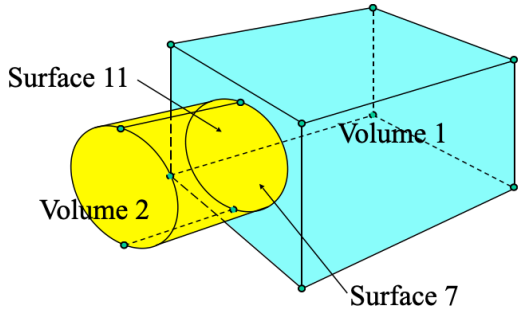
Surface mesh generation

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

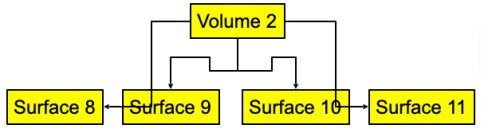
B-reps (Boundary representations)

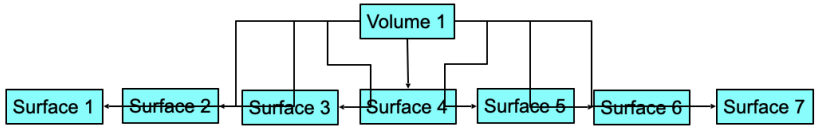


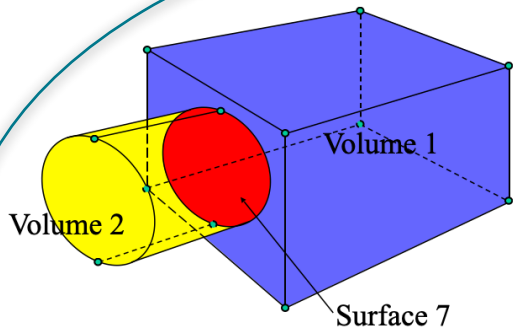




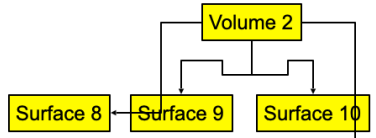
Manifold Geometry: *Each volume maintains its own set of unique surfaces*

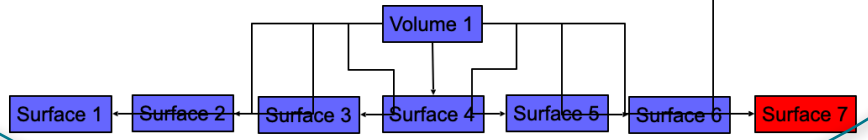






Non-Manifold Geometry: *Volumes share matching surfaces*



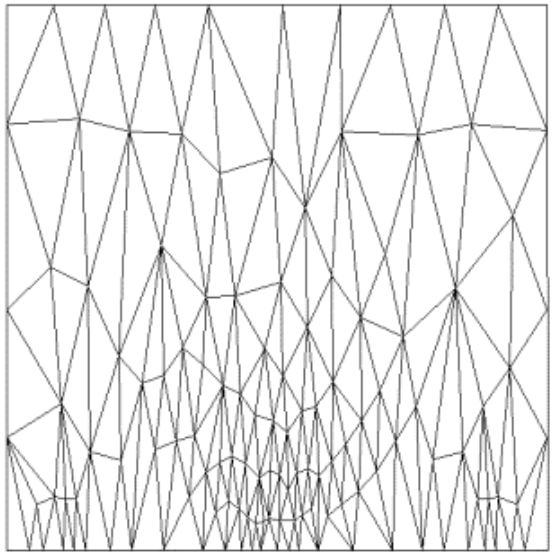


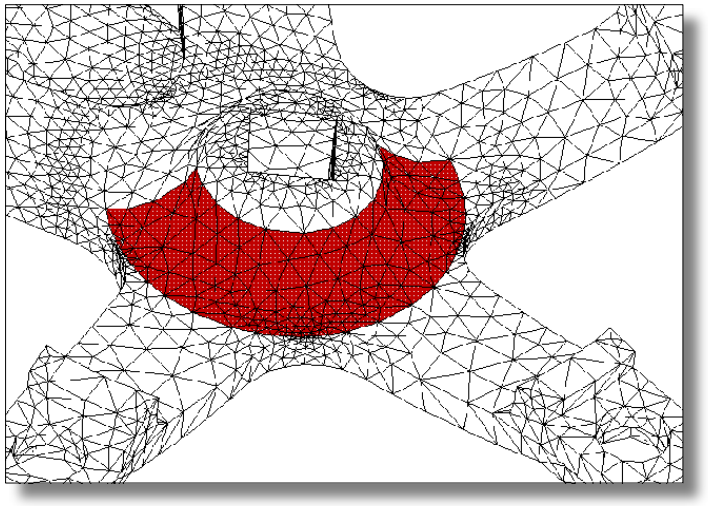
Steve Owen, 14th IMR, short course

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Parametric Space Mesh Generation

•Use essentially the same isotropic methods for 2D mesh generation, except distances and angles are now measured with respect to the local metric tensor **M**(X).
•Can use Delaunay (George, 99) or Advancing Front Methods (Tristano,98)





Steve Owen, 14th IMR, short course

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Metric (differential geometry)

A Riemannian metric $\mathbf{g}$ on a smooth manifold M is a smoothly chosen inner product $\mathbf{g}_x : T_xM \times T_xM \to \mathbb{R}$ on each of the tangent spaces T_xM of M satisfying:

(1) $\mathbf{g}(u, v) = \mathbf{g}(v, u), \forall u, v \in T_xM$;

(2) $\mathbf{g}(u, v) \geq 0, \forall v \in T_xM$; and

(3) $\mathbf{g}(v, v) = 0 \iff v = 0$.

A special case, when $M = \mathbb{R}^d$, the real d-dim Euclidean space, $\mathbf{g}$ is a $d \times d$ symmetric positive definite matrix.

Carl Friedrich Gauß (1777-1855)

A metric allows defining distances, areas and angles on the manifold M. The length of a vector $v \in T_xM$:

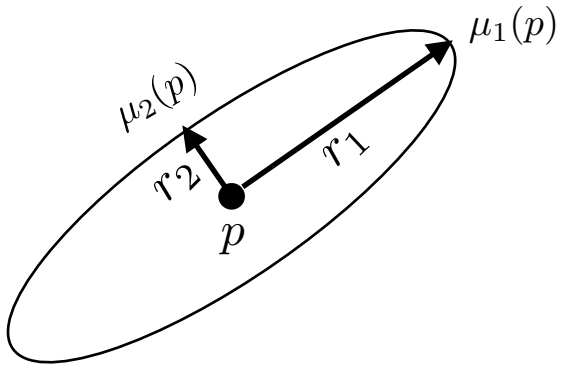
$$\|v\| = \sqrt{\mathbf{g}(v, v)} = \sqrt{v^T \mathbf{g} v}. \text{ in matrix form}$$

The angle θ between two vectors $u, v \in T_xM$ is:

$$\cos(\theta) = \frac{\mathbf{g}(u, v)}{\|u\|\|v\|}$$

Let γ be a curve from a to b in M, the curve length

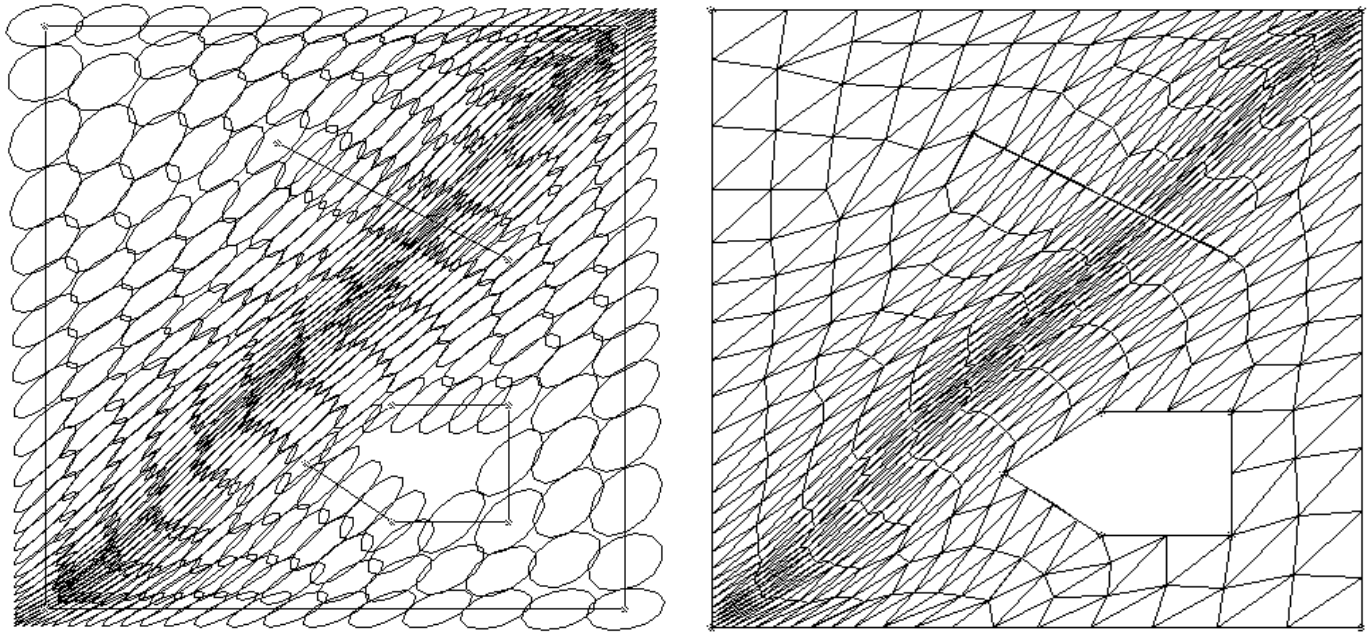
$$l_{\mathbf{g}}(\gamma) = \int_a^b \sqrt{\|\gamma'(t)\|} \mathrm{dt}, \ \mathrm{t} \in [\mathrm{a}, \mathrm{b}].$$



Bernhard Riemann (1826-1866)

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Metric-based anisotropic mesh generation/adaptation



- Ellipses packing Li Teng Ungor (99) , Yamakawa Shimada (03)
- Anisotropic Delaunay refinement Borouchaki George et al. (97)
  Frey Alauzet (2004), Dobrzynsk Frey (08)
- Continuous mesh Loseille Alauzet (09)
- Anisotropic mesh optimization

Pliant method Bossen Heckbert (96)
Li Shepard Beall (2005)

- Anisotropic Voronoi diagrams, Labelle Shewchuck (03), Boissonnat et al(08).

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Geometry approximation

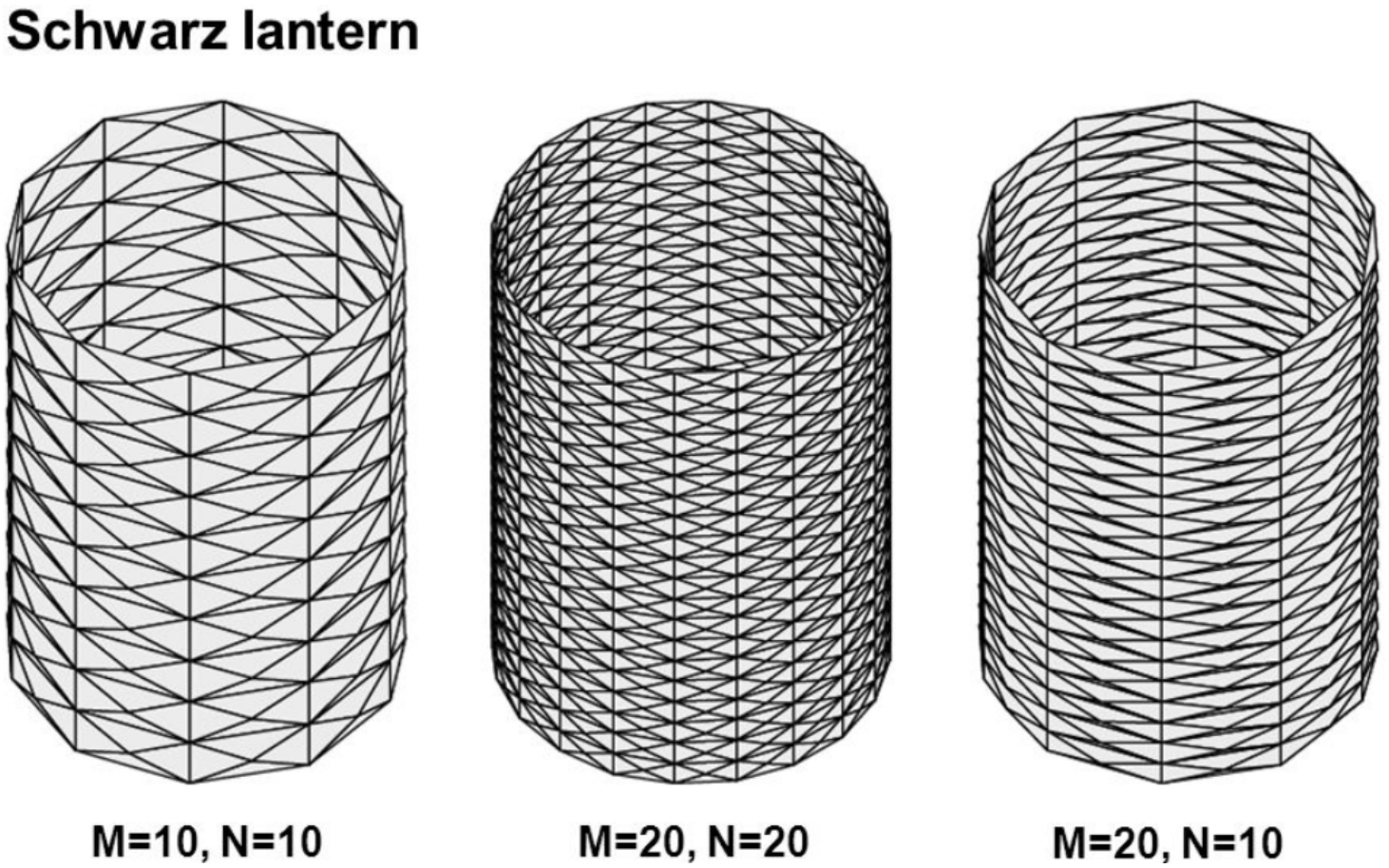
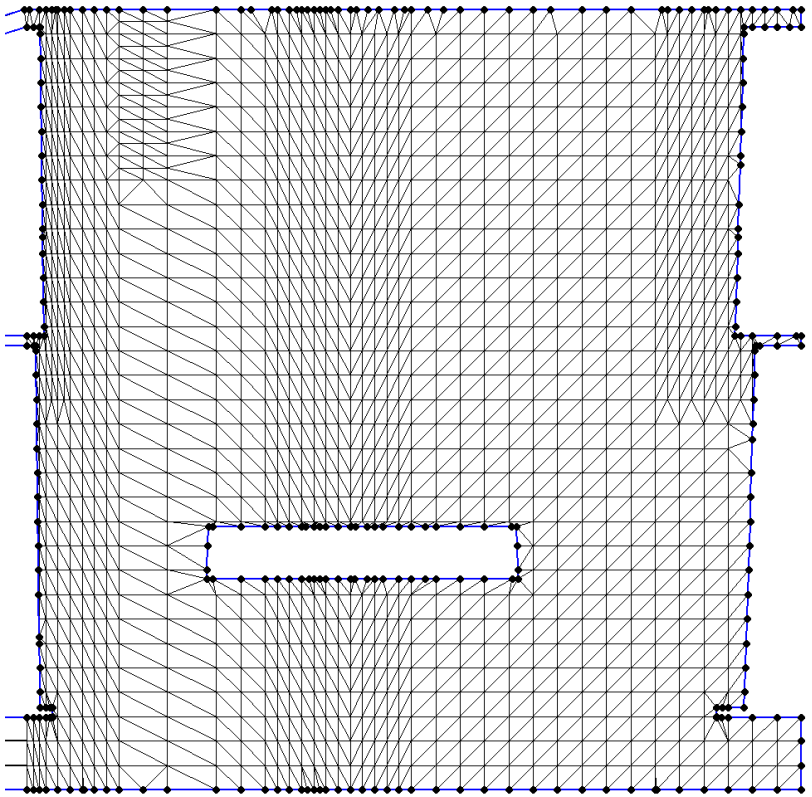


