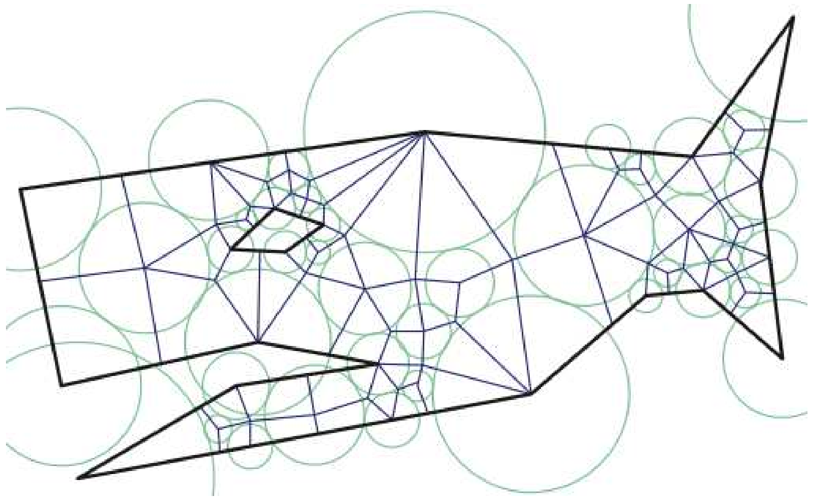
Theoretically Guaranteed Mesh Generation– In Practice

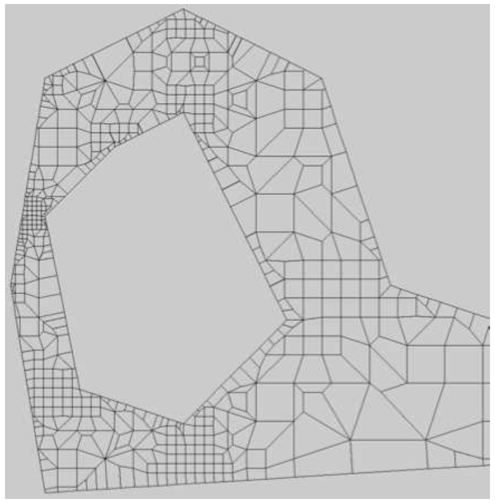
Jonathan Richard Shewchuk
UC Berkeley

Short Course
Journées de Géométrie Algorithmique
Marseille–Luminy, 11–12 March 2010

Goal

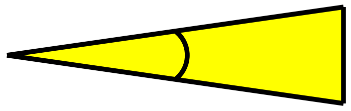
- To study mesh generation algorithms that are both provably good (with theoretical guarantees) and useful in practice.
  - I omit the huge literature on heuristic meshing algorithms.
  - I omit all quadrilateral/hexahedral meshing.


Marshall Bern and David Eppstein, ‘‘Quadrilateral Meshing by Circle Packing,’’ Sixth International Meshing Roundtable, pages 7−19, October 1997.

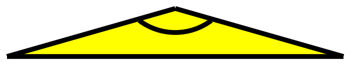

F. Betul Atalay, Suneeta Ramaswami, and Dianna Xu, ‘‘Quadrilateral Meshes with Bounded Minimum Angle,’’ Seventeenth IMR, pages 73−91, 2008.

How Meshes Affect Solution

Skinny elements cause problems.



Small angles cause poor conditioning.

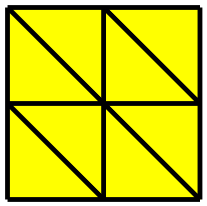


Large angles cause discretization error & big errors in interpolated derivatives.

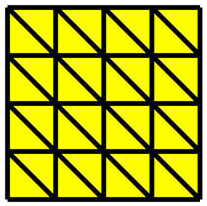
For tetrahedra, this applies to the dihedral angles.

(Not the plane angles!)

The number of elements matters.

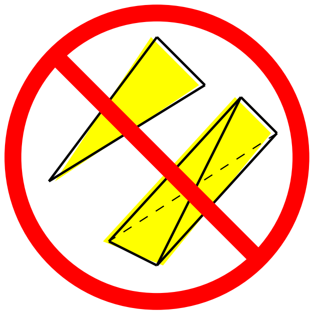


Fewer elements ➡ faster solution.

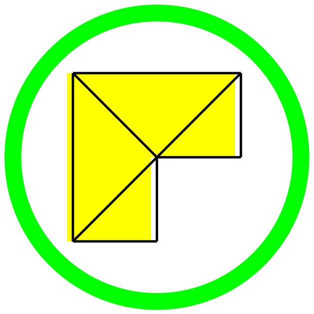


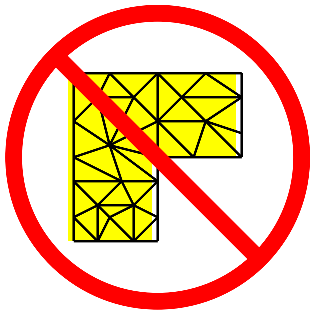
More elements ➡ more accurate solution.

Properties of a Good Mesh Generator



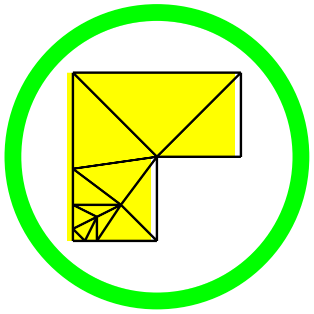
No poorly−shaped elements (triangles or tetrahedra).





Ability to generate a small mesh − one with relatively few elements.

(Refinement is easy; coarsening is hard.)

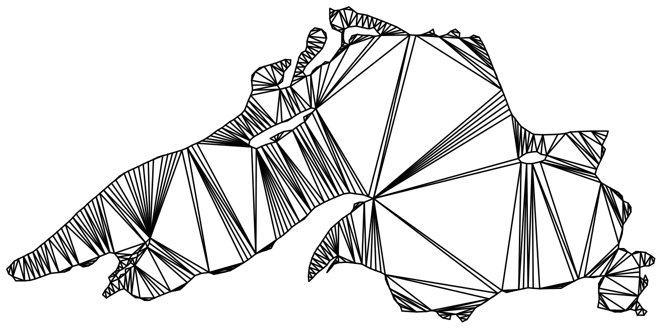


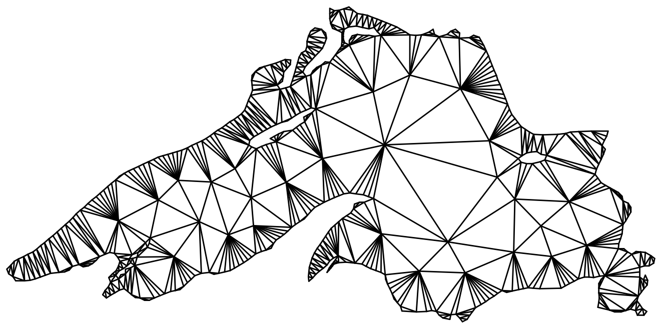
Ability to generate more elements in regions where higher accuracy is needed, and to exhibit good grading from small to large elements.

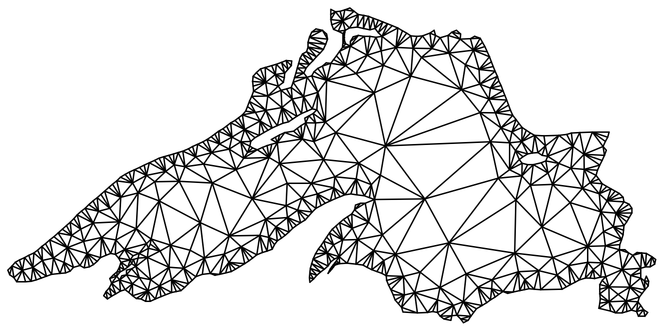
Well−Shaped Elements vs. Few Elements

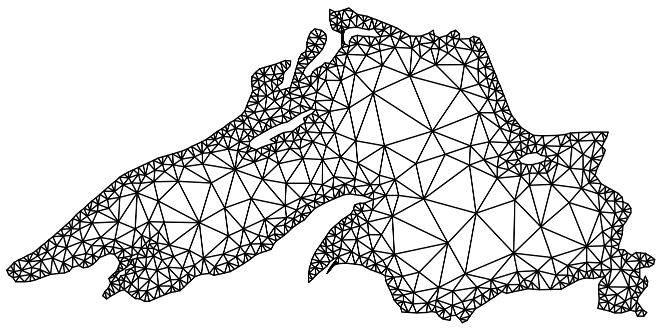
somewhat contradictory goals

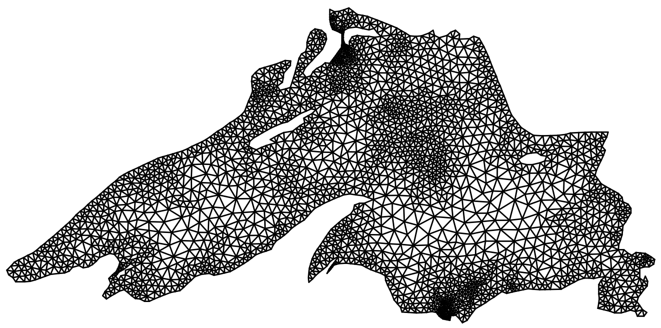
Lake Superior


No minimum angle
518 elements


5° minimum angle
593 elements


15° minimum angle
917 elements


25° minimum angle
1,427 elements


34.2° minimum angle
4,886 elements

These meshes generated by Ruppert's Delaunay refinement algorithm.

Great Moments in Theoretical Meshing

1987: William H. Frey introduces circumcenter insertion. ‘‘Selective Refinement: A New Strategy for Automatic Node Placement in Graded Triangular Meshes,’’ International Journal for Numerical Methods in Engineering **24** (11):2183−2200, November 1987.

1988: Brenda S. Baker, Eric Grosse, and C. S. Rafferty introduce first provably good mesh generation algorithm. ‘‘Nonobtuse Triangulation of Polygons,’’ Discrete & Computational Geometry **3**(2):147−168, 1988.

1989: L. Paul Chew introduces first provably good Delaunay refinement algorithm. ‘‘Guaranteed−Quality Triangular Meshes,’’ Technical Report TR−89−983, Department of Computer Science, Cornell University, 1989.

1990: Marshall Bern, David Eppstein, and John R. Gilbert introduce first algorithm with provably good grading & size−optimality. In 31st Foundations of CS. Journal version cited page 2.

1993: Jim Ruppert introduces first Delaunay refinement algorithm with good grading & size−optimality. In Fourth Symposium on Discrete Algorithms. Journal version cited page 30.

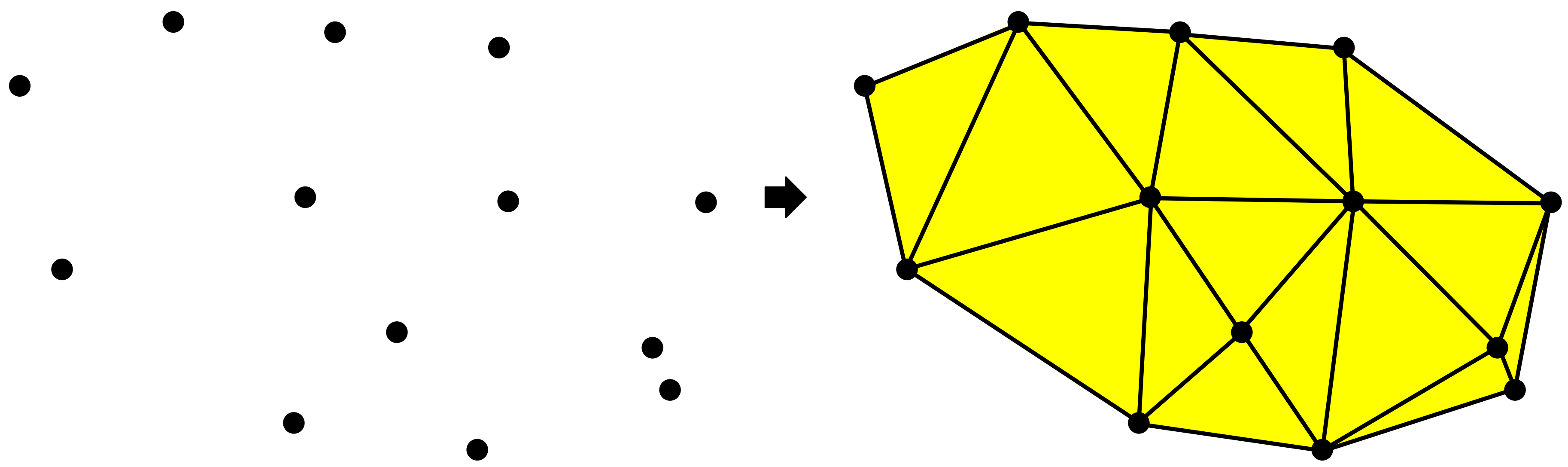
Outline

8 Delaunay Triangulation Review
19 Delaunay Refinement Mesh Generation
29 Ruppert's 2D Delaunay Refinement Algorithm
40 Analysis of Ruppert's Algorithm
46 Constrained Delaunay Triangulations
56 Curved Boundaries
63 2D Domains with Small Angles
73 3D Delaunay Refinement
90 Eliminating Slivers
98 3D Domains with Small Angles
102 Conclusions: Other Stuff & Open Problems