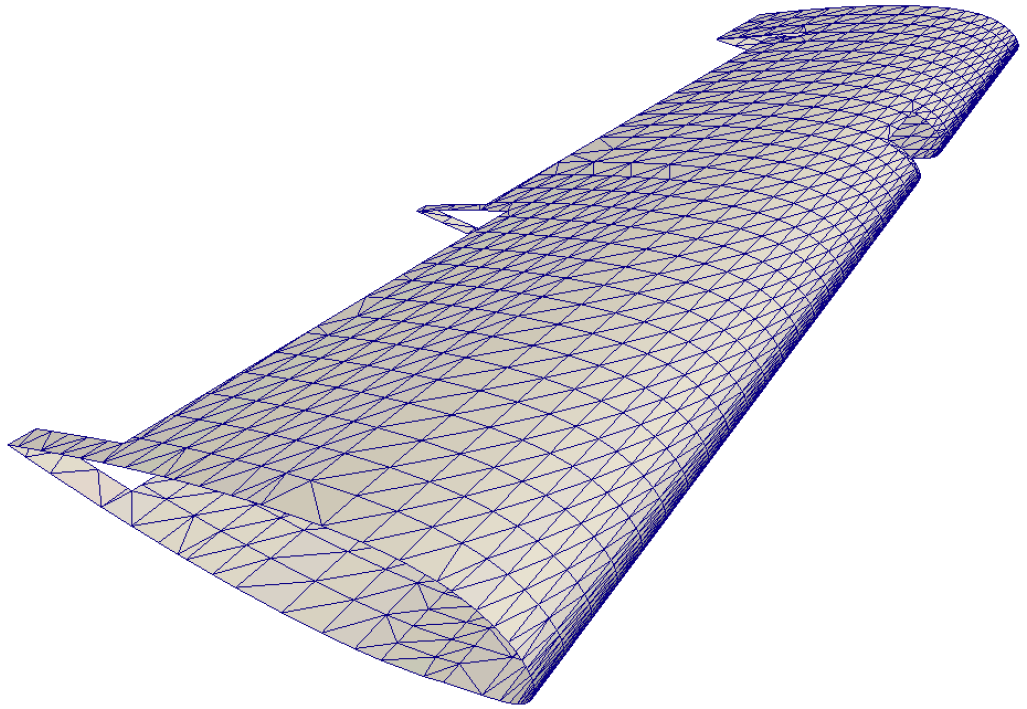
Figure: Schwartz's lantern: an example of Hausdorff convergence only.

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Example: wing 33



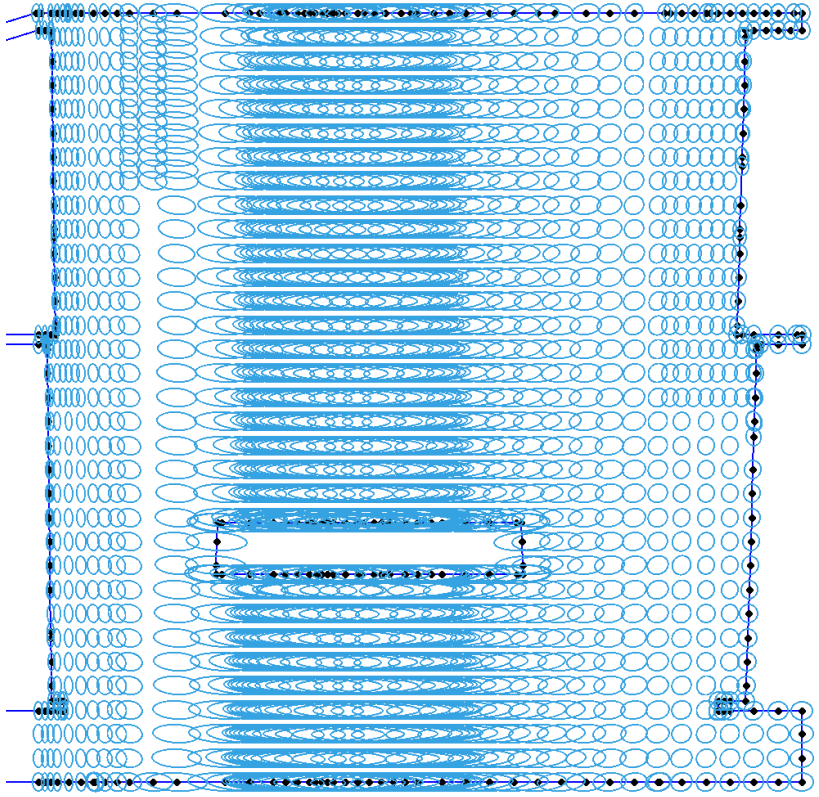
uv



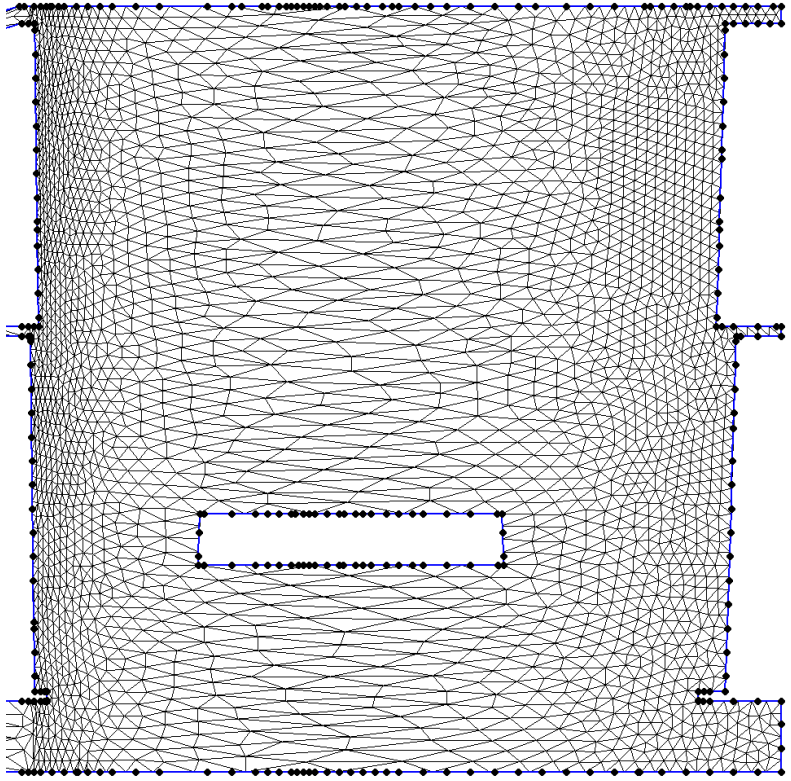
xyz

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Example: wing 33



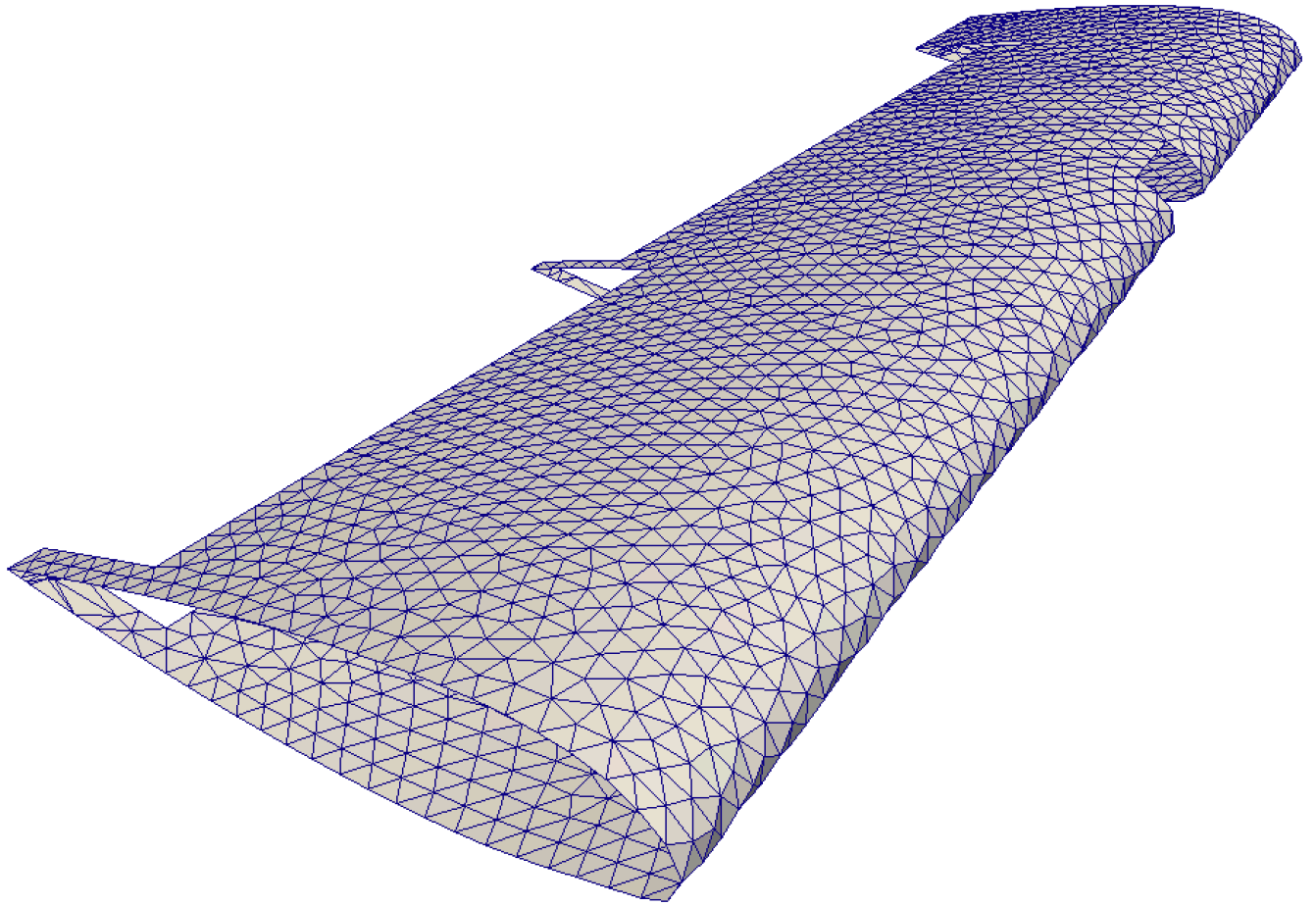
Riemann metric



anisotropic mesh

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

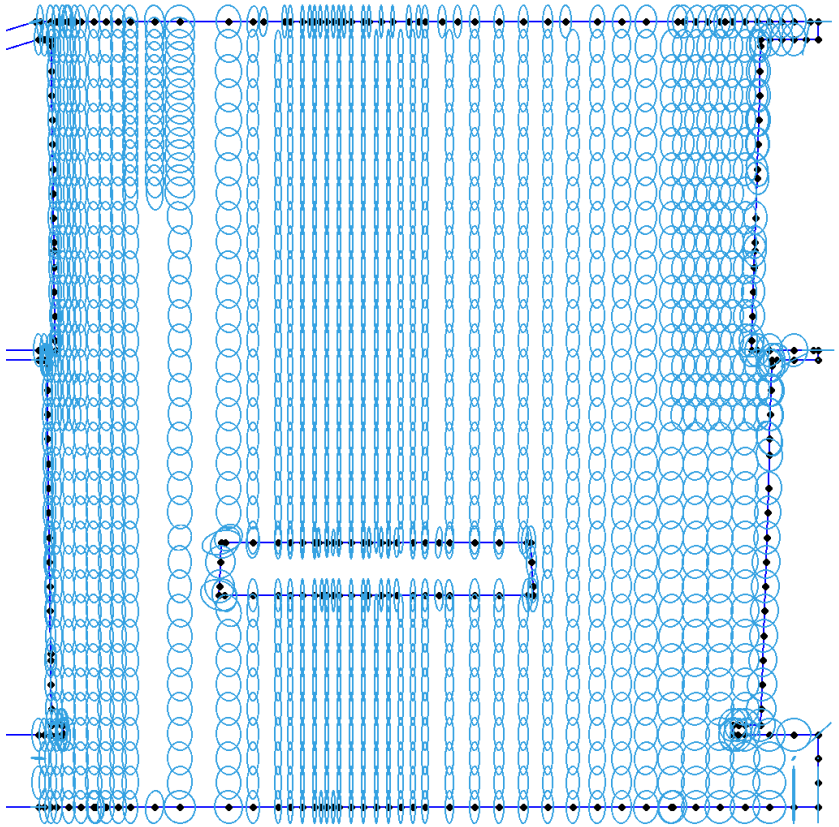
Example: wing 33



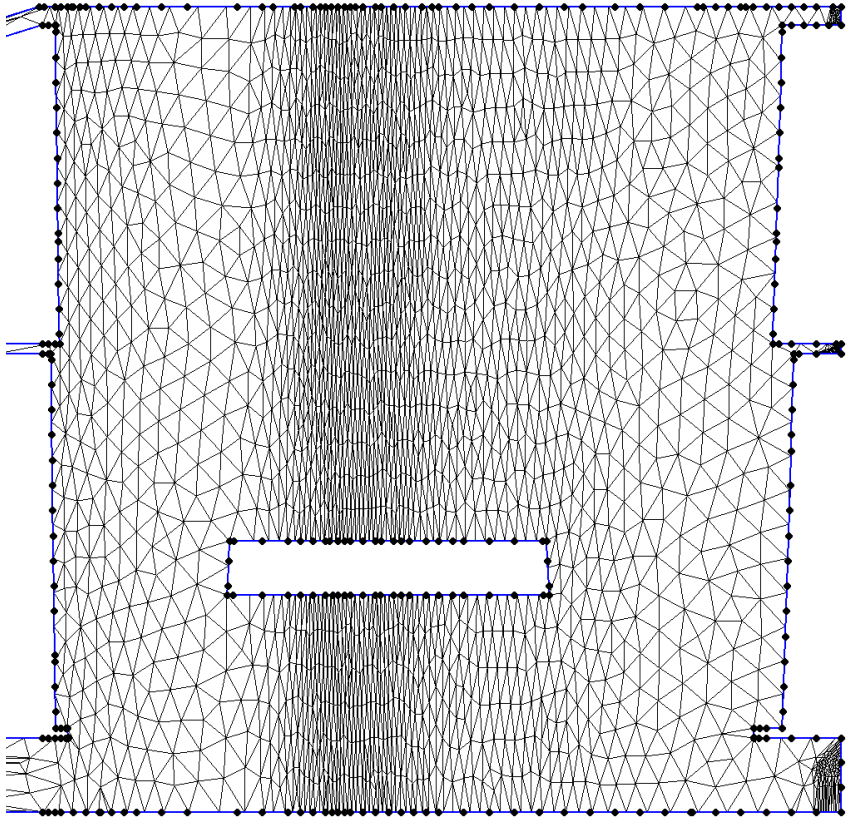
surface mesh from anisotropic Riemann metric

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Example: wing 33



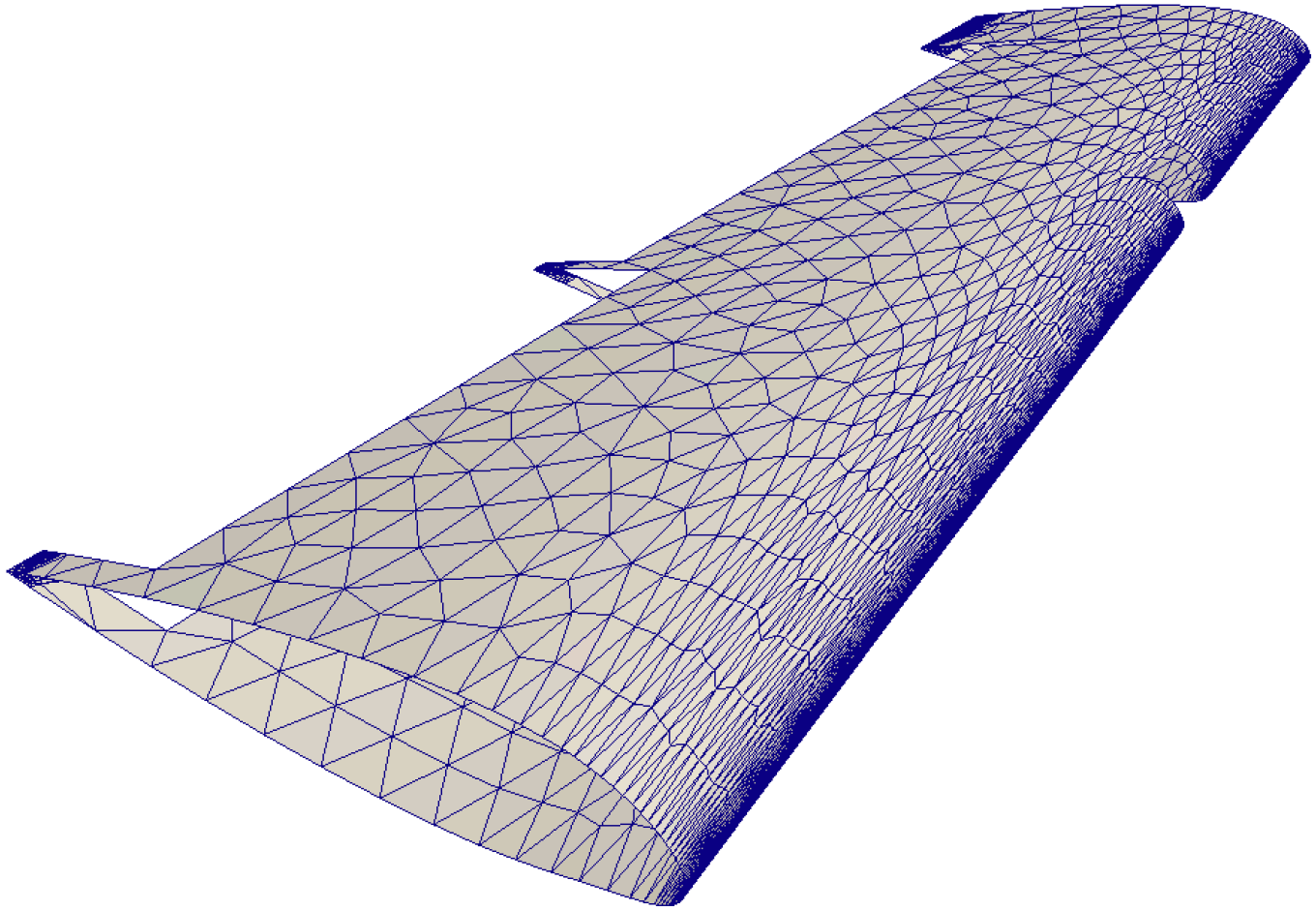
Riemann + curvature metric



anisotropic mesh

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Example: wing 33



surface mesh from anisotropic Riemann + curvature metric

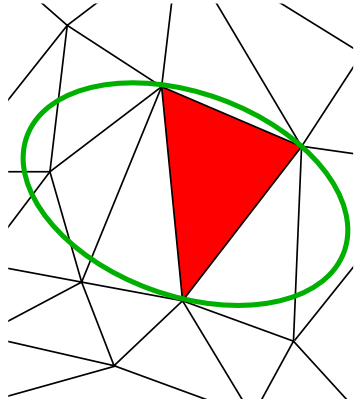
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Anisotropic Delaunay refinement

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Non uniform metric

An anisotropic Voronoi diagram



Delaunay triangulation is not globally defined. No “empty circum-ellipse” property.

F. Labelle and J. Shewchuk, 2016

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Labelle and Shewchuck approach

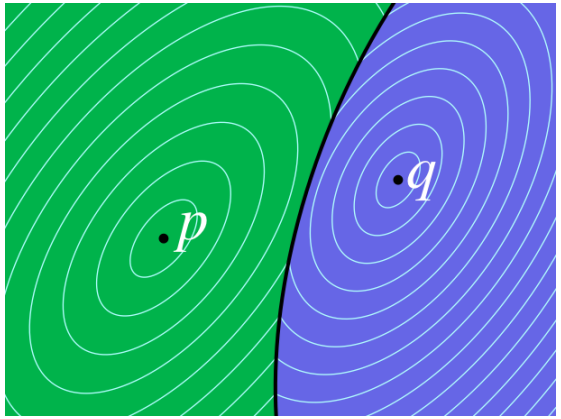
Let $D \subset \mathbb{R}^d$ be a domain
with a metric field defined on D :
$\forall p \in D \longrightarrow M_p$.

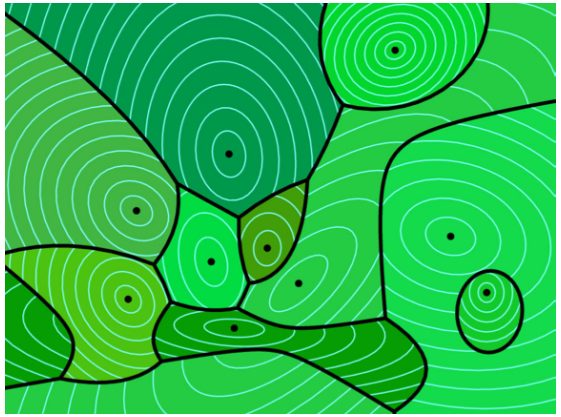
$$d_p(x, y) = \sqrt{(x - y)^t M_p (x - y)}$$

Anisotropic Voronoi diagram

P a set of sites in D
$\forall p \in P$, Voronoi cell $V(p)$

$$V(p) = \{x \in \mathbb{R}^d \quad : \quad d_p(p, x) \leq d_q(q, x), \\ \forall q \in P, q \neq p\}$$





Cells are not connected.
The dual is not a triangulation.
Labelle and Shewchuck approach :
refine the set of sites until
the dual is a triangulation.
Works only in 2D.

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

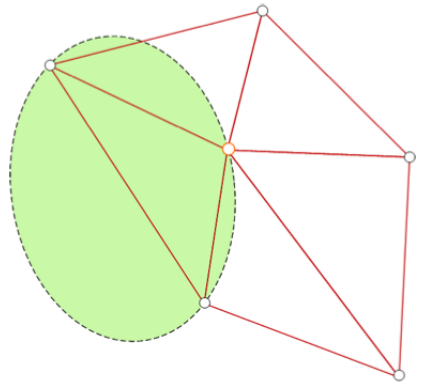
Locally uniform Anisotropic Delaunay Mesh (M. Yvinec, INRIA)

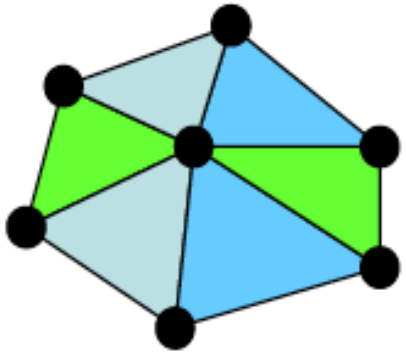
Build a mesh such that:
the star of each vertex is Delaunay
and well shaped
wrt the metric at that vertex.

V set of vertices, $v \in V$:
M_v metric at v
$\mathrm{Del}_v(V)$ Delaunay triangulation of V
computed with metric M_v
S_v: the star of v in $\mathrm{Del}_v(V)$

Overview of the meshing algorithm

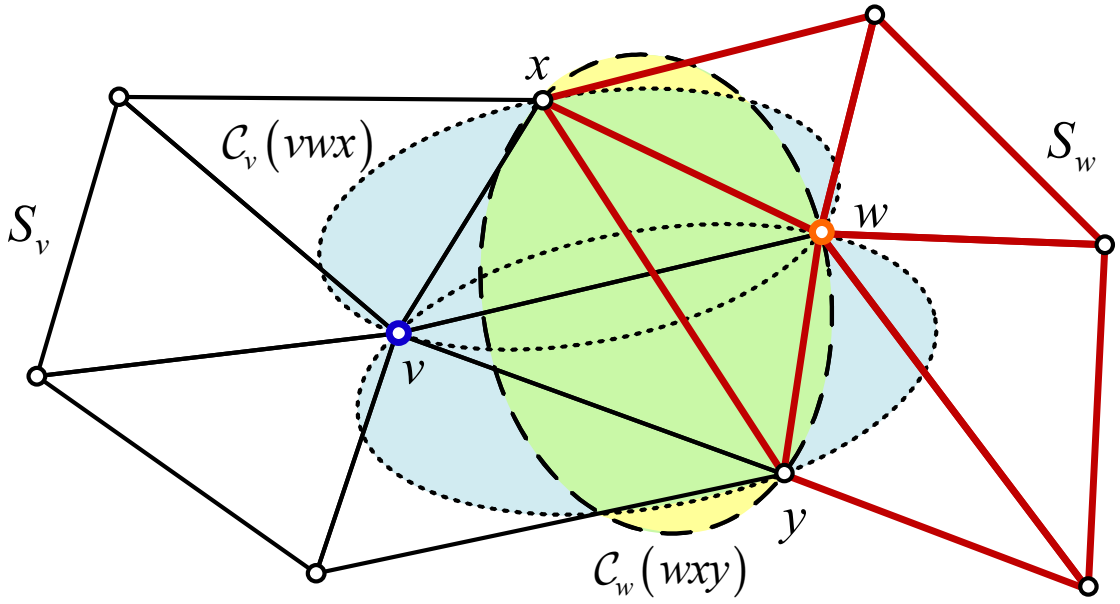
- Maintain the set of stars $S(V) = \{S_v : v \in V\}$
- Refine V until stars are consistent





© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Locally uniform Anisotropic Delaunay Mesh (M. Yvinec, INRIA)



Inconsistency : some simplex τ with vertices $\{v, w, \ldots\}$ appears in star S_v but not in star S_w.