Delaunay Review

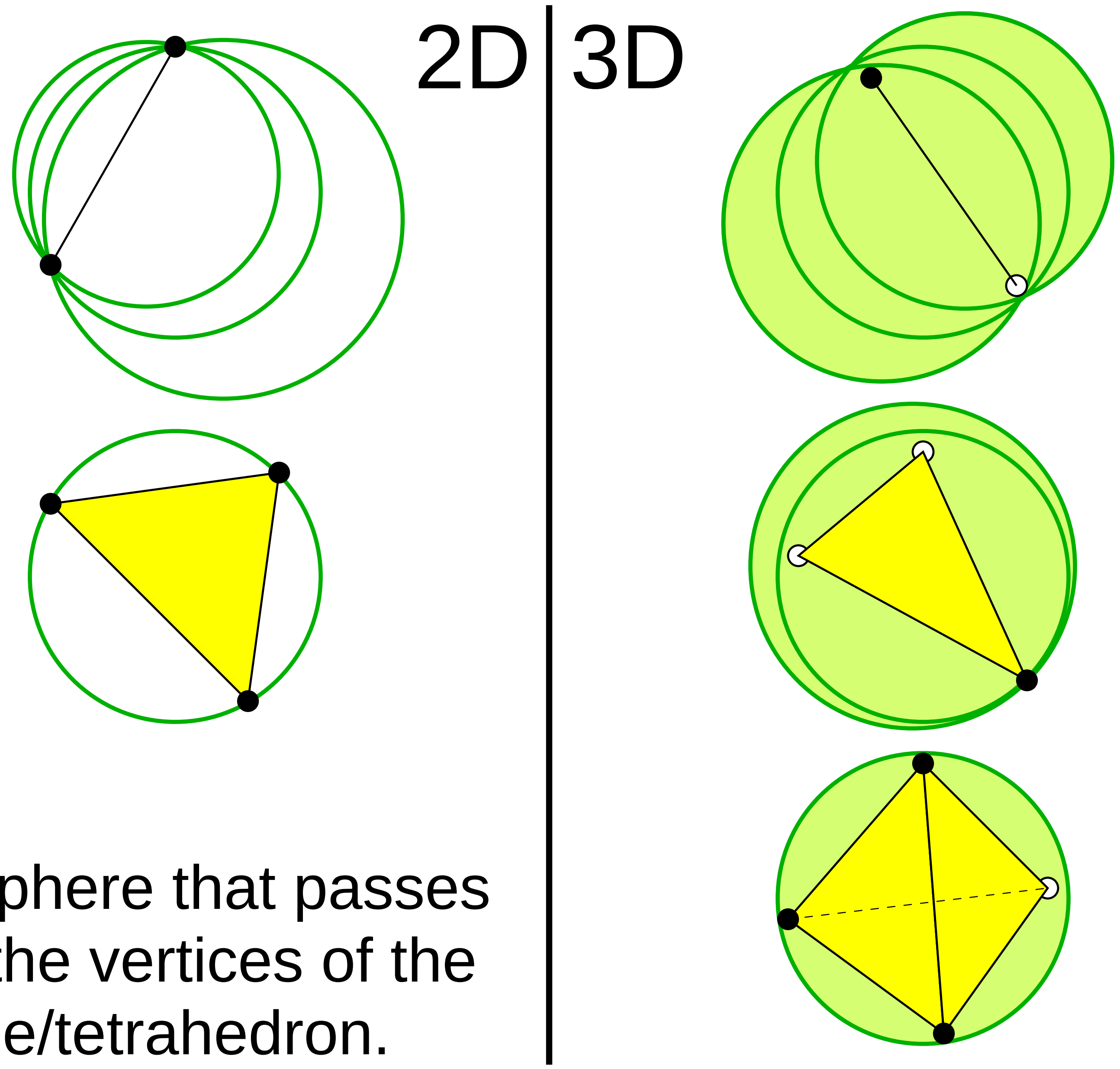
The Delaunay Triangulation

Every point set has a Delaunay triangulation. Think of it as a function that takes a set of points and outputs a triangulation.



(Some point sets have more than one Delaunay triangulation. Just pick one.)

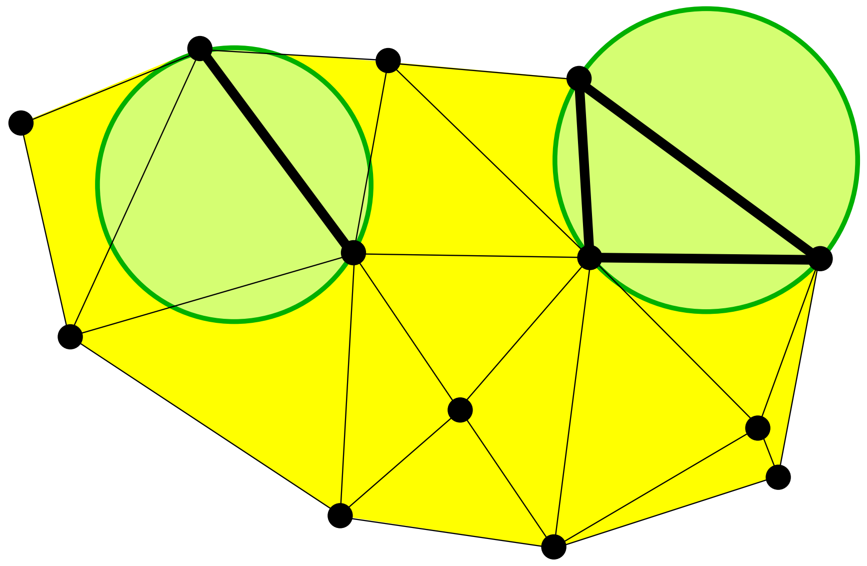
Circumcircle = Circumscribing Circle
Circumsphere = Circumscribing Sphere



Any circle/sphere that passes through all the vertices of the edge/triangle/tetrahedron.

The Delaunay Triangulation

...is a triangulation whose edges and triangles are all *Delaunay.*

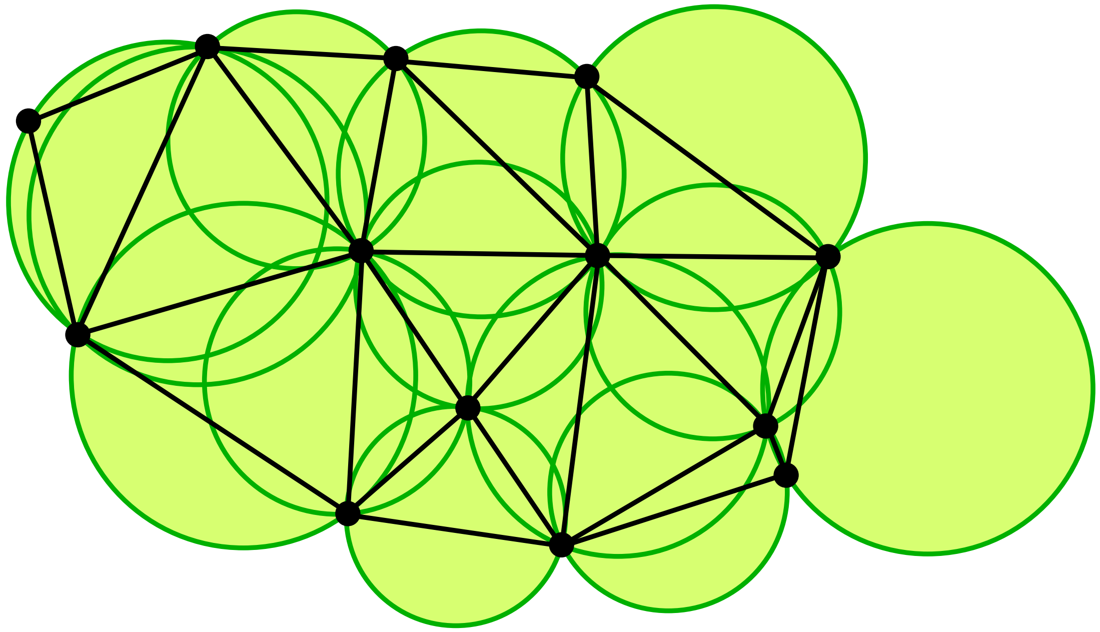


An edge/triangle is Delaunay if it has an *empty* circumcircle − one that encloses no vertex.

(There can be any number of vertices *on* the circumcircle.)

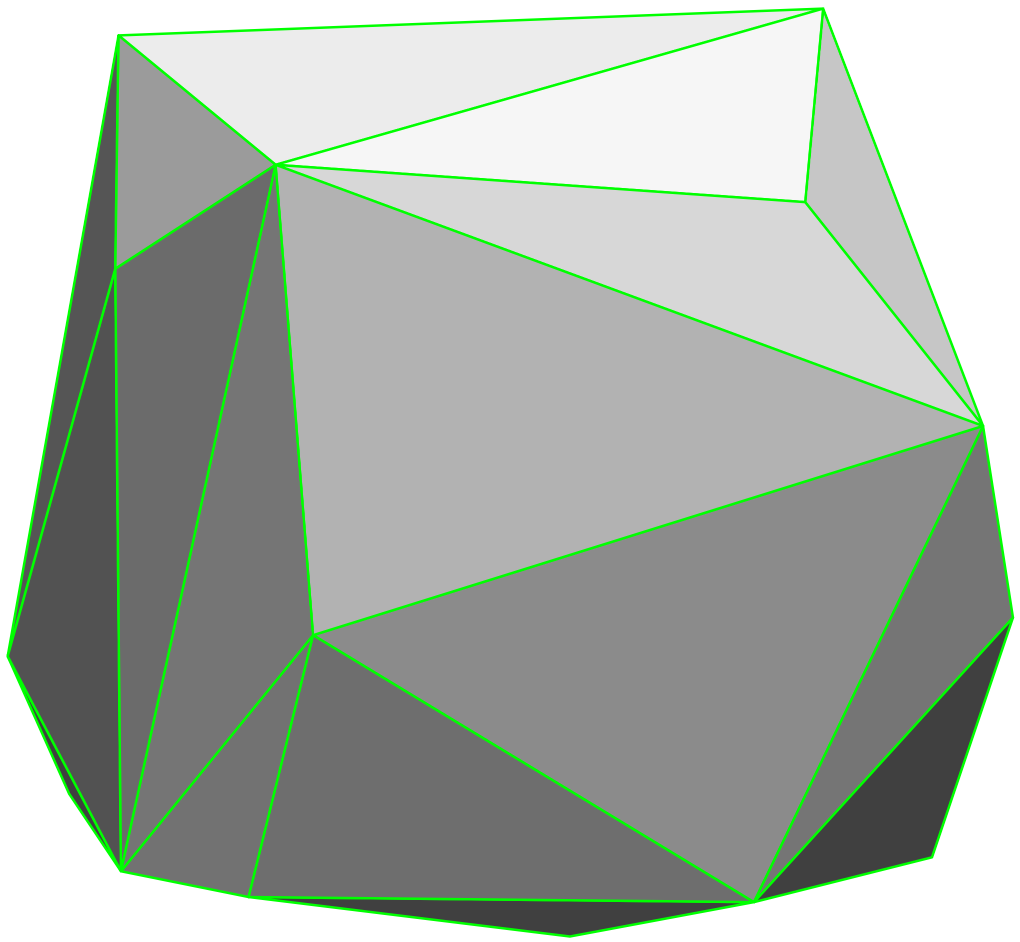
The Delaunay Triangulation

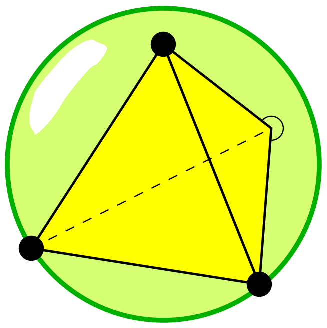
The circumcircle of every Delaunay triangle is empty.



The Delaunay Triangulation

...generalizes to higher dimensions. In 3D, every edge, triangular face, and tetrahedron of the DT has an empty circumsphere.





Boris Nikolaevich Delaunay, ‘‘Sur la Sphère Vide,’’ Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennykat Nauk **7**:793−800, 1934.
Delaunay was a Russian mathematician whose name transliterates to ‘‘Delaunay’’ in French, and ‘‘Delone’’ in English. His biographies are mostly found under ‘‘Boris Nikolaevich Delone.’’ His name allegedly came from an Irish ancestor named ‘‘Deloney.’’

Great Moments in DT Construction Algorithms

1970: C. O. Frederick, Y. C. Wong, and F. W. Edge unknowingly publish first 2D DT algorithm.
''Two−Dimensional Automatic Mesh Generation for Structural Analysis,'' International Journal for Numerical Methods in Engineering **2**:133−144, 1970.

1975: Michael Ian Shamos and Dan Hoey publish first O(*n* log *n*) 2D DT algorithm. ''Closest−Point Problems,'' 16th Symposium on Foundations of Computer Science, pp. 151−162, October 1975.

1977: Charles L. Lawson introduces 2D flip & incre−mental insertion algorithms; proves that 2D DT maximizes the minimum angle. ''Software for C^1 Surface Interpolation,'' Mathematical Software III (John R. Rice, editor), pages 161−194, Academic Press, New York, 1977.