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Anisotropic Delaunay Refinement (M. Yvinec, INRIA)

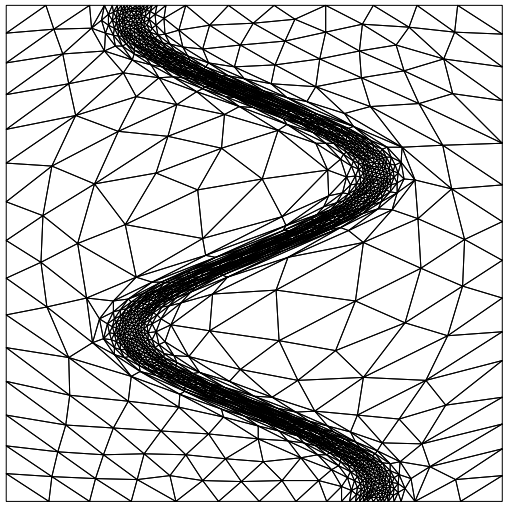
V current set of sites, $S(V) = \{S_v : v \in V\}$ the star set
Domain Ω: each star S_v is restricted to Ω
i.e. $S_v = \{\tau \in \mathrm{Del}_v(V) : v \in \tau \text{ and } c_v(\tau) \in \Omega\}$

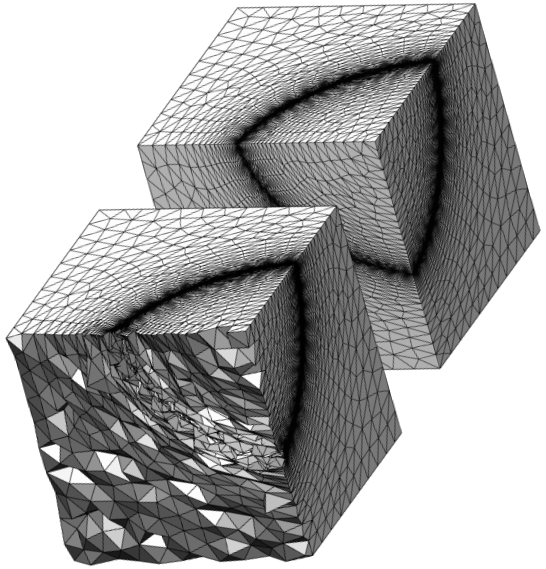
Apply following refinement rules with priority order:

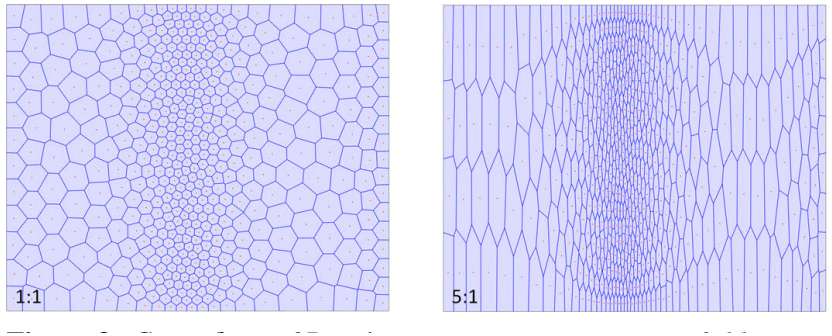
1. Sizing field - Distorsion
   While $\exists \tau \in S_v$ s.t. $r_v(\tau) \geq \alpha\, \mathrm{lf}(c_v(\tau))$, refine τ
2. Radius-edge ratio
   While $\exists \tau \in S_v$ s.t. $\rho_v(\tau) \geq \rho_0$, refine τ
3. Slivers
   While $\exists$ a sliver $\tau \in S_v$ $(\rho_v(\tau) \leq \rho_0,\ \sigma_v(\tau) \leq \sigma_0)$, refine τ
4. Inconsistencies
   While $\exists$ an inconsistent simplex $\tau \in S_v$, refine τ

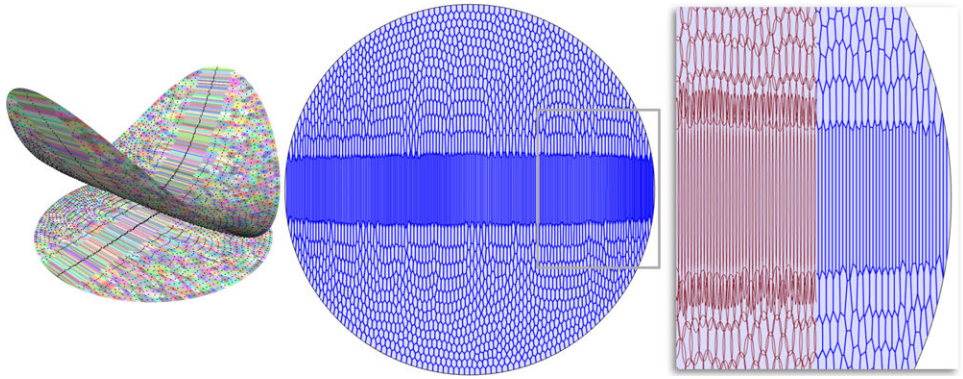
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Anisotropic mesh improving







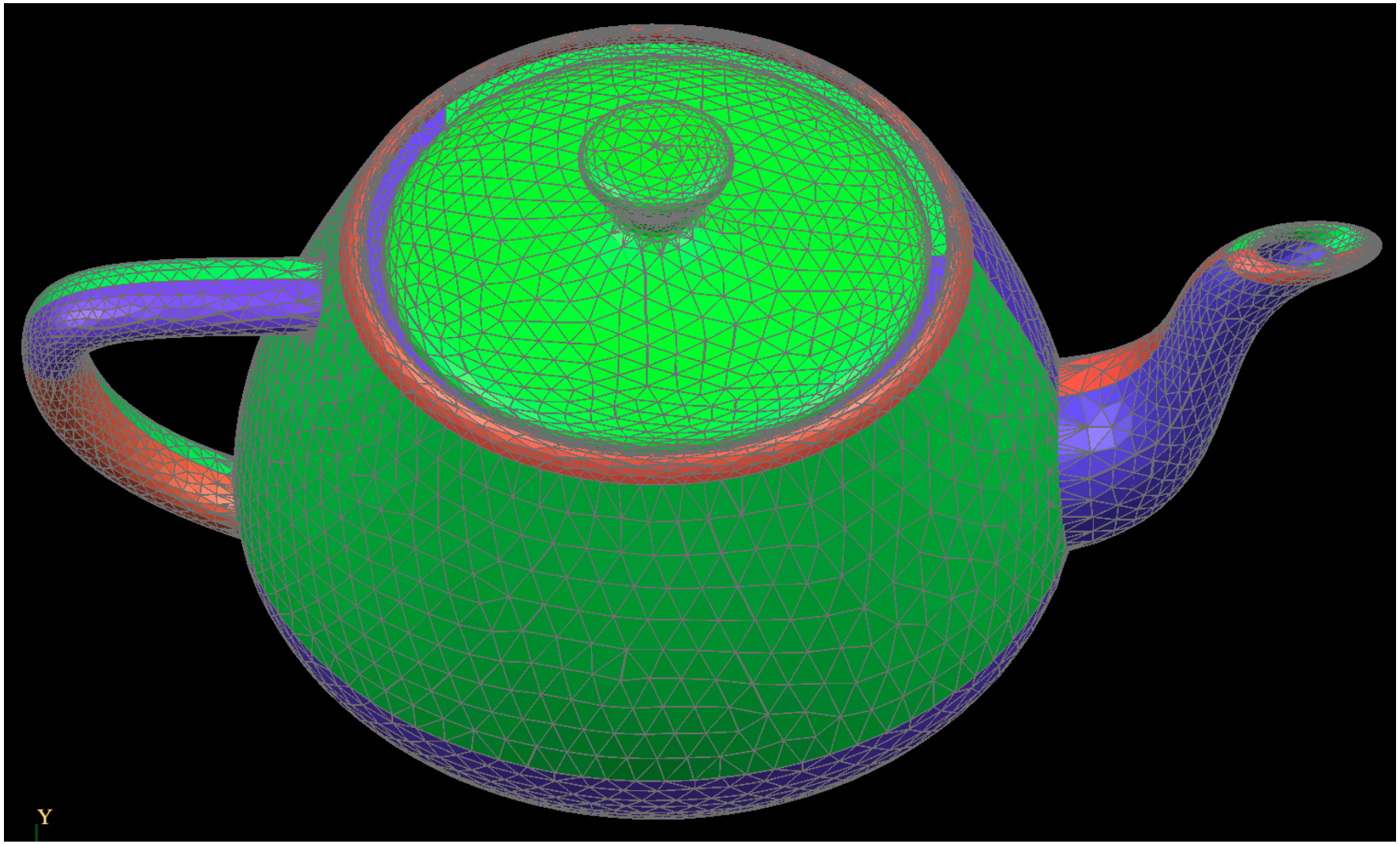


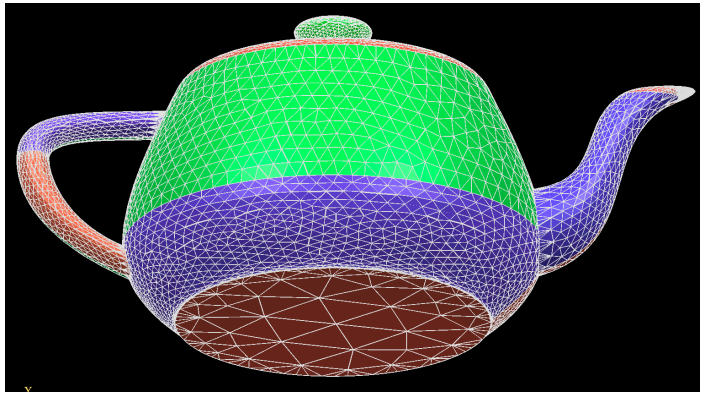
X. Fu, Y. Liu, J. Snyder, B. Guo, Anisotropic Simplical Meshing Using Local Covex Functions. ACM Transactions on Graphics, Volume 33, Issue No. 6, Article Number 182, pp 1—11, 2014

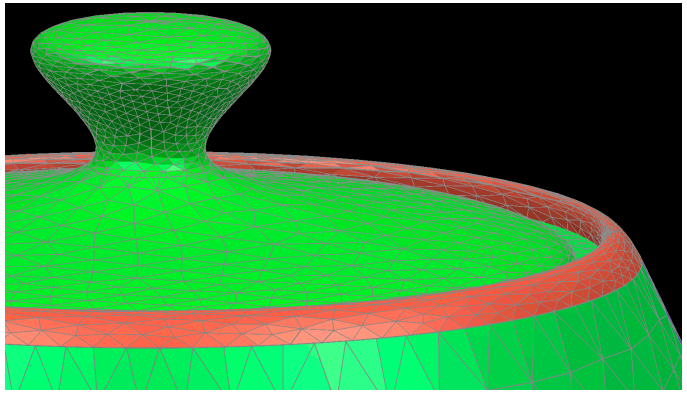
M. Budniskiy, B. Liu, F. De Goes, Y. Tong, P. Alliez, M. Desbrun, Optimal Voronoi Tessellations with Hessian-based Anisotropy. ACM Transactions on Graphics, Volume 35, Issue No. 6, Artical No. 242, pp 1—12, 2016

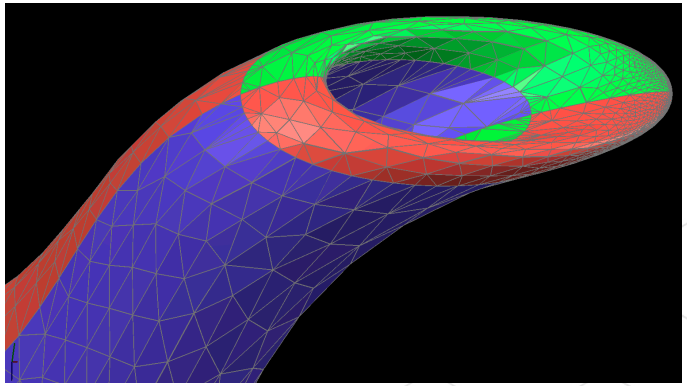
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Teapot (step)



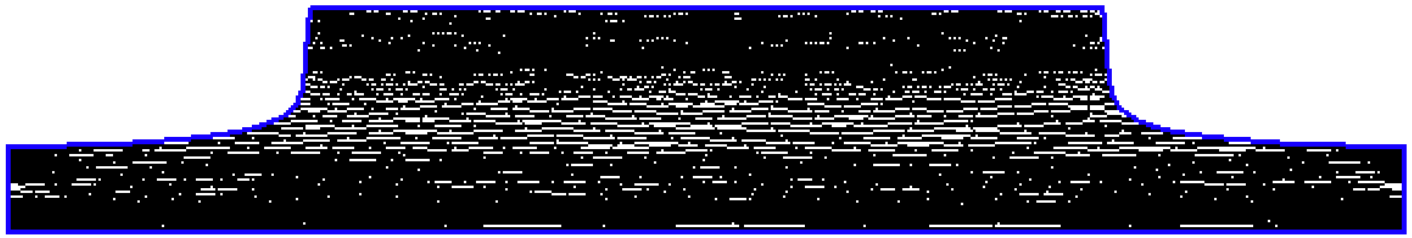


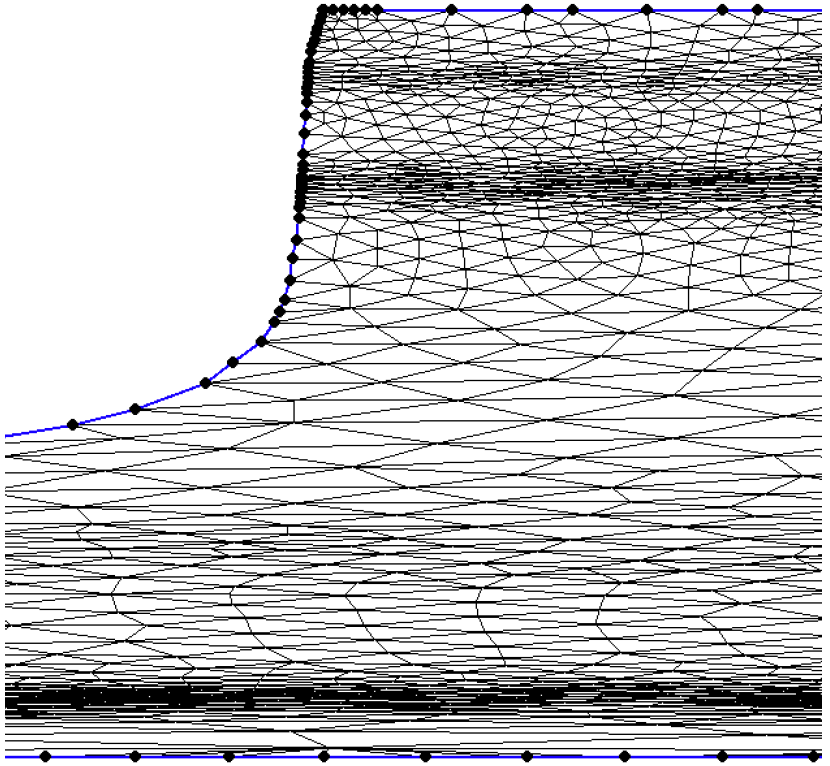


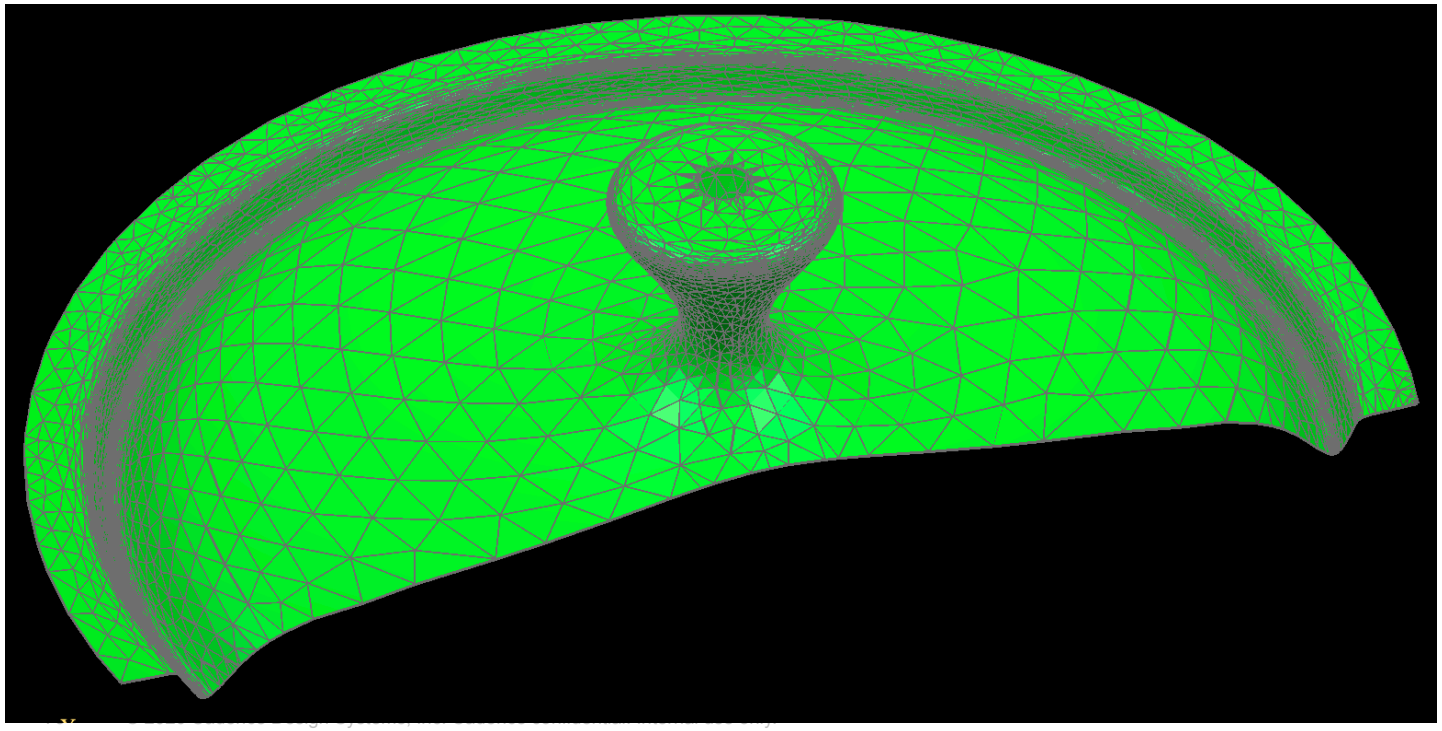


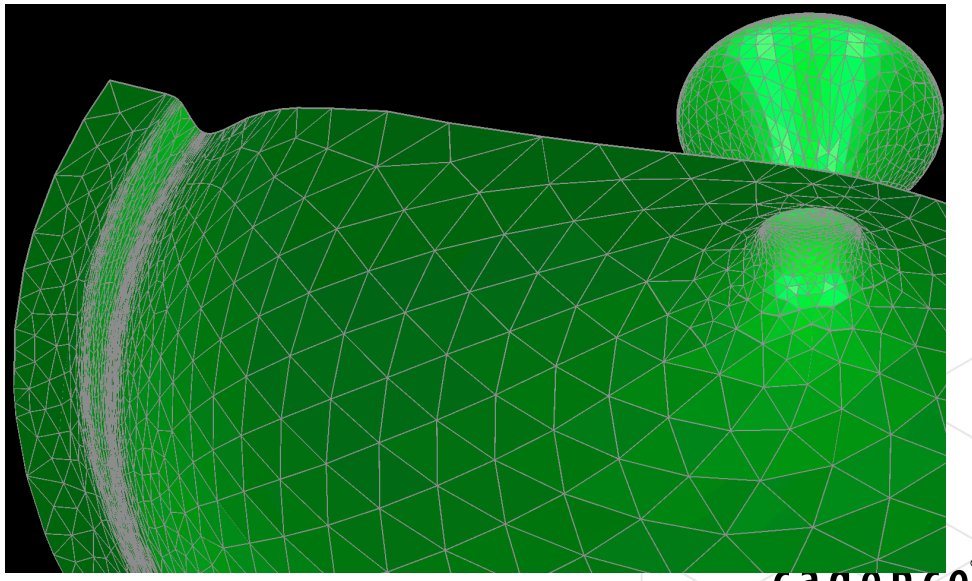
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