1981: Adrian Bowyer and David F. Watson publish *n*−dimensional incremental insertion alg.
Bowyer: ''Computing Dirichlet Tessellations,'' Computer Journal **24**(2):162−166, 1981.
Watson: ''Computing the *n*−dimensional Delaunay Tessellation with Application to Voronoi Polytopes,'' Computer Journal **24**(2):167−172, 1981. (Independent papers, published side by side!)

1989: Kenneth L. Clarkson and Peter W. Shor give optimal *n*−dimensional insertion alg. ''Applications of Random Sampling in Computational Geometry, II,'' Discrete & Computational Geometry **4**(1):387−421, 1989.

Good Choices for Implementation

2D: **Leonidas J. Guibas and Jorge Stolfi's elaboration of Shamos−Hoey divide−and−conquer.**

''Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams,'' ACM Transactions on Graphics **4**(2):74−123, April 1985. Includes data structures & detailed pseudocode.

3D: **Use Bowyer−Watson incremental insertion alg.**

Computer Journal **24**(2):162−172, 1981. (See full citations on previous page.)

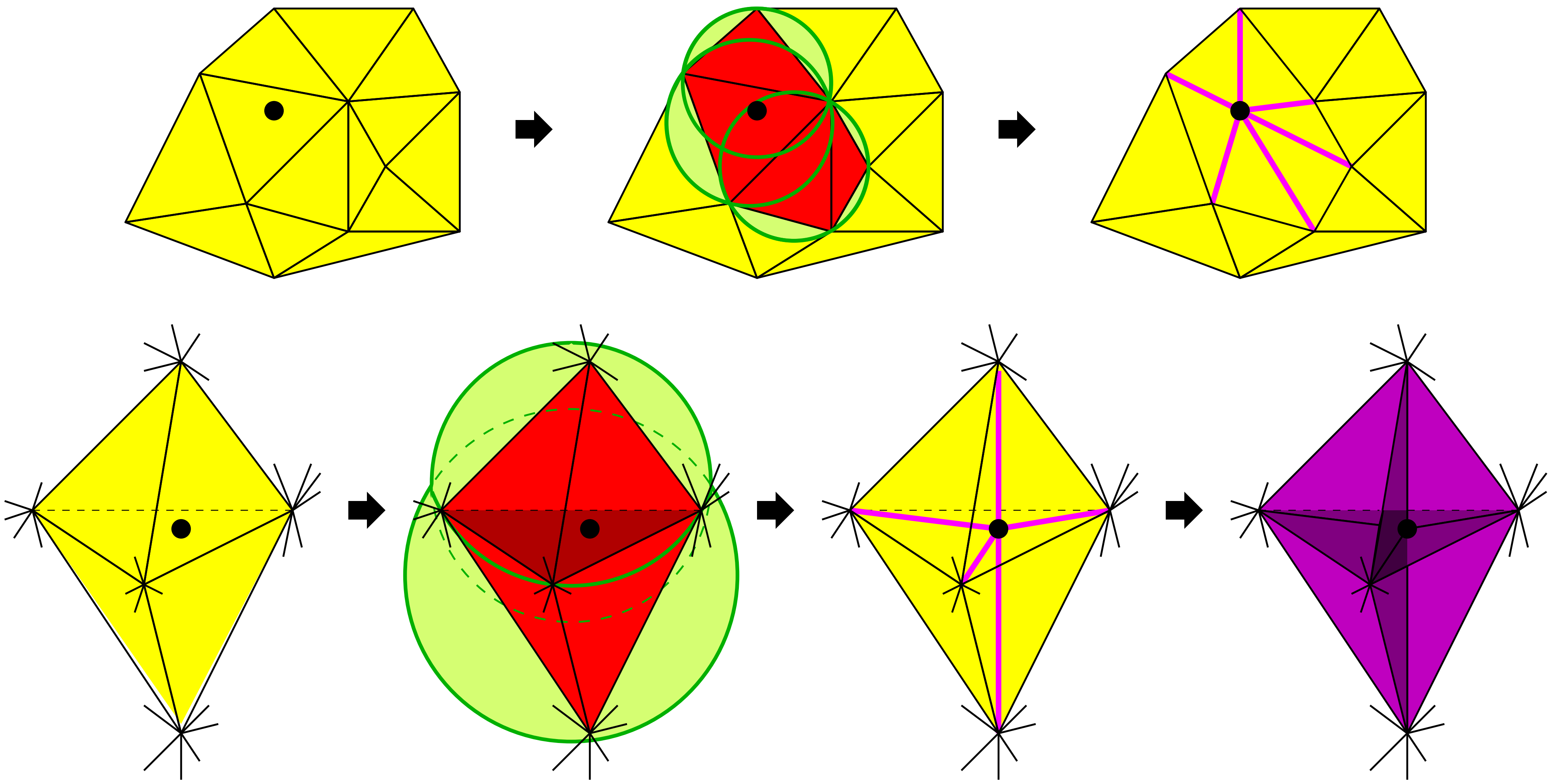
But, use insertion ordering & point location of Nina Amenta, Sunghee Choi, and Günter Rote.

''Incremental Constructions con BRIO,'' Proceedings of the Nineteenth Annual Symposium on Computational Geometry, pages 211−219, June 2003.
Shows how to order the vertex insertions in a way that is friendly to the memory hierarchy, gives very fast point location in practice, and is theoretically optimal like Clarkson−Shor.

And, use the mesh dictionary data structure of Daniel K. Blandford, Guy E. Blelloch, David E. Cardoze, and Clemens Kadow.

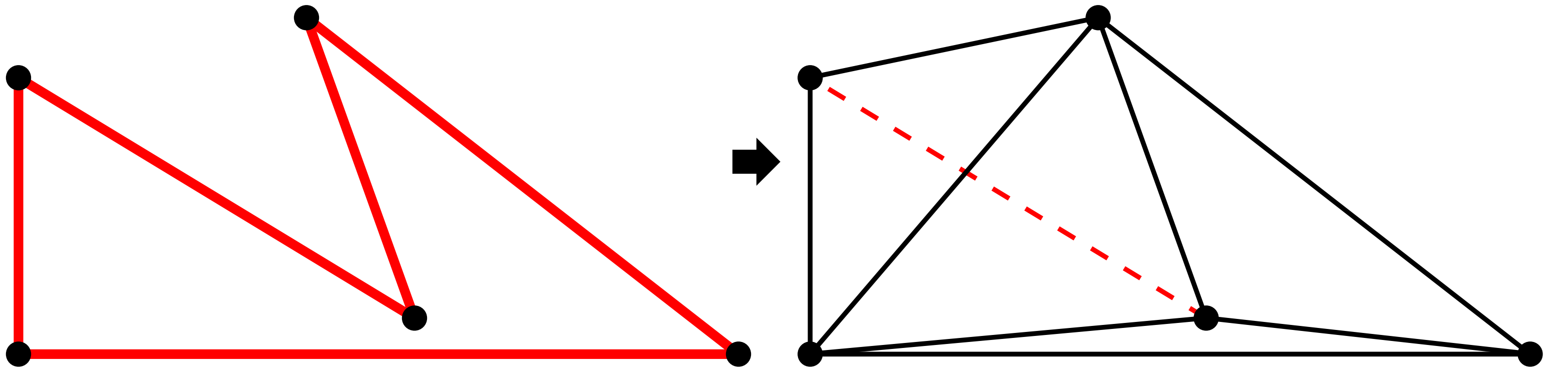
''Compact Representations of Simplicial Meshes in Two and Three Dimensions,'' Twelfth International Meshing Roundtable, pages 135−146, September 2003.
Because the data structure has no pointers between tetrahedra, programming on top of it is much easier and much less bug−prone, even if you don't implement the compaction that is the subject of the paper.

The Bowyer–Watson Incremental Insertion Algorithm for Delaunay Triangulation



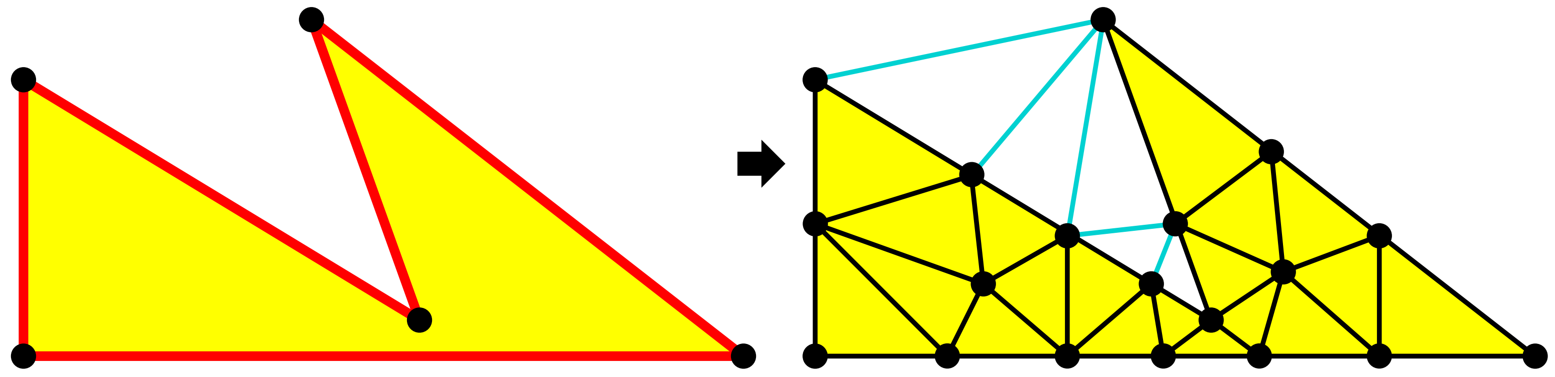
Insert one vertex at a time.
Remove all triangles/tetrahedra that are no longer Delaunay.
Retriangulate the cavity with a fan around the new vertex.

Why the Delaunay Triangulation Alone Doesn't Solve the Problems of Meshing



- The Delaunay triangulation maximizes the minimum angle, but the minimum angle may still be too small.
- The Delaunay triangulation might not conform to the domain boundaries.

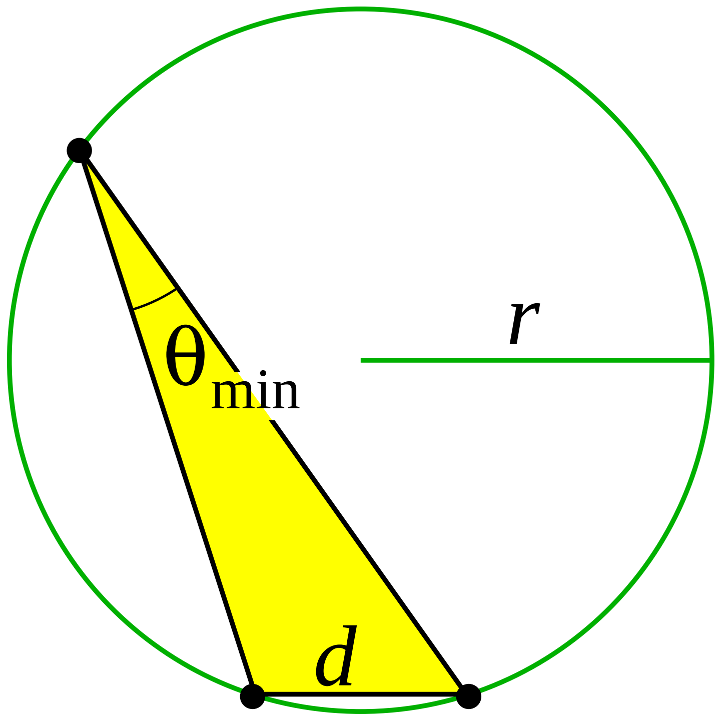
Solution: Add more vertices

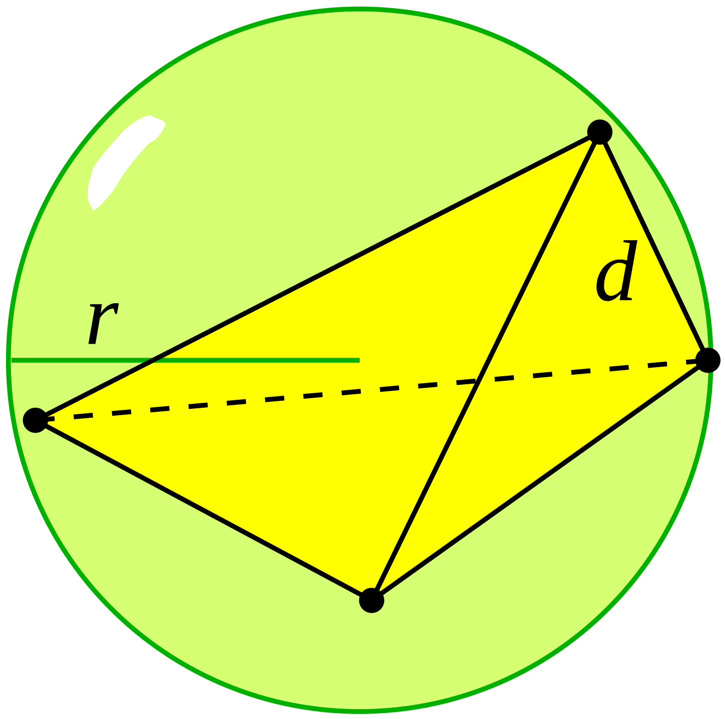


The Big Question: Where?