surface patch 11

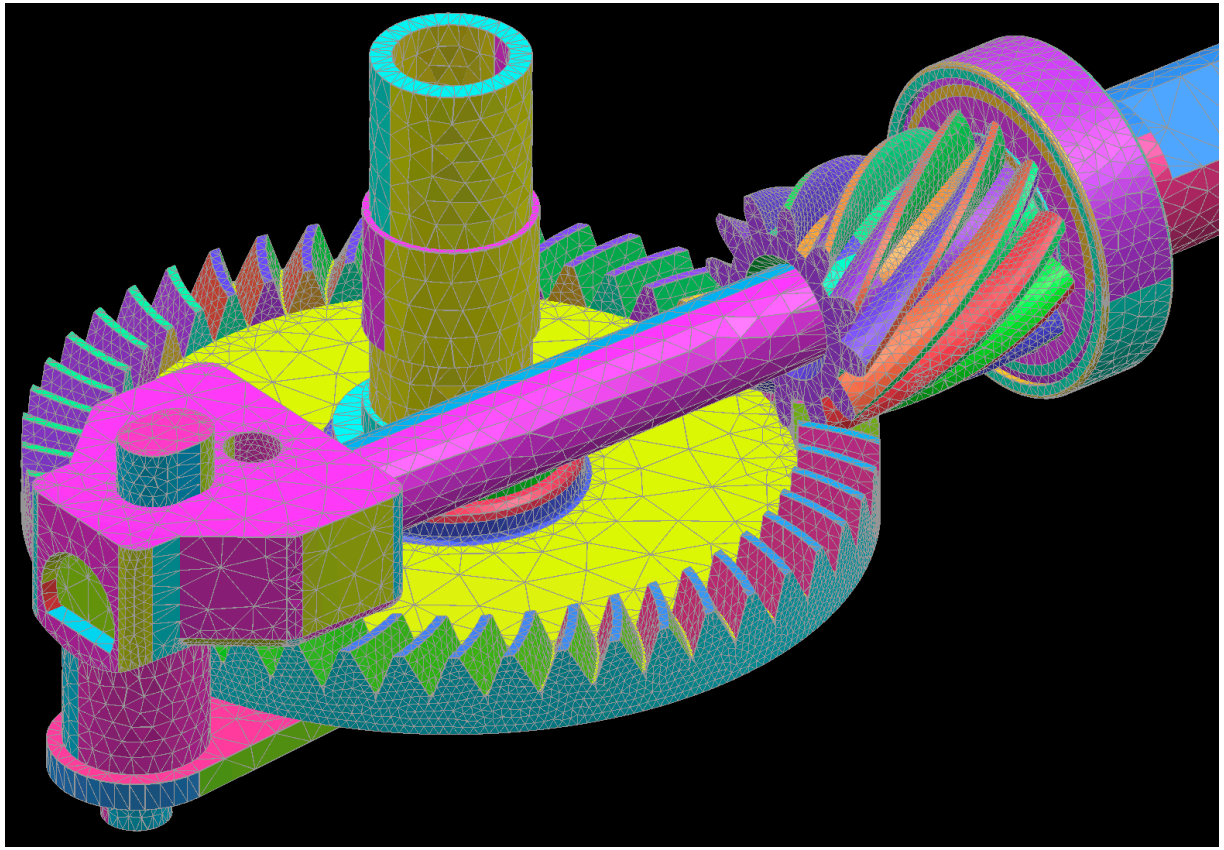


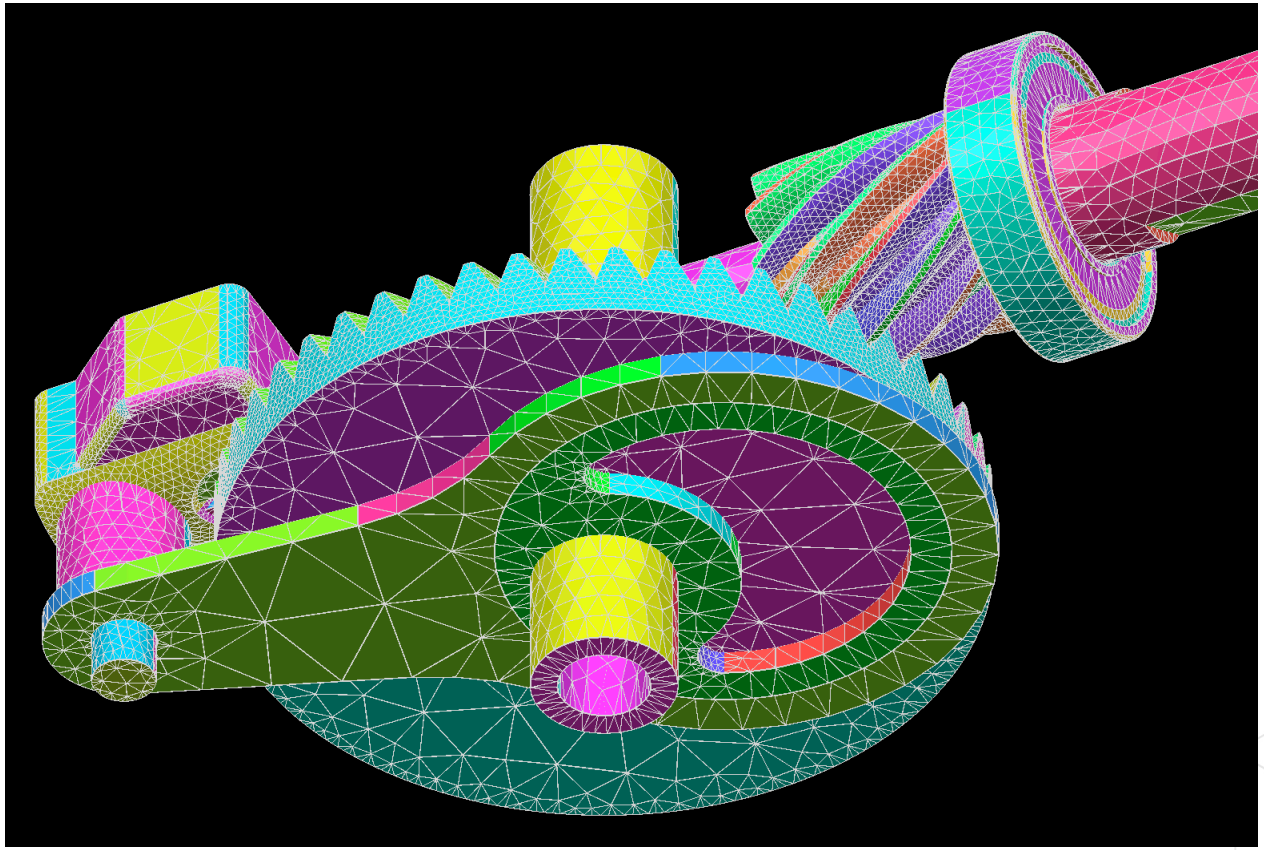




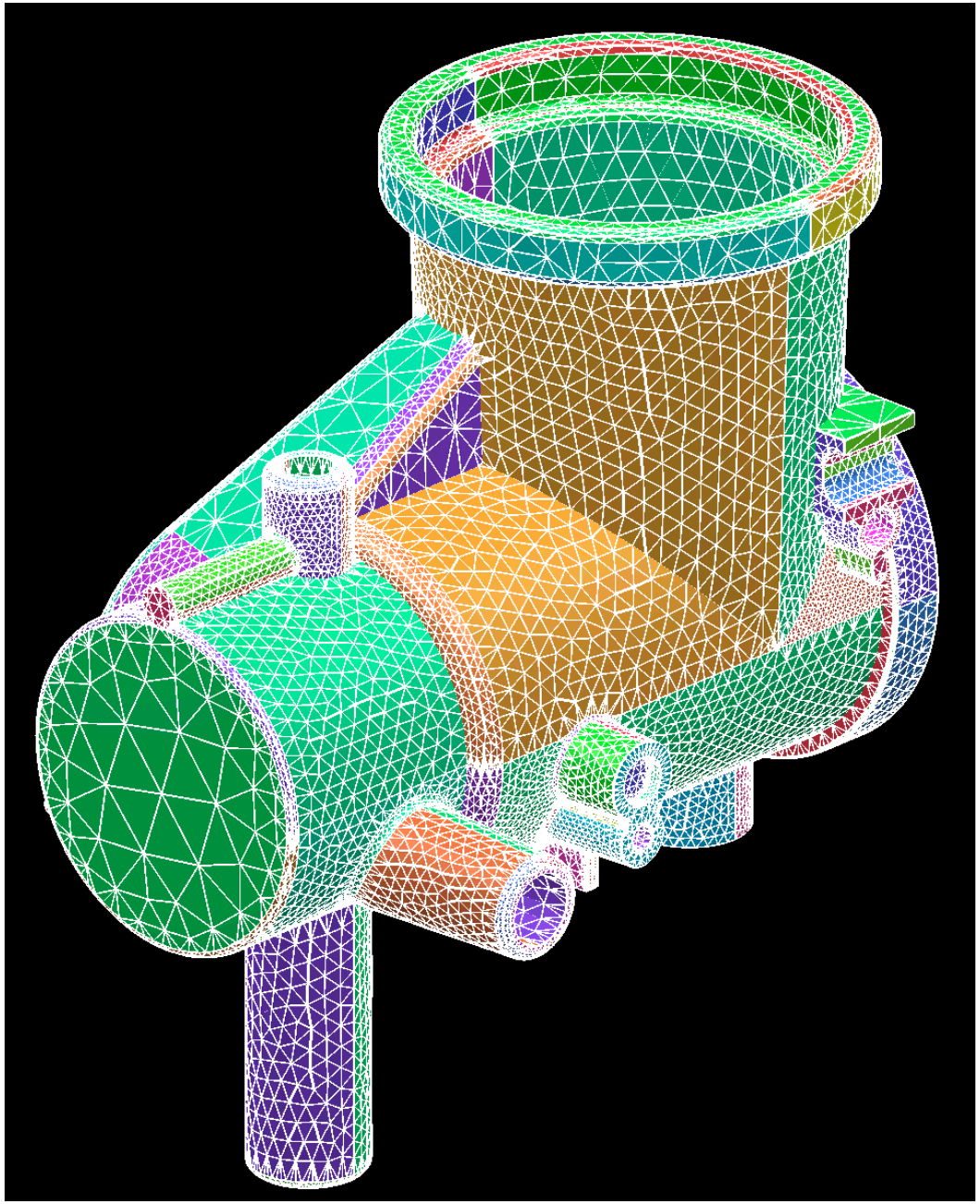


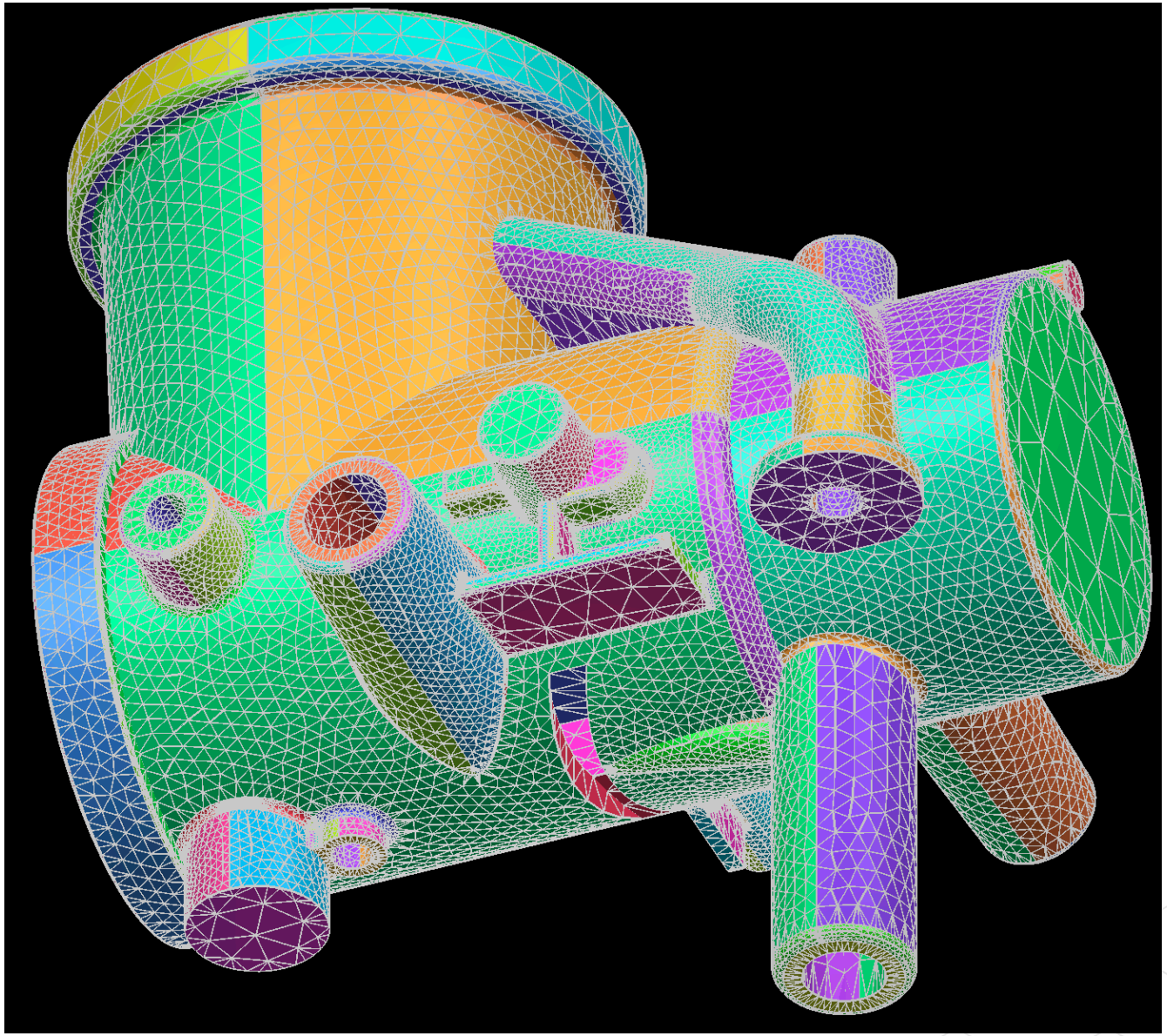
cadence®



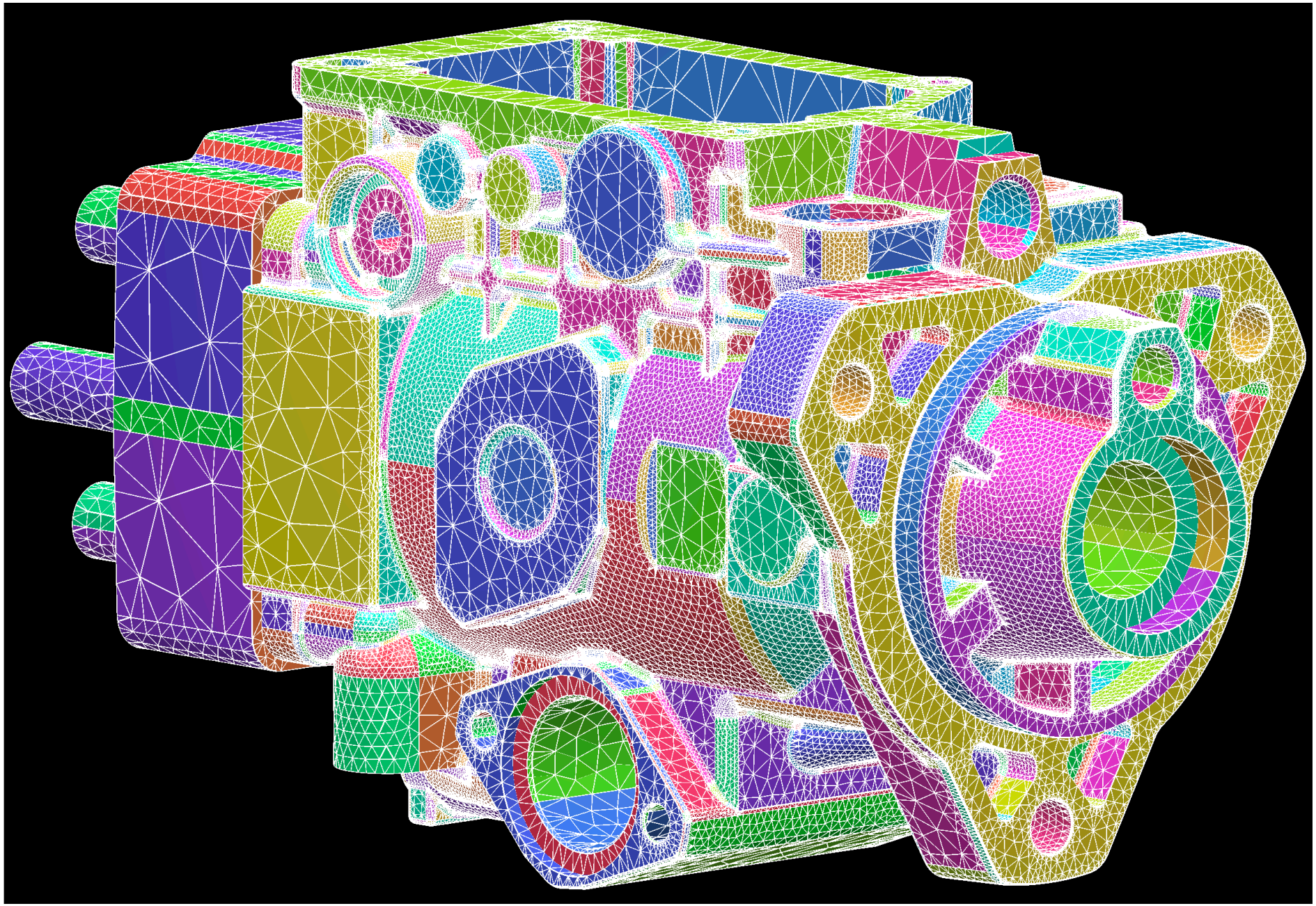


© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.





© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.



© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Joint work with Prof. Lei and Prof. Gu, IMR 2023

Surface Remeshing Based on Conformal Uniformization

Key Idea : Convert 3D meshing to 2D

1. Flatten a curved surface onto a planar domain using conformal(angle-preserving) mappings;
2. Generate a high quality mesh on the planar domain;
3. Pull back the triangulation from the planar domain to the curved surface.

A conformal mapping maps the planar Delaunay triangulations to the geodesic Delaunay triangulations on the surface. The sampling density on the plane can be adapted, such that the sampling on the surface normal cycle is uniform. This produces high quality surface meshes.

Gu talk, IMR 2023

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Discrete surface Ricci flow

Definition (Discrete Surface Ricci Flow)

Given a marked surface (S, V) with a polyhedral metric d and a triangulation T, suppose the target Gaussian curvature $\bar{K} : V \to \mathbb{R}$ is given, then the Ricci flow is defined as

$$\frac{d\lambda(v_i, t)}{dt} = \bar{K}(v_i) - K(v_i, t),$$

during the flow, T is updated to be Delaunay.

Gu talk, IMR 2023

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

Theorem (Discrete Surface Uniformization)

Given a polyhedral metric d on a closed marked surface (S, V), and target curvature $\bar{K} : V \to (-\infty, 2\pi)$, such that $\bar{K}$ satisfies the Gauss-Bonnet conditon $\sum K(v) = 2\pi\chi(S)$, there is a $\bar{d}$ discrete conformal to d, and $\bar{d}$ realized the curvature $\bar{k}$. $\bar{d}$ is unique update to a scaling, and can be obtained by the discrete surface Ricci flow.

The discrete uniformization theorem guarantees the existence and the uniqueness of the solution to the discrete surface Ricci flow, which can be obtained by optimizing the convex discrete surface Ricci energy.

Gu talk, IMR 2023

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

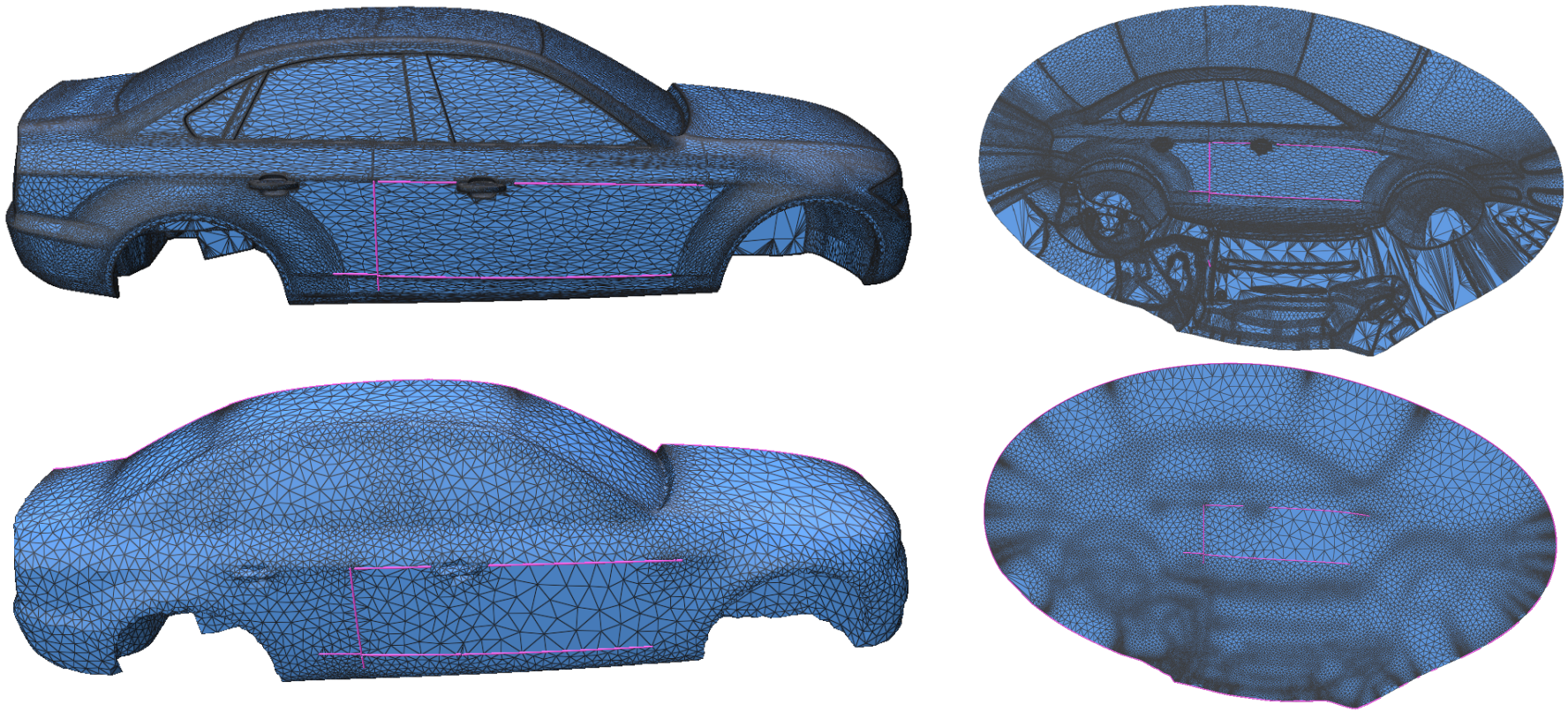


Figure: Conformal parameterization and remesh result.

© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

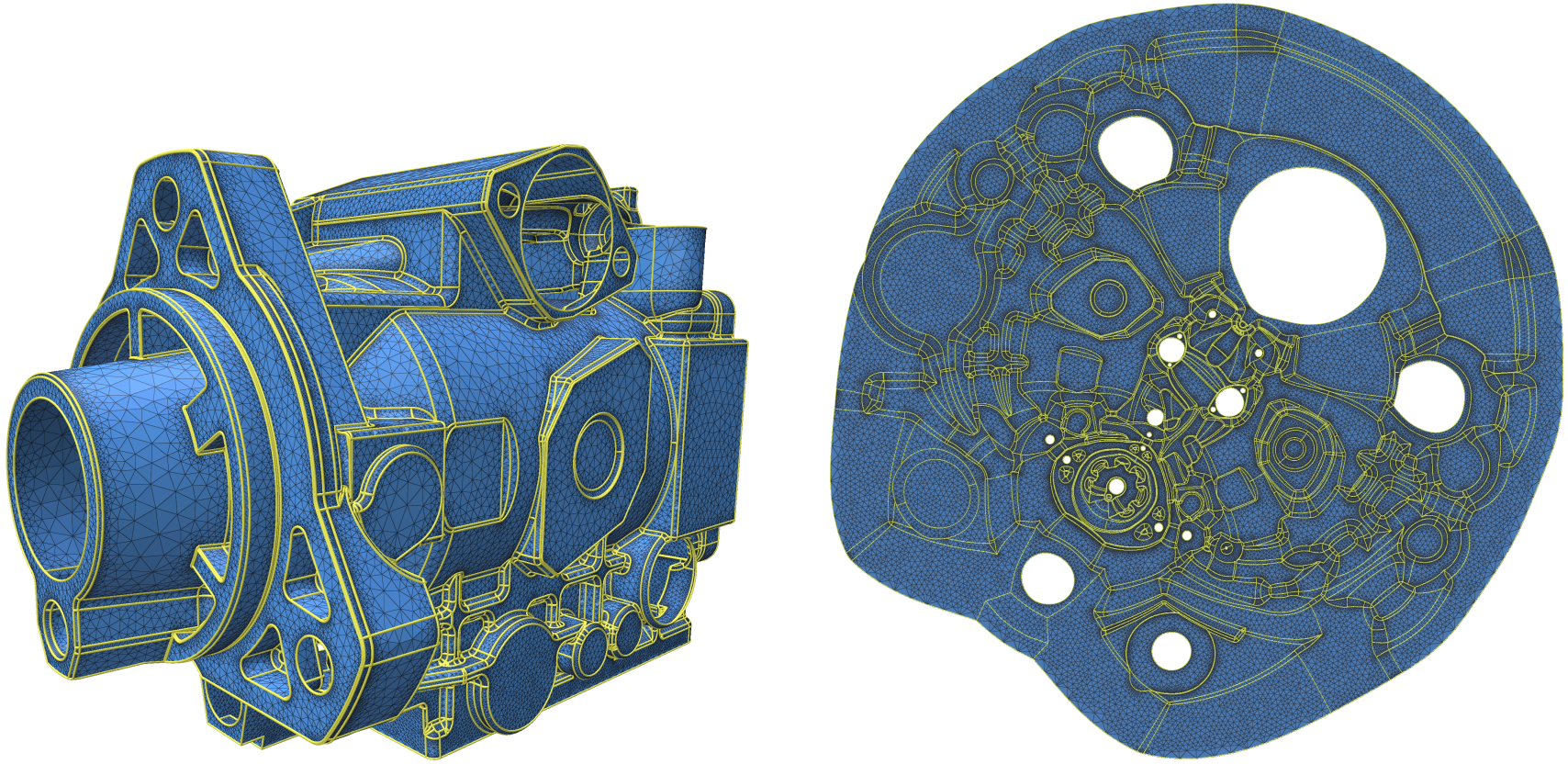
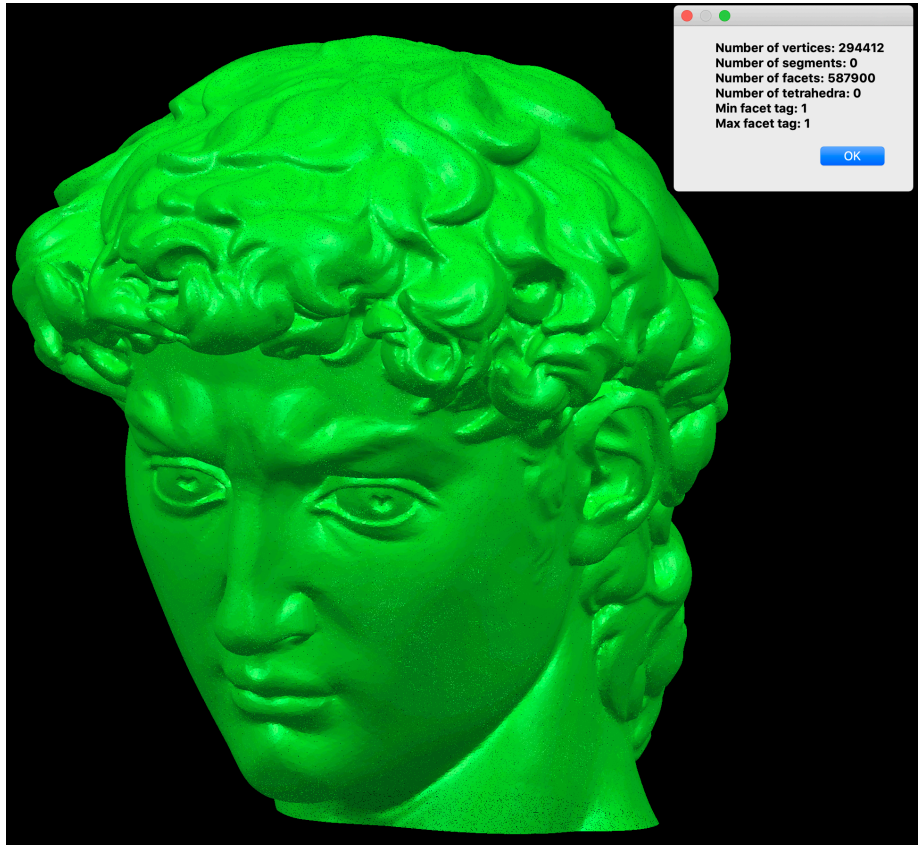
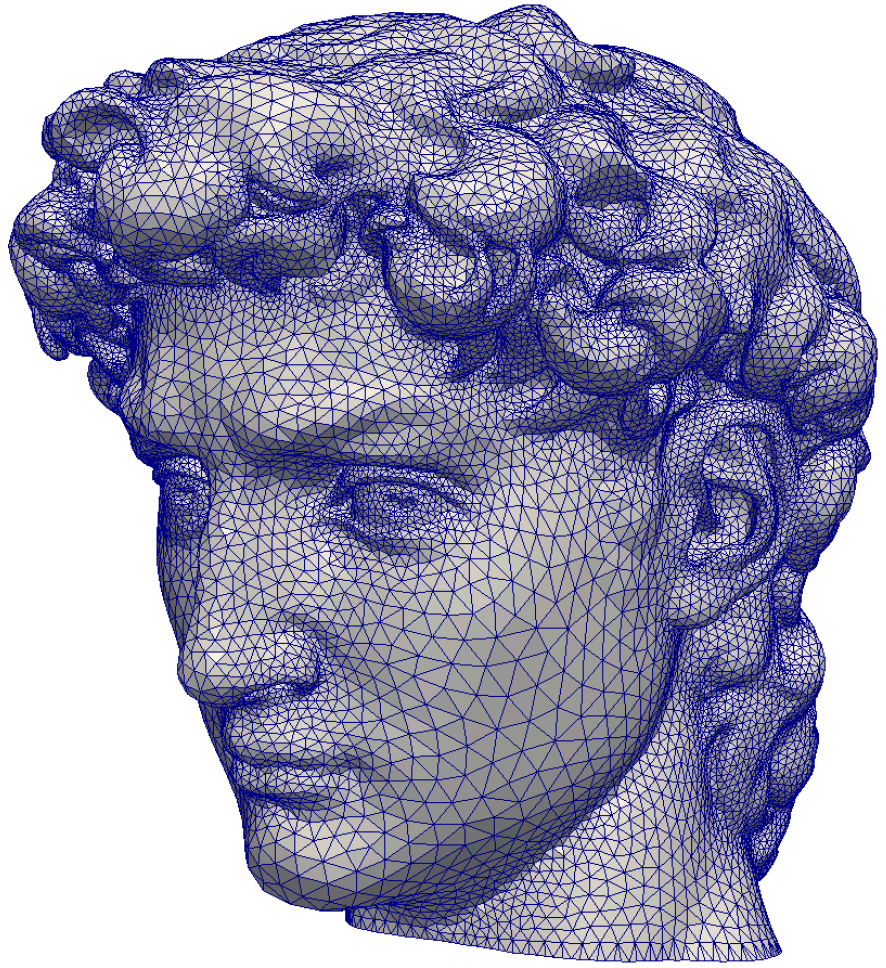


Figure: remesh result.