Delaunay Refinement

A Quality Measure for Simplices



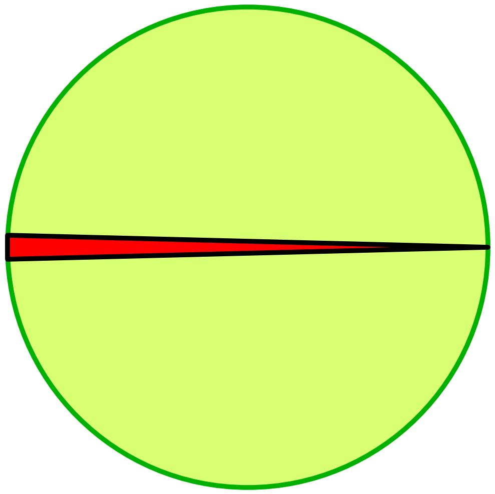


Circumradius−to−shortest edge ratio is r/d.

In two dimensions, $\frac{r}{d} = \frac{1}{2 \sin \theta_{\min}}$. Small ratio ⬌ big $\theta_{\min}$.

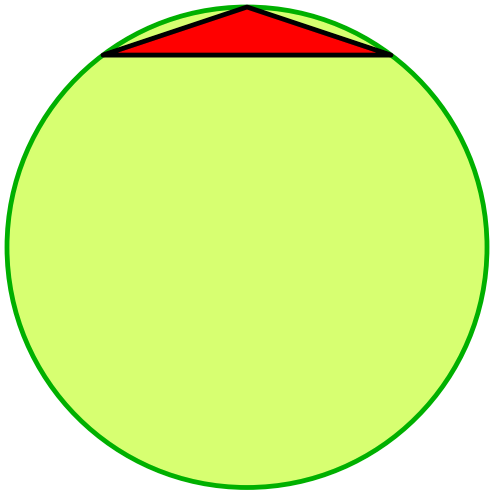
In 2D or 3D, the smaller the ratio, the better.
In 3D, it’s not a perfect quality measure, but it’s the best we can prove things about.

Skinny Triangles



Needles

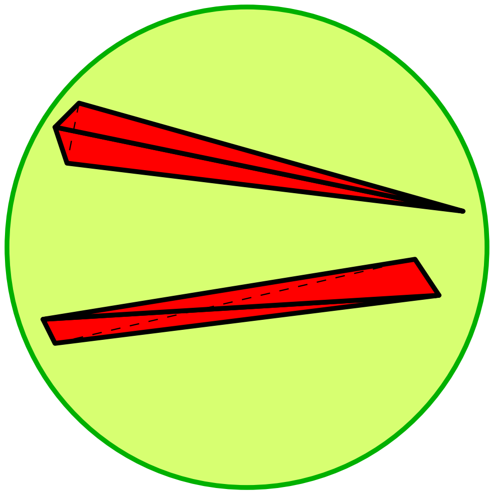
(disparate edge lengths)



Caps

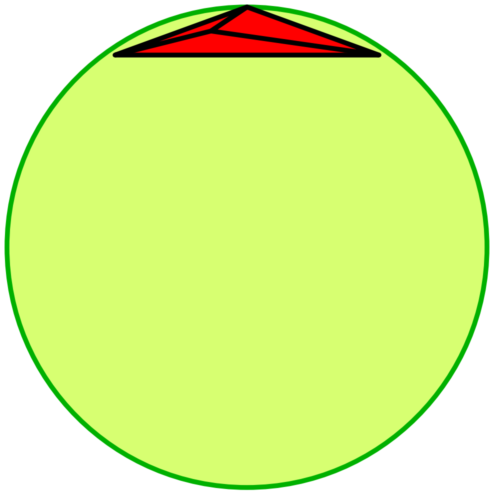
(angle near 180°)

Skinny Tetrahedra



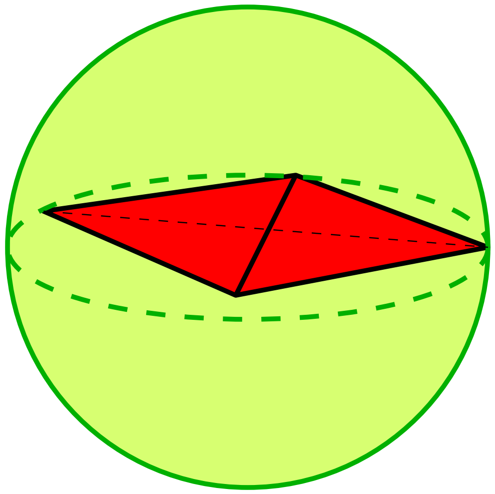
Needles

(disparate edge lengths)



Caps

(large solid angle)

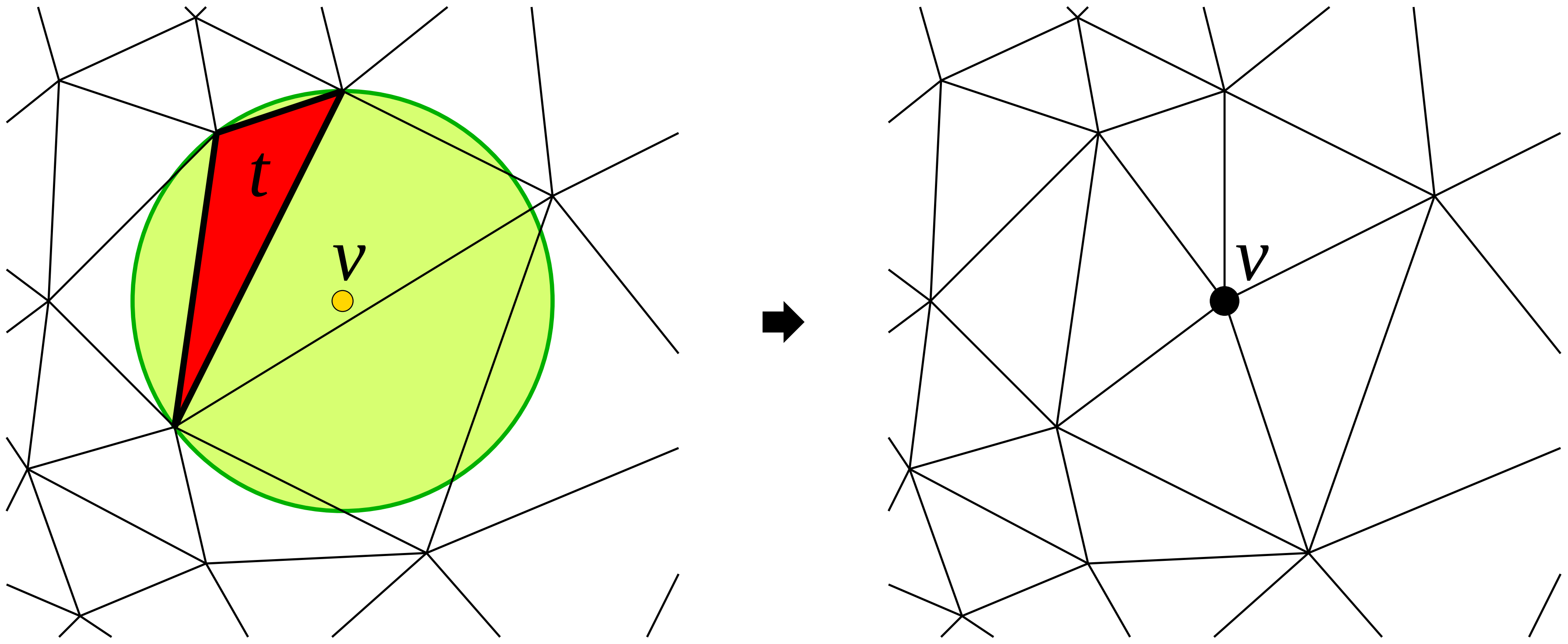


Slivers

(slivers are evil; they can have very small circumradius−to−shortest edge ratios, but awful dihedrals)

Delaunay Refinement

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

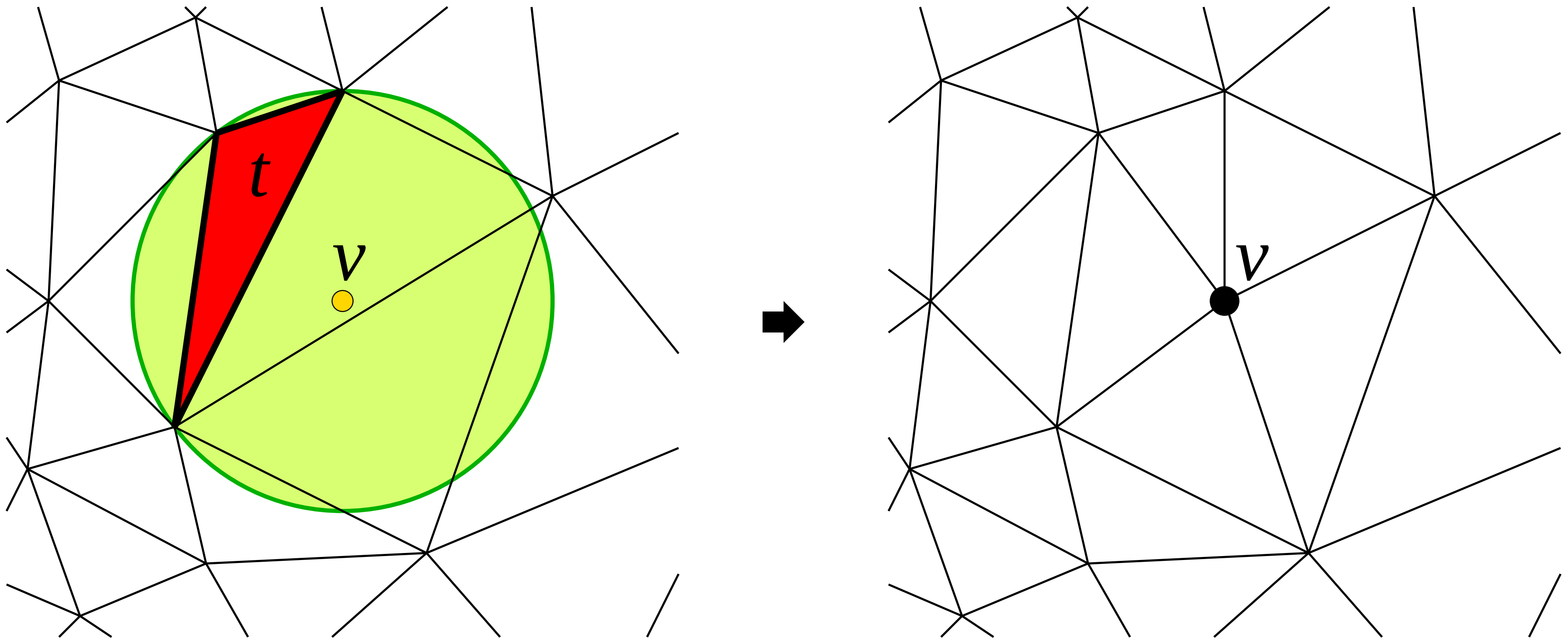


Kill each skinny triangle by inserting vertex at circumcenter.
(Bowyer−Watson algorithm.)

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

Paul Chew's Idea

L. Paul Chew, ''Guaranteed−Quality Triangular Meshes,'' Technical Report TR−89−983, Department of Computer Science, Cornell University, 1989.

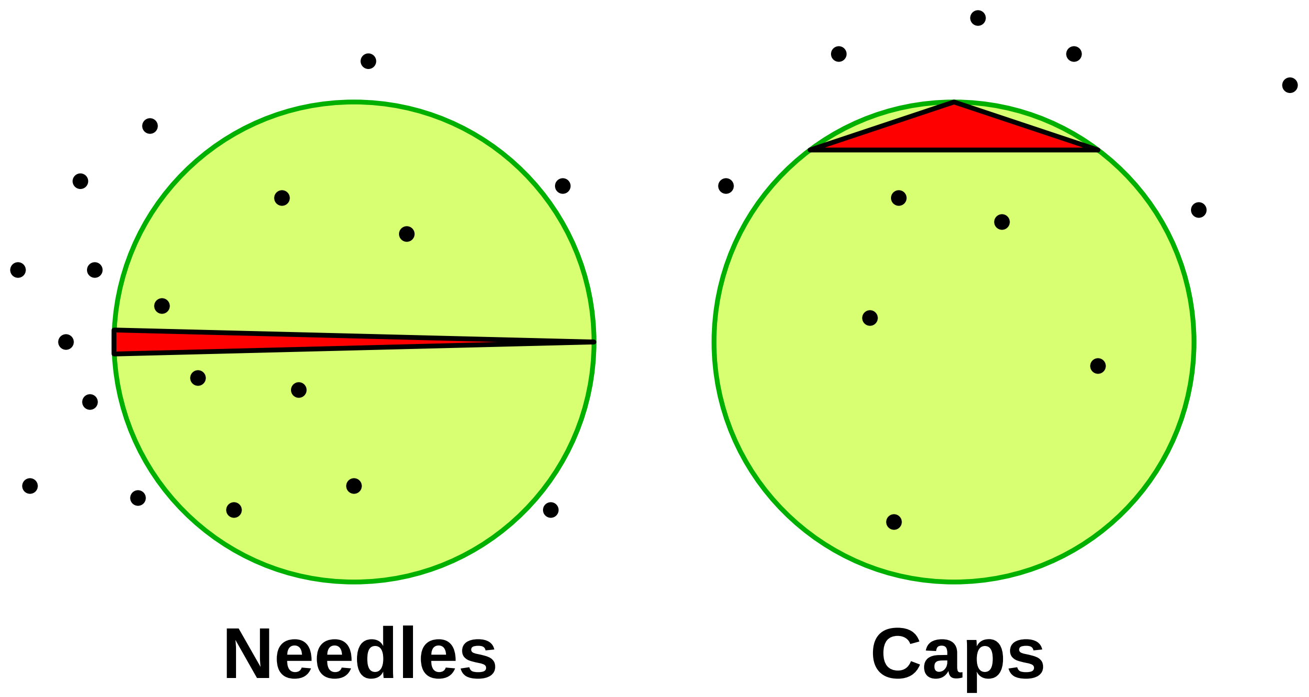


Call a triangle or tetrahedron ''skinny'' only if circumradius−to−shortest edge ratio > 1.

Then all new edges are longer than shortest edge of *t*.

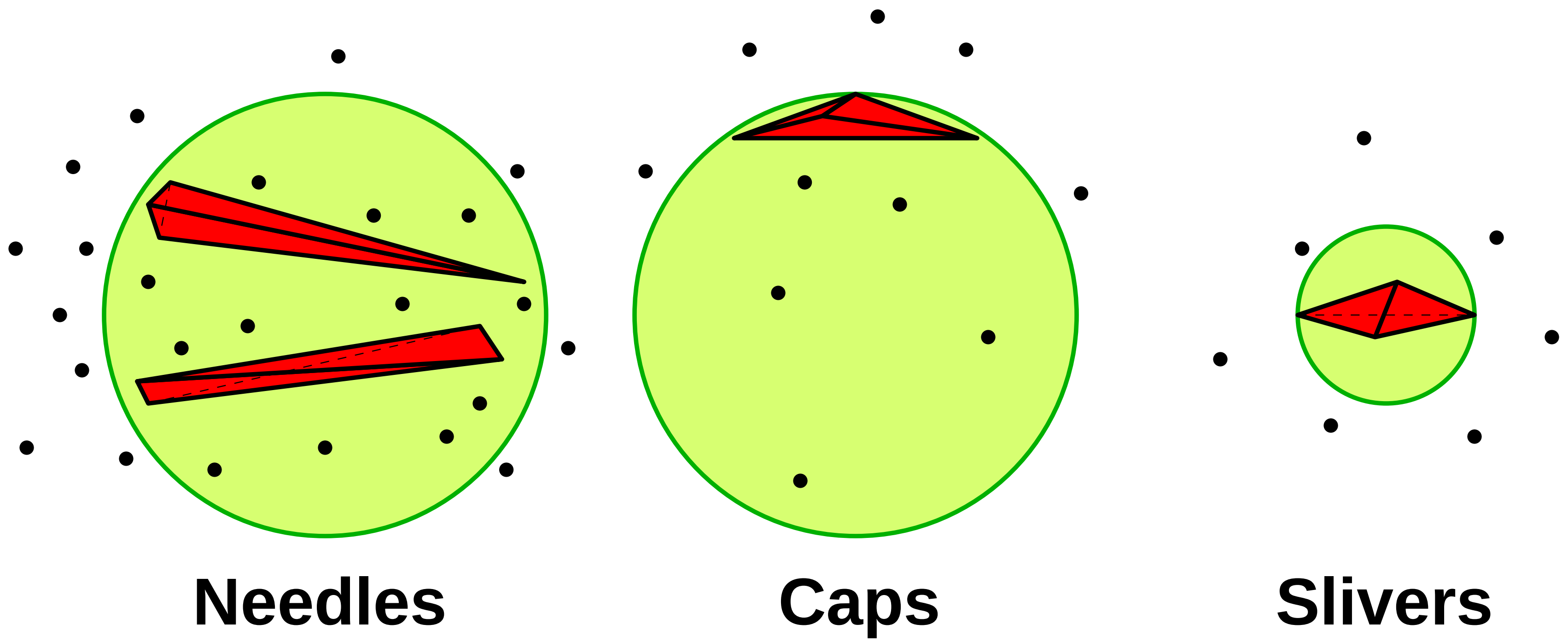
You never create an edge shorter than the shortest pre−existing edge. Therefore, the algorithm must terminate!

Skinny Triangles



Delaunay refinement scatters vertices with spacing proportional to the shortest nearby edge. A triangle whose circumradius is much bigger than its shortest edge cannot survive.

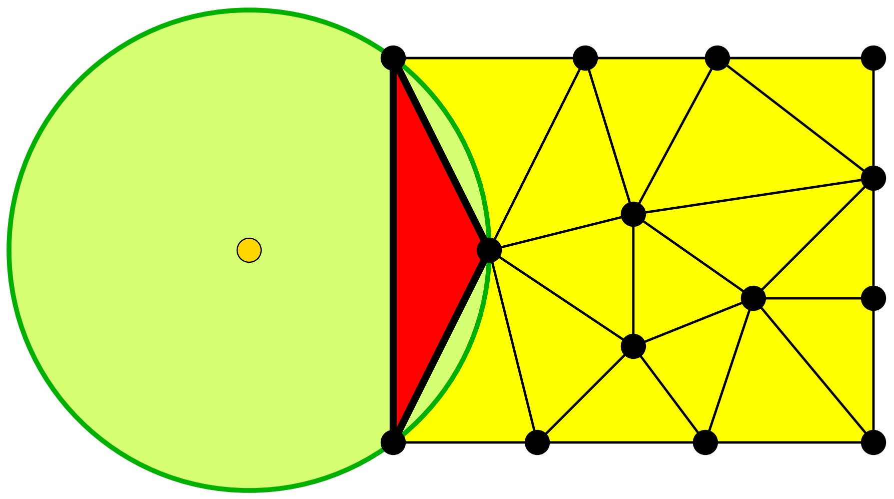
Skinny Tetrahedra



Same goes for tetrahedra with big circumspheres.

Alas, slivers with small circumradius−to−shortest edge ratios can survive.

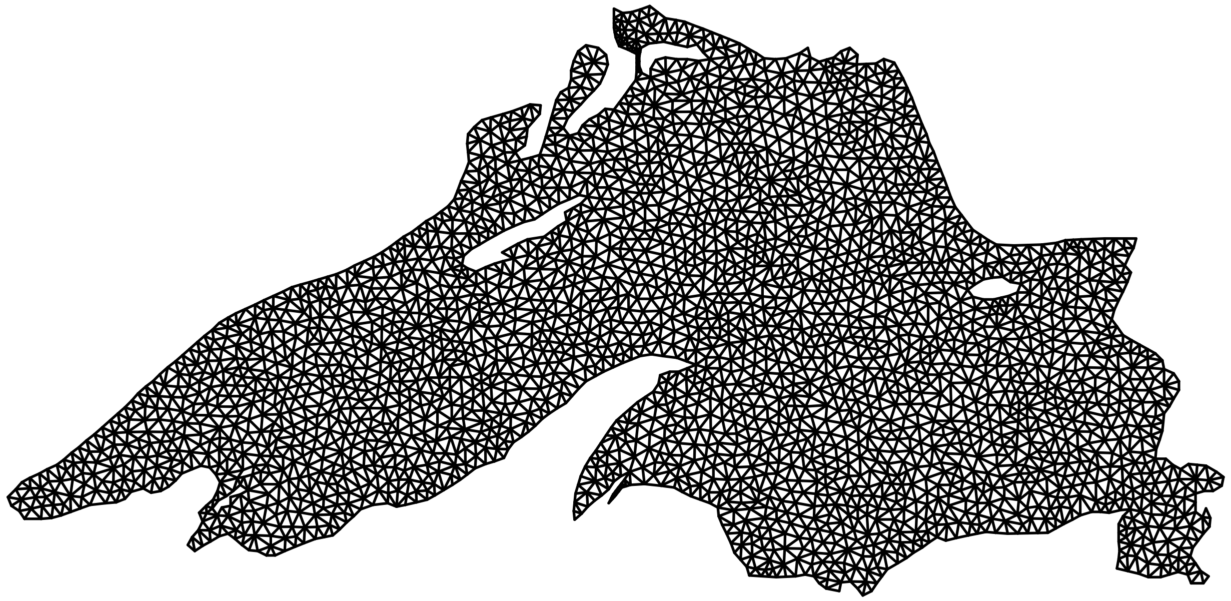
What if a circumcenter is outside the domain?

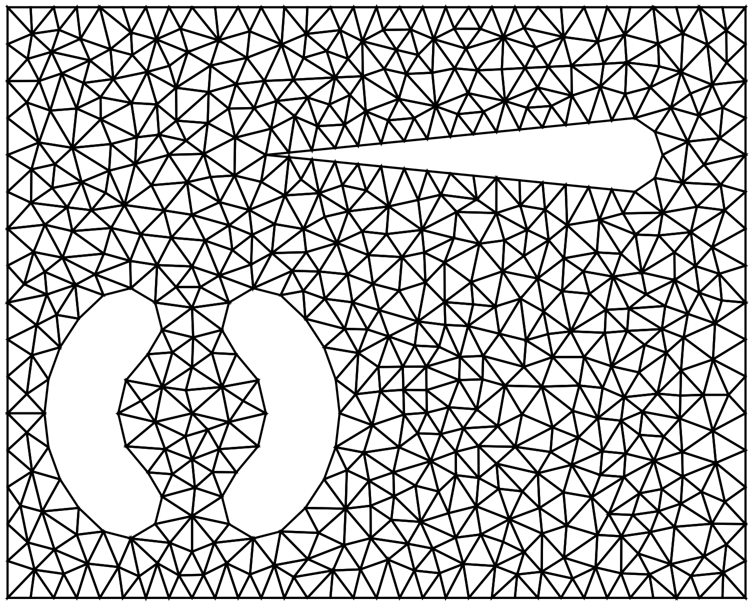


Domain boundaries are responsible for all the complications of Delaunay refinement algorithms, and the differences between them.

Chew’s First Delaunay Refinement Algorithm

L. Paul Chew, ‘‘Guaranteed−Quality Triangular Meshes,’’ Technical Report TR−89−983, Department of Computer Science, Cornell University, 1989.





Courtesy Paul Chew

Subdivides boundary segments into roughly equal edges before applying Delaunay refinement.

Uses constrained Delaunay triangulations.

Generates mesh with all angles between 30° and 120°.

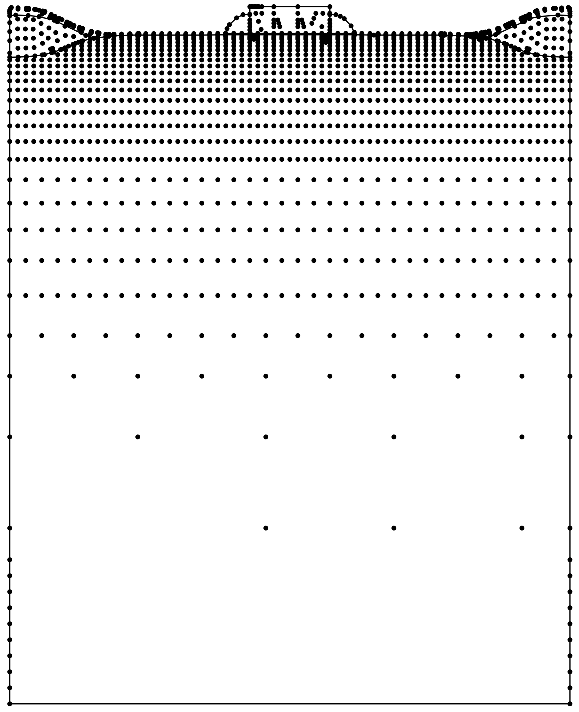
Cannot produce graded meshes.

Ruppert’s Algorithm

Jim Ruppert's Delaunay Refinement Algorithm

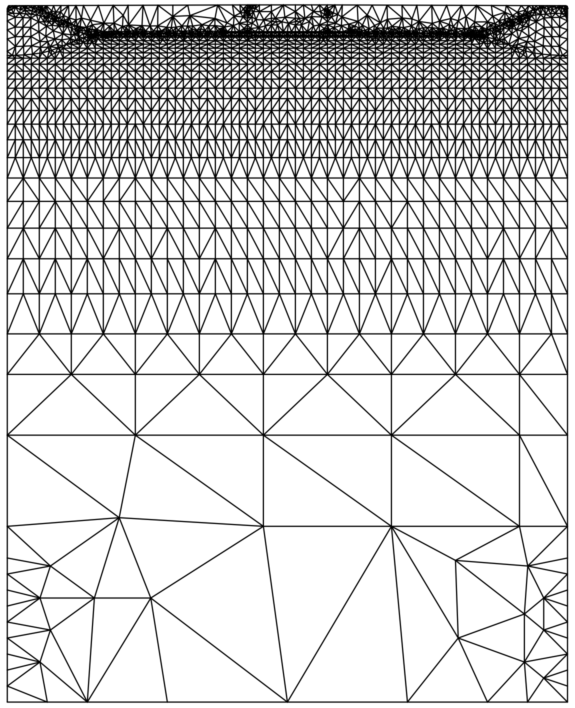
''A Delaunay Refinement Algorithm for Quality 2−Dimensional Mesh Generation,'' Journal of Algorithms **18** (3):548−585, May 1995.

- The input is a planar straight line graph (PSLG): a set of vertices and non−crossing segments.