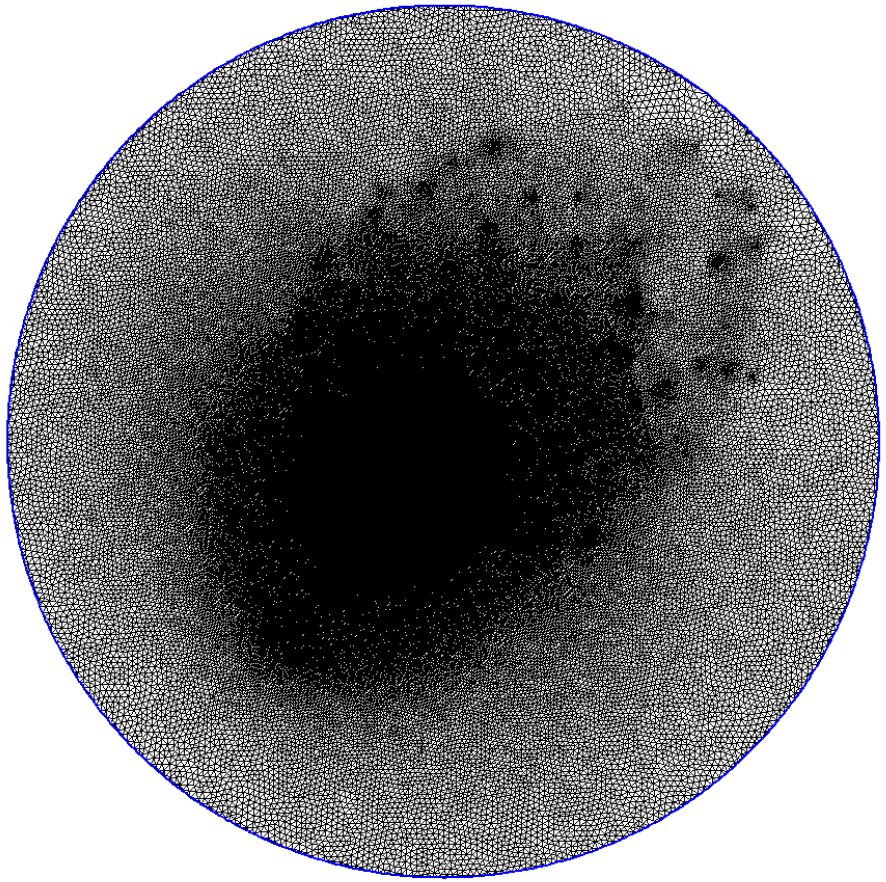
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

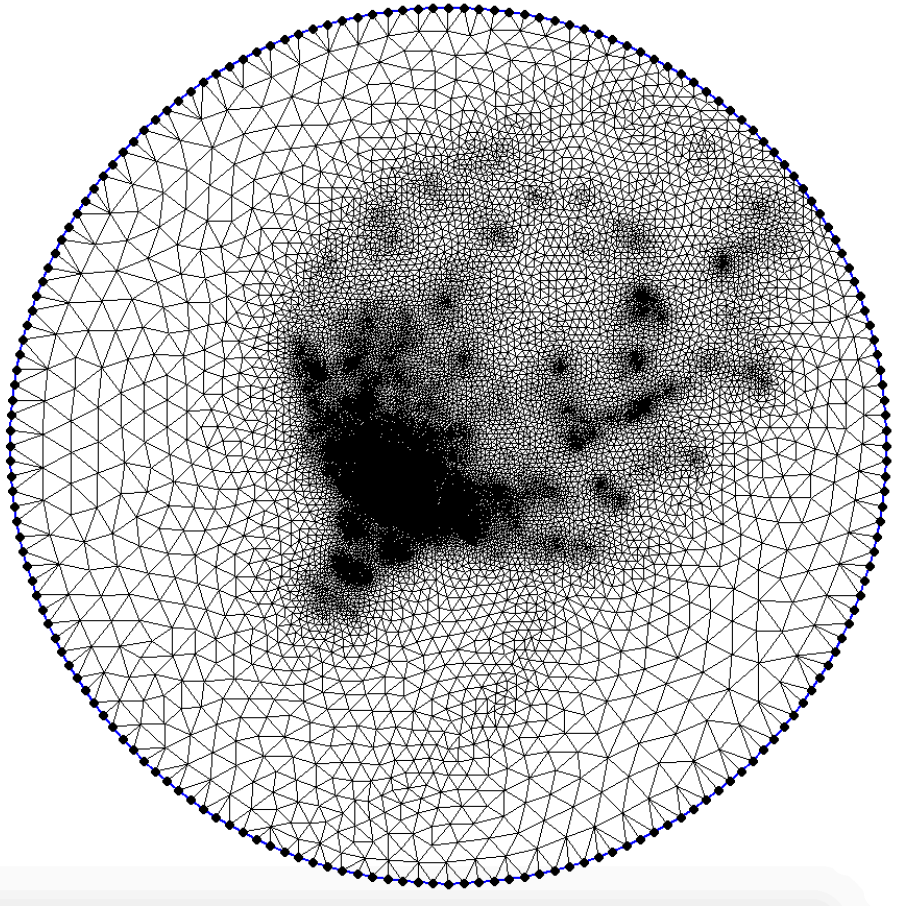




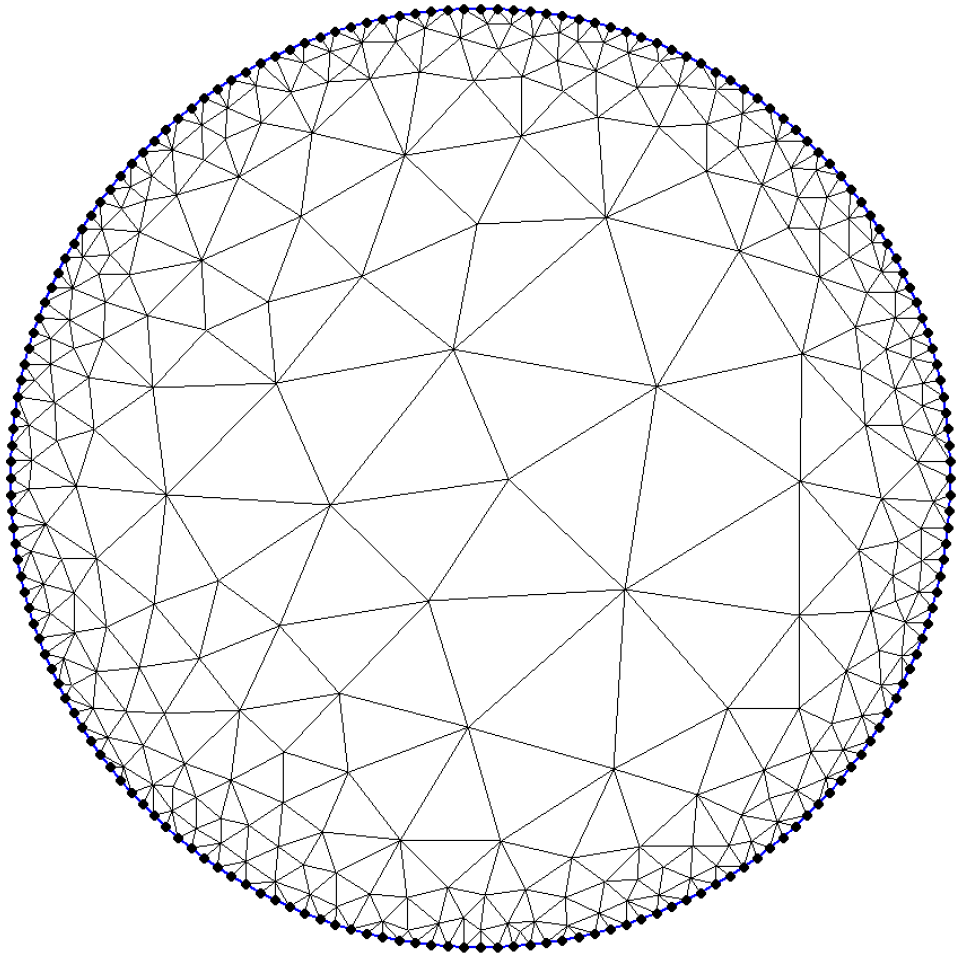
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

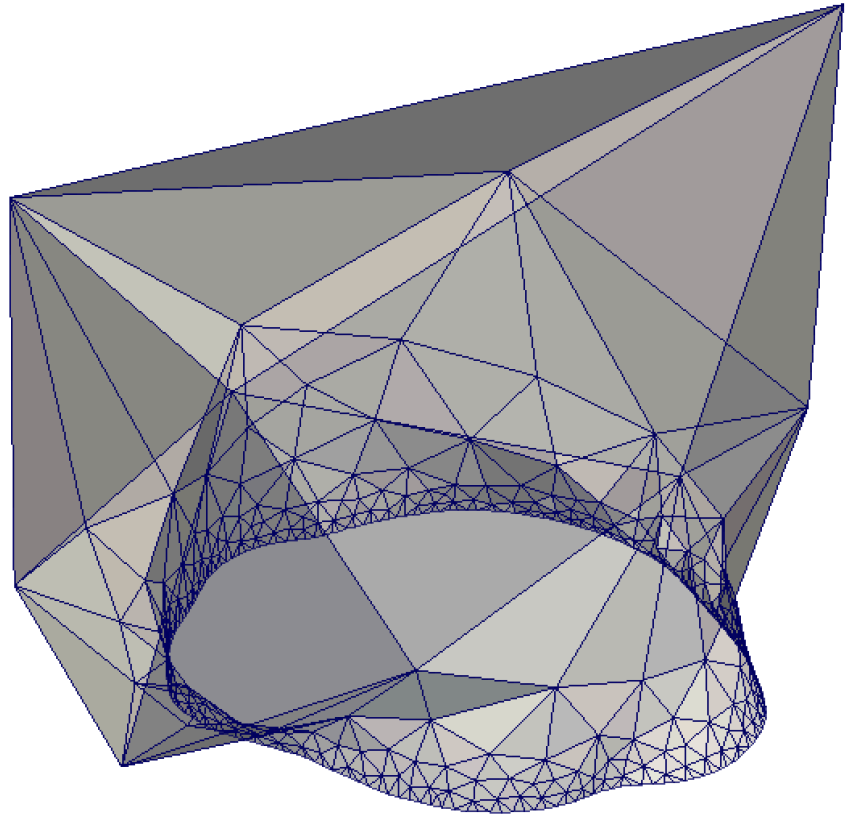
cadence





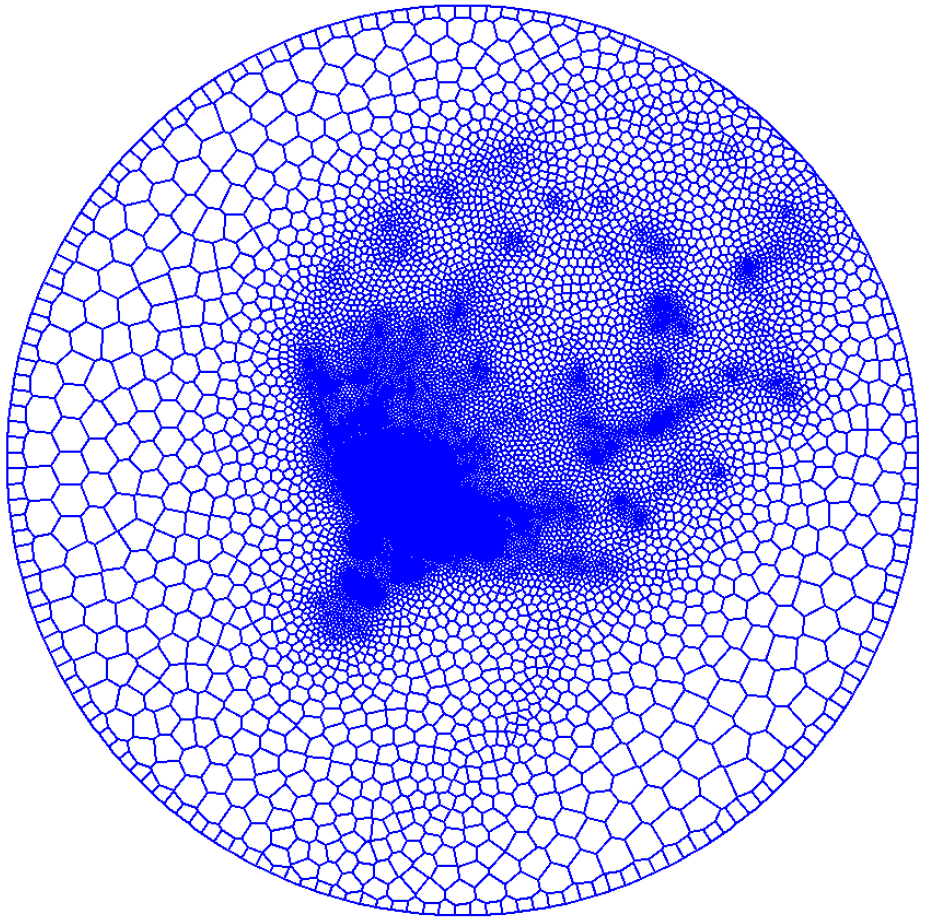
© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.





© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

cadence



© 2020 Cadence Design Systems, Inc. Cadence confidential. Internal use only.

cadence®

© 2020 Cadence Design Systems, Inc. All rights reserved worldwide. Cadence, the Cadence logo, and the other Cadence marks found at www.cadence.com/go/trademarks are trademarks or registered trademarks of Cadence Design Systems, Inc. Accellera and SystemC are trademarks of Accellera Systems Initiative Inc. All Arm products are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All MIPI specifications are registered trademarks or service marks owned by MIPI Alliance. All PCI-SIG specifications are registered trademarks or trademarks of PCI-SIG. All other trademarks are the property of their respective owners.