PSLG

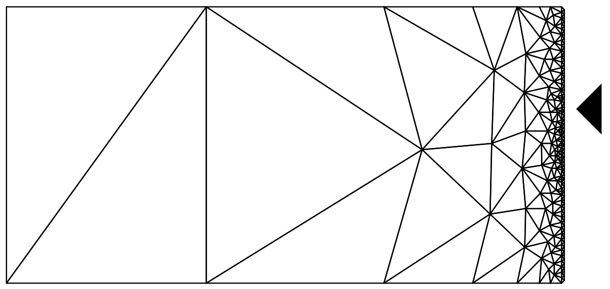
20° mesh



- You choose the minimum acceptable angle θ, up to 20.7°. (Up to ~33° in practice.) Implies 180°− 2 θ maximum. Any triangle with angle < θ is ''skinny.''

Jim Ruppert's Delaunay Refinement Algorithm

''A Delaunay Refinement Algorithm for Quality 2−Dimensional Mesh Generation,'' Journal of Algorithms **18** (3):548−585, May 1995.

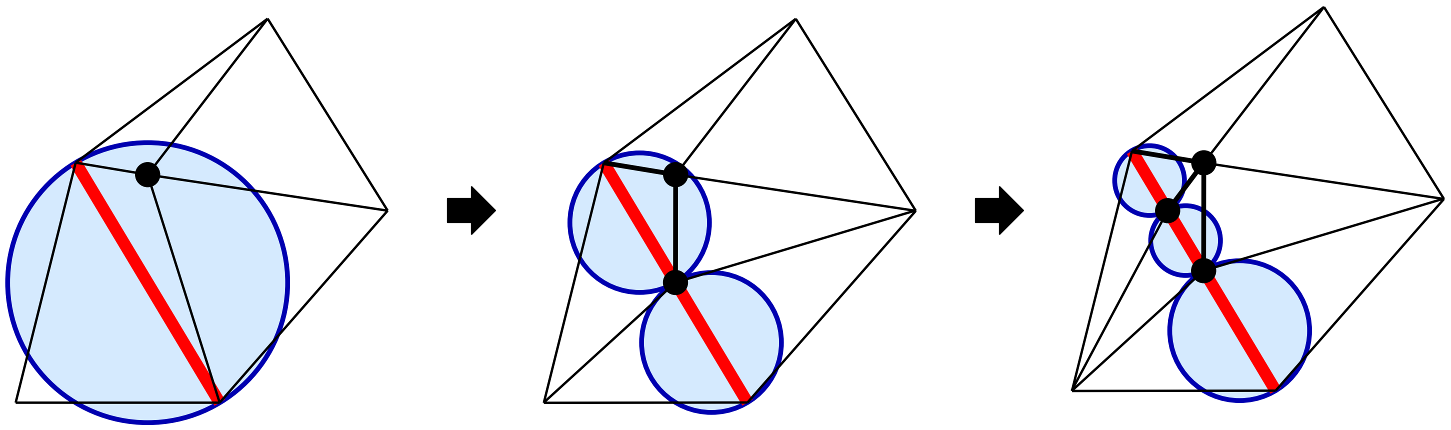

This side has a thin vertical layer to mesh.

- Provably good grading: all edge lengths are $\geq C$ times the "local feature size." C is reasonable (e.g. 1/9 for $\theta = 15°$ minimum angle). Theoretical grading guarantee deteriorates as $\theta \rightarrow 20.7°$.

- ''Size−optimal'': number of triangles is within a constant factor of the *smallest* possible mesh with minimum angle θ. (The constant is too large to give a meaningful guarantee in practice.)

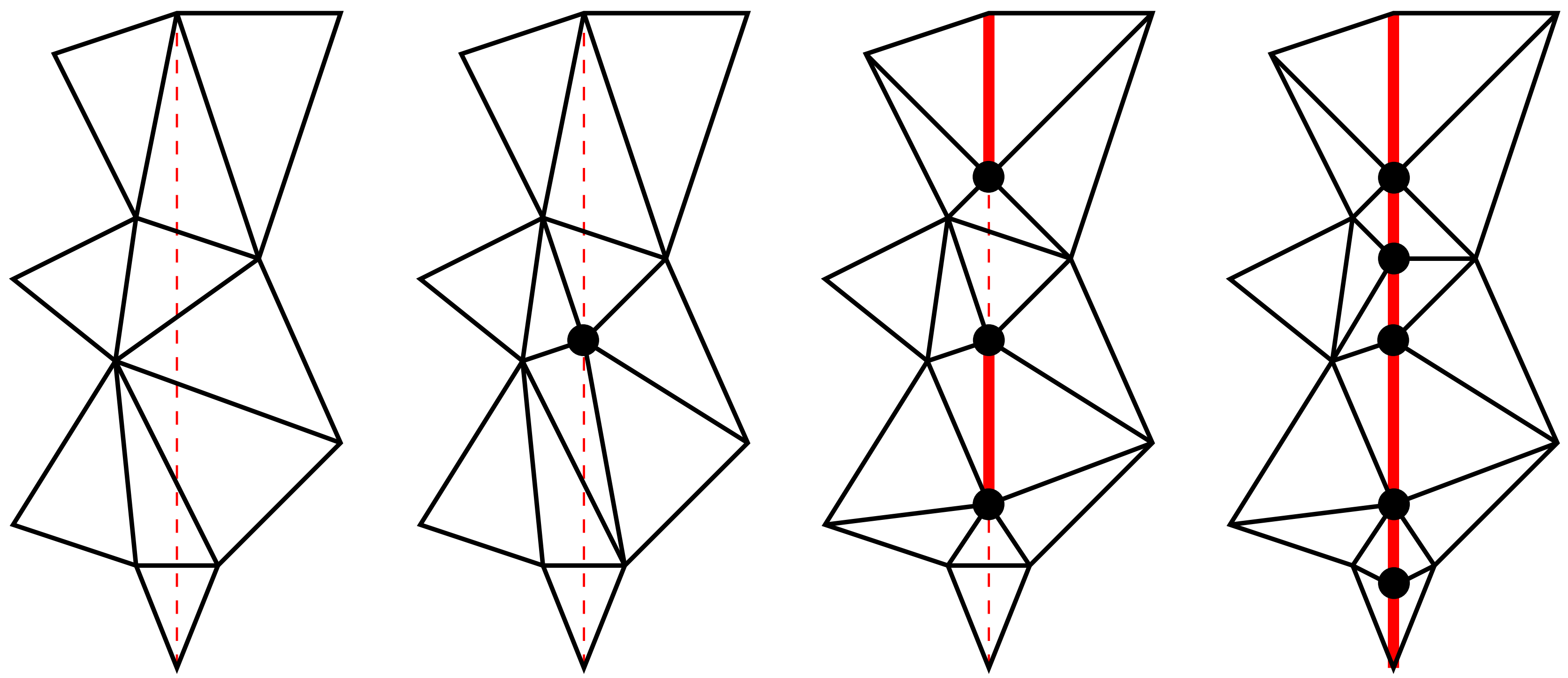
Vertex Insertion Rule 1

An input segment is said to be *encroached* if there is a vertex inside its *diametral circle.* (Its smallest circumcircle.)



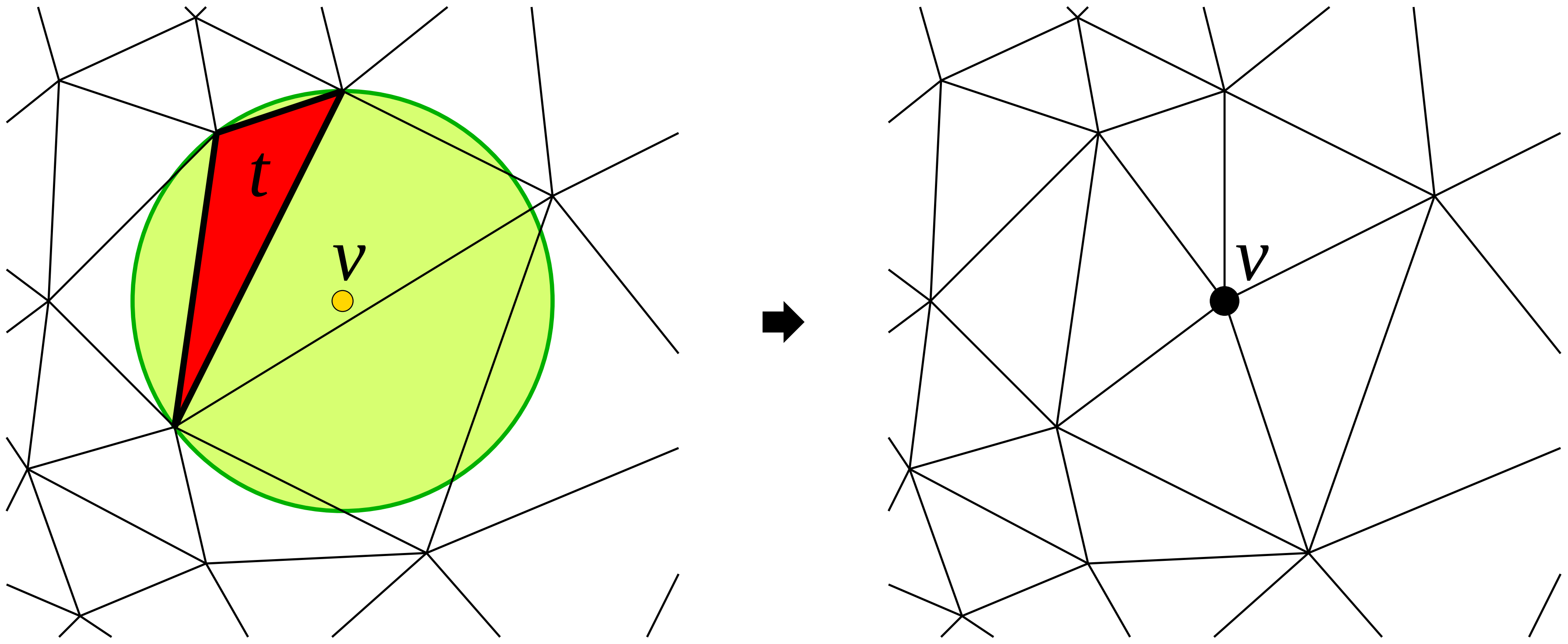
Any encroached segment is *split* into *subsegments* by inserting a new vertex at its midpoint.

Segment Recovery by “Stitching” (by Rule #1)



Missing segments and subsegments are encroached. Split them at their midpoints until no subsegment is missing.

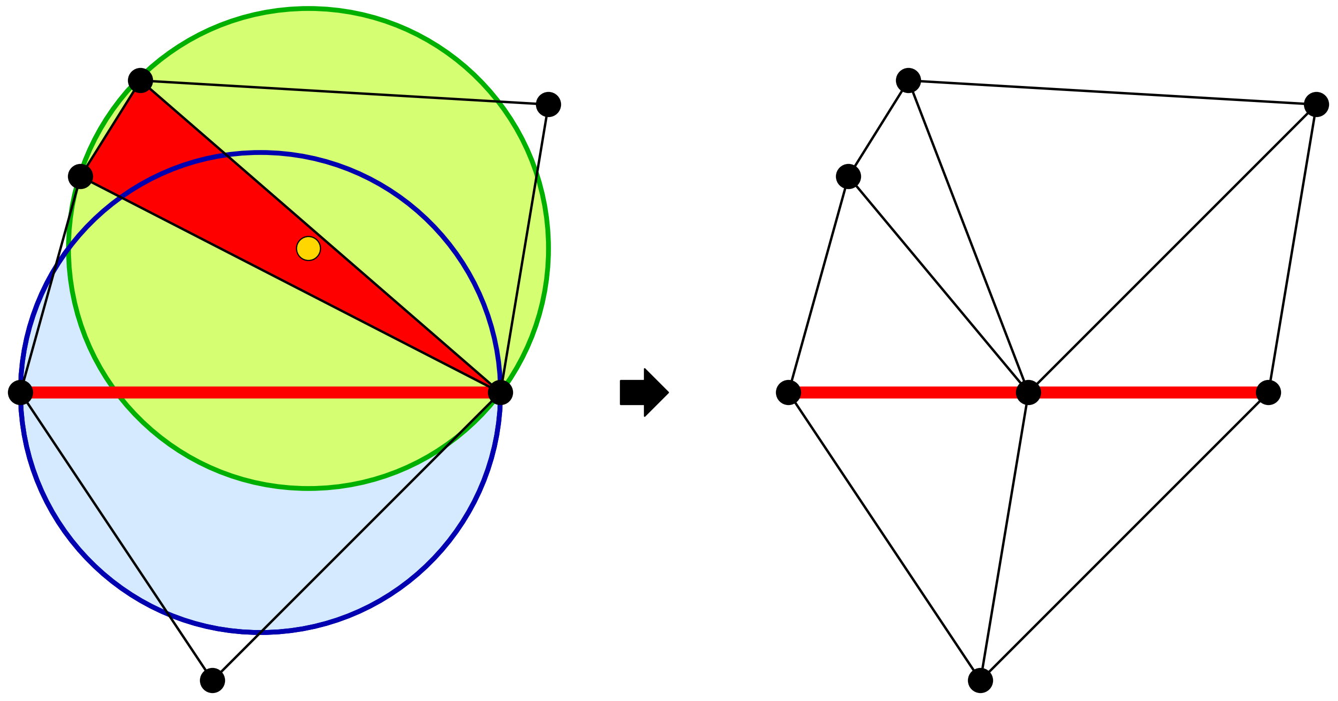
Vertex Insertion Rule 2



Insert vertices at circumcenters of triangles with small angles (e.g., < 20.7°).

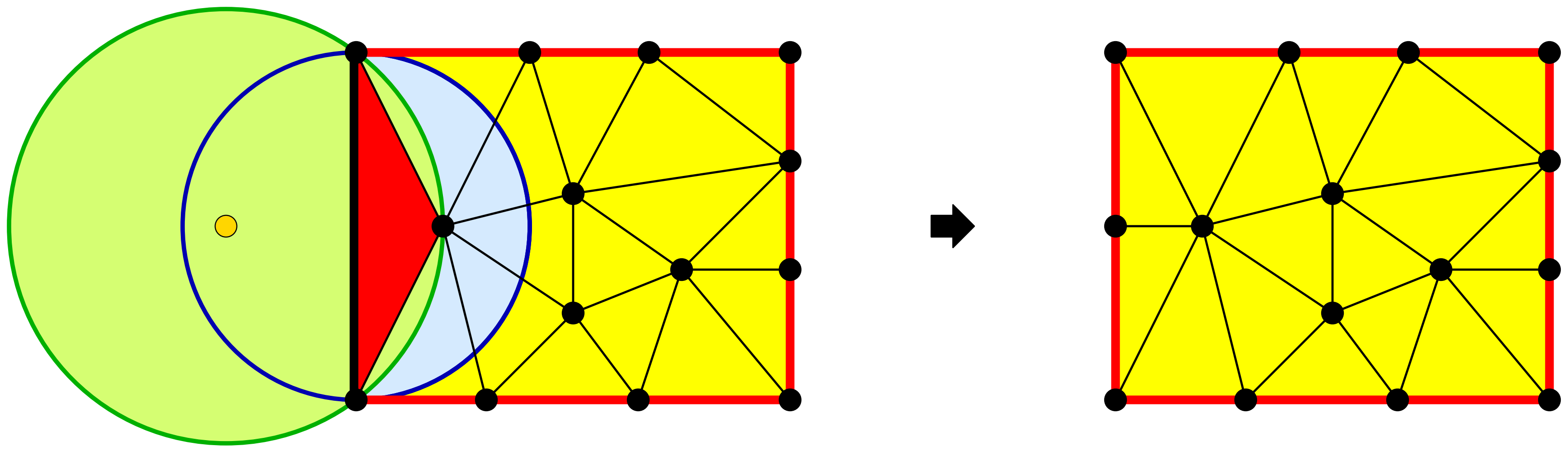
Triangles that are too large are treated likewise.

Encroached Subsegments Have Priority over Skinny Triangles



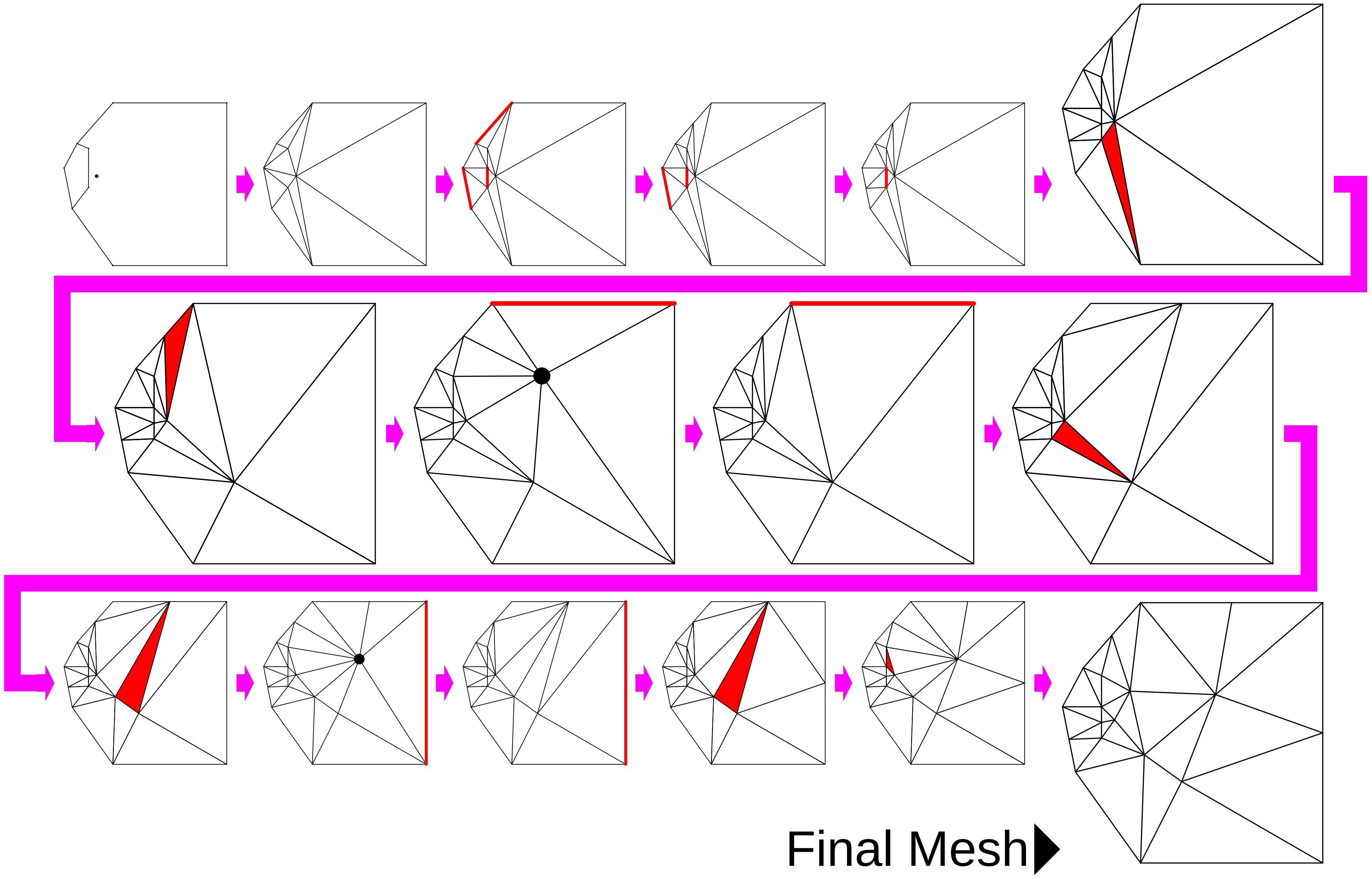
If the circumcenter of a skinny triangle encroaches upon a subsegment, reject the circumcenter.
Split the subsegment instead.

What if a circumcenter is outside the domain?



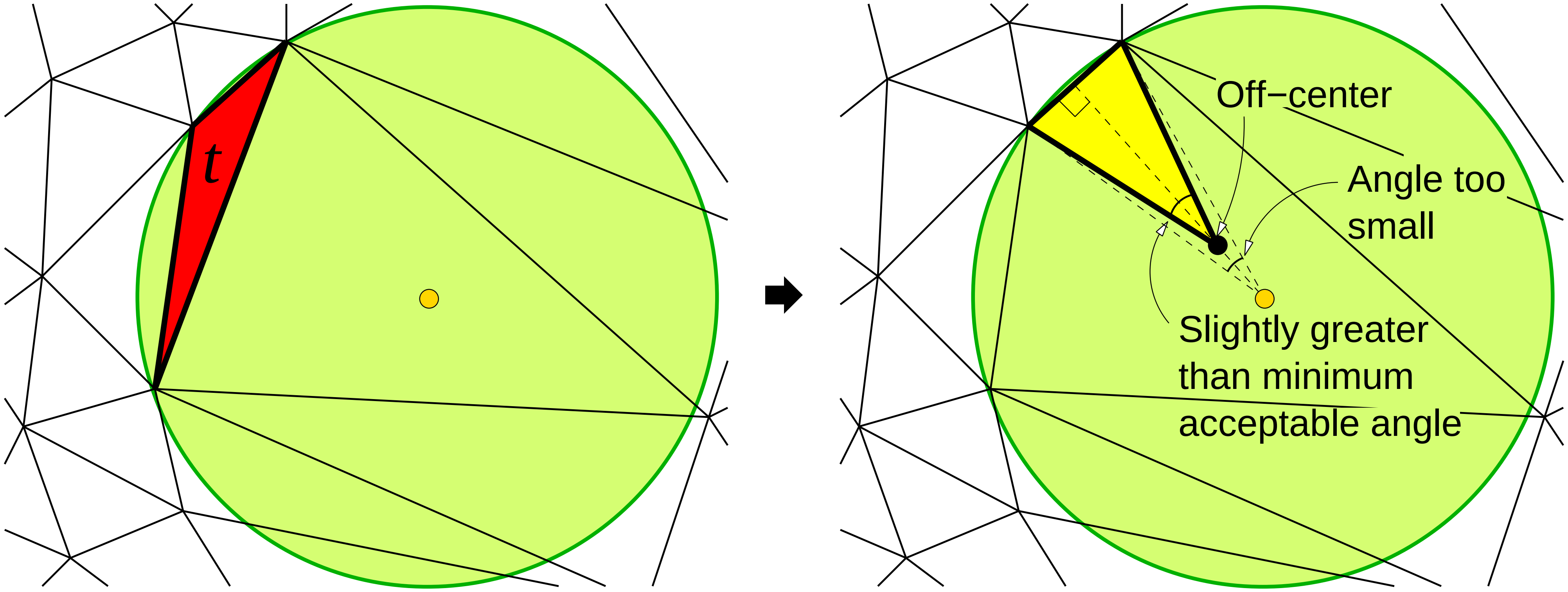
Then a boundary segment is encroached. Split it.

Ruppert's Algorithm in Action



Alper Üngör’s “Off−Centers”

‘‘Off−Centers: A New Type of Steiner Points for Computing Size−Optimal Quality−Guaranteed Delaunay Triangulations,’’ LATIN 2004: Theoretical Informatics, 6th Latin American Symposium, Lecture Notes in Computer Science volume 2976, Springer, April 2004.



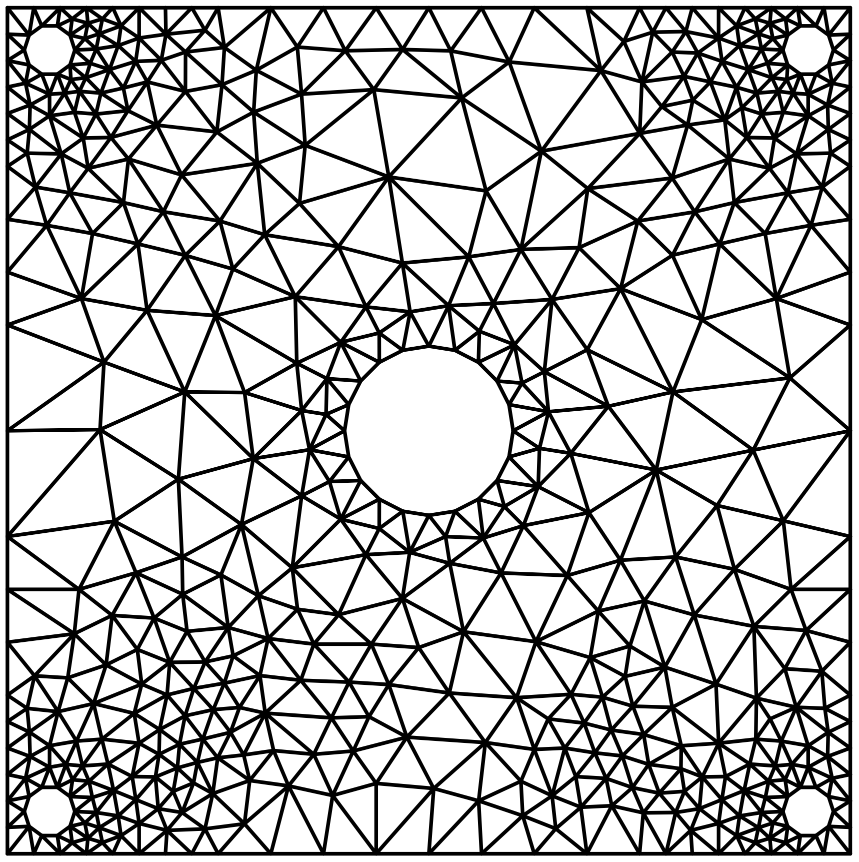
If circumcircle is so big that new triangle adjoining shortest edge of t will be skinny, place new vertex off−center so new triangle will be a few degrees better than minimum acceptable angle.

Warning: to get benefits, you must experiment with how far to move the off−center toward the short edge. Note: off−centers turn Delaunay refinement into an advancing front algorithm!

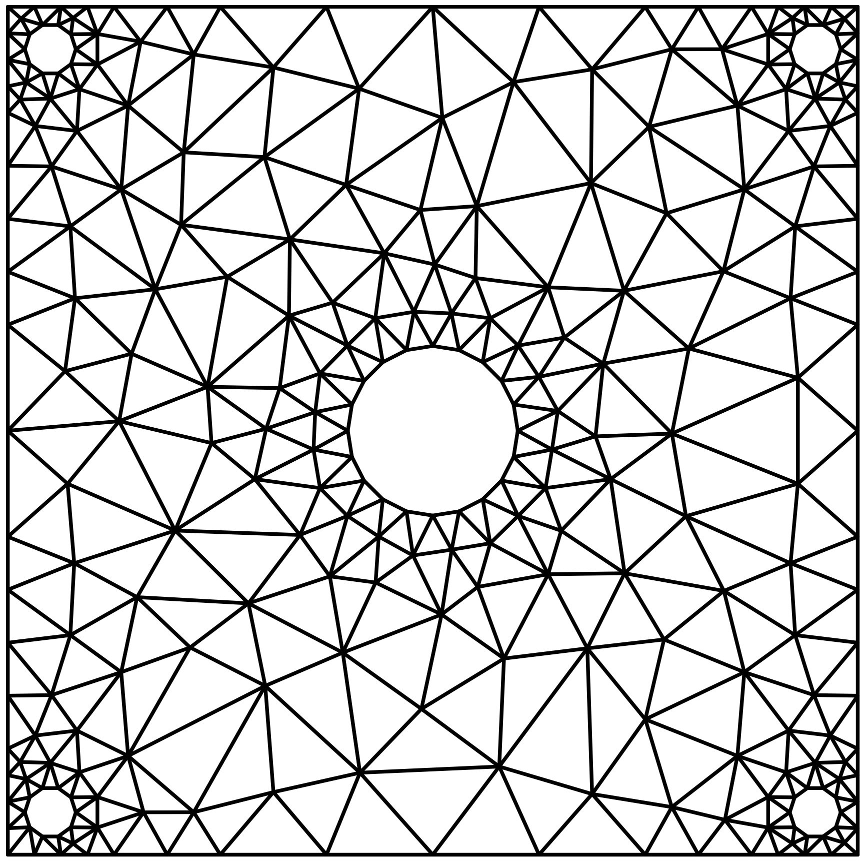
Alper Üngör's "Off−Centers"

"Off−Centers: A New Type of Steiner Points for Computing Size−Optimal Quality−Guaranteed Delaunay Triangulations," LATIN 2004: Theoretical Informatics, 6th Latin American Symposium, Lecture Notes in Computer Science volume 2976, Springer, April 2004.

Meshes with 33° minimum angle.



Produced by Triangle v. 1.4. (Ruppert−Chew hybrid.) 894 triangles.

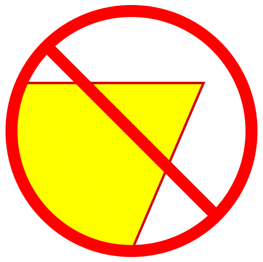


Produced by Triangle v. 1.6. (Chew's 2nd algorithm with off−centers.) 444 triangles.

Analysis

Analysis of Ruppert’s Algorithm

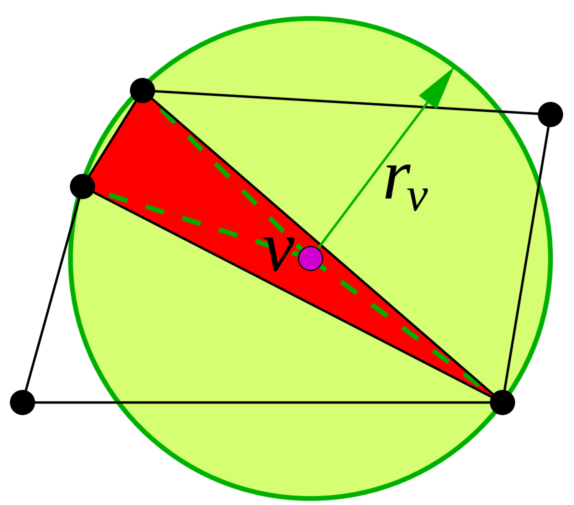
- **Restriction:** Input domain has no angle < 90°.
  (We’ll fix this later.)



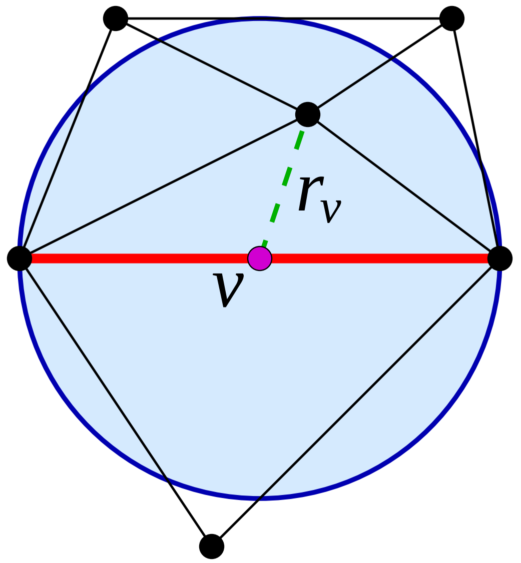
- **Goal:** Show that if we attack every skinny triangle, the algorithm eventually terminates. (It terminates if and only if there are no skinny triangles left.)

The Insertion Radius of a Vertex

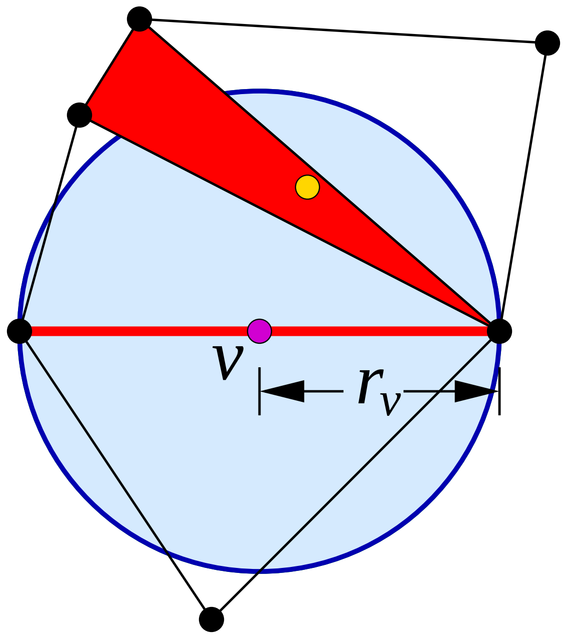
...is the length of the shortest edge adjoining a vertex immediately after the vertex appears in the mesh.



triangle circumcenter



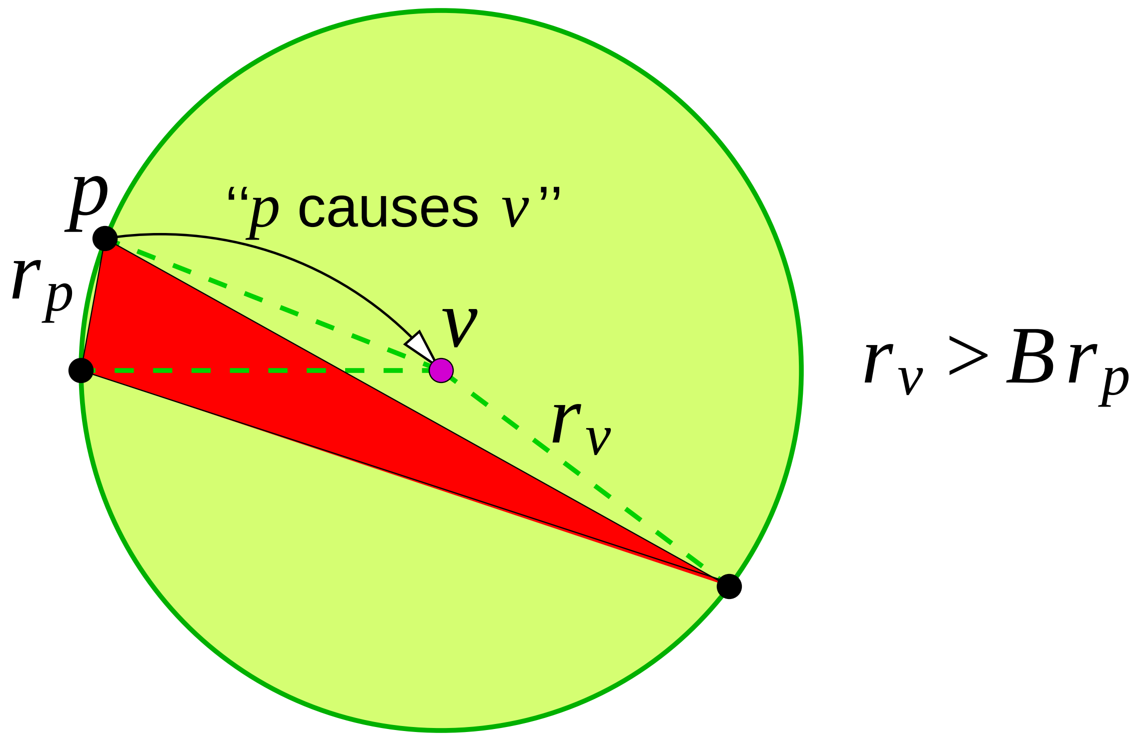
segment midpoint



segment midpoint

(Note: in a Delaunay triangulation, the insertion radius of a vertex is the distance to its nearest neighbor when it is inserted. In a constrained Delaunay triangulation, however, it's the distance to its nearest *visible* neighbor.)

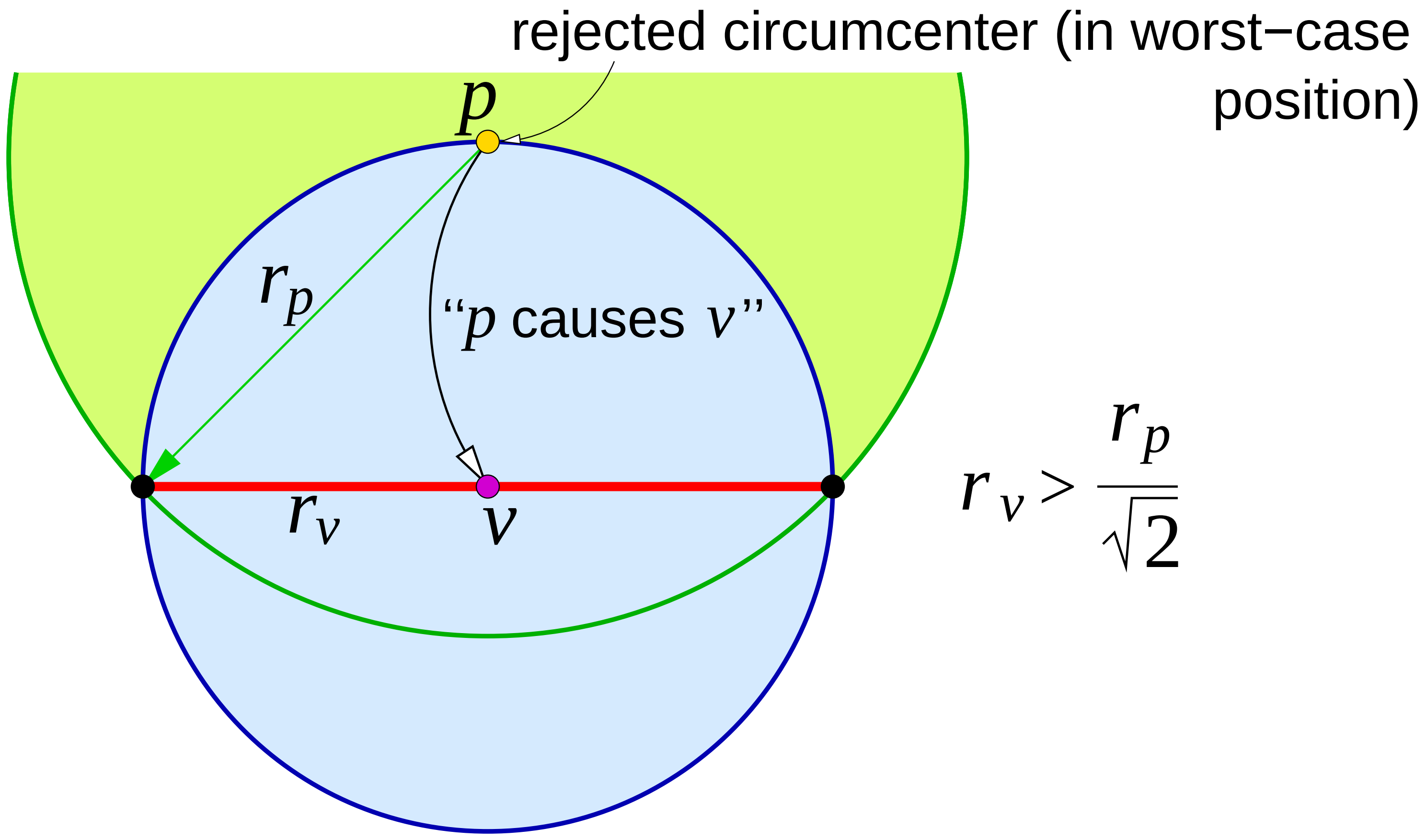
Insertion Radii of Circumcenters



Say a triangle is "skinny" if its circumradius−to−shortest edge ratio $> B$. Then its circumcenter v has insertion radius at least B times greater than that of some other vertex p.

(p is whichever endpoint of the short edge appeared in the mesh last. The inequality holds for off−centers too.)

Insertion Radii of Subsegment Midpoints


rejected circumcenter (in worst−case position)
p
r_p
"p causes v"
r_v
v
$r_v > \frac{r_p}{\sqrt{2}}$

The midpoint v has insertion radius at least $1/\sqrt{2}$ times that of the rejected circumcenter p.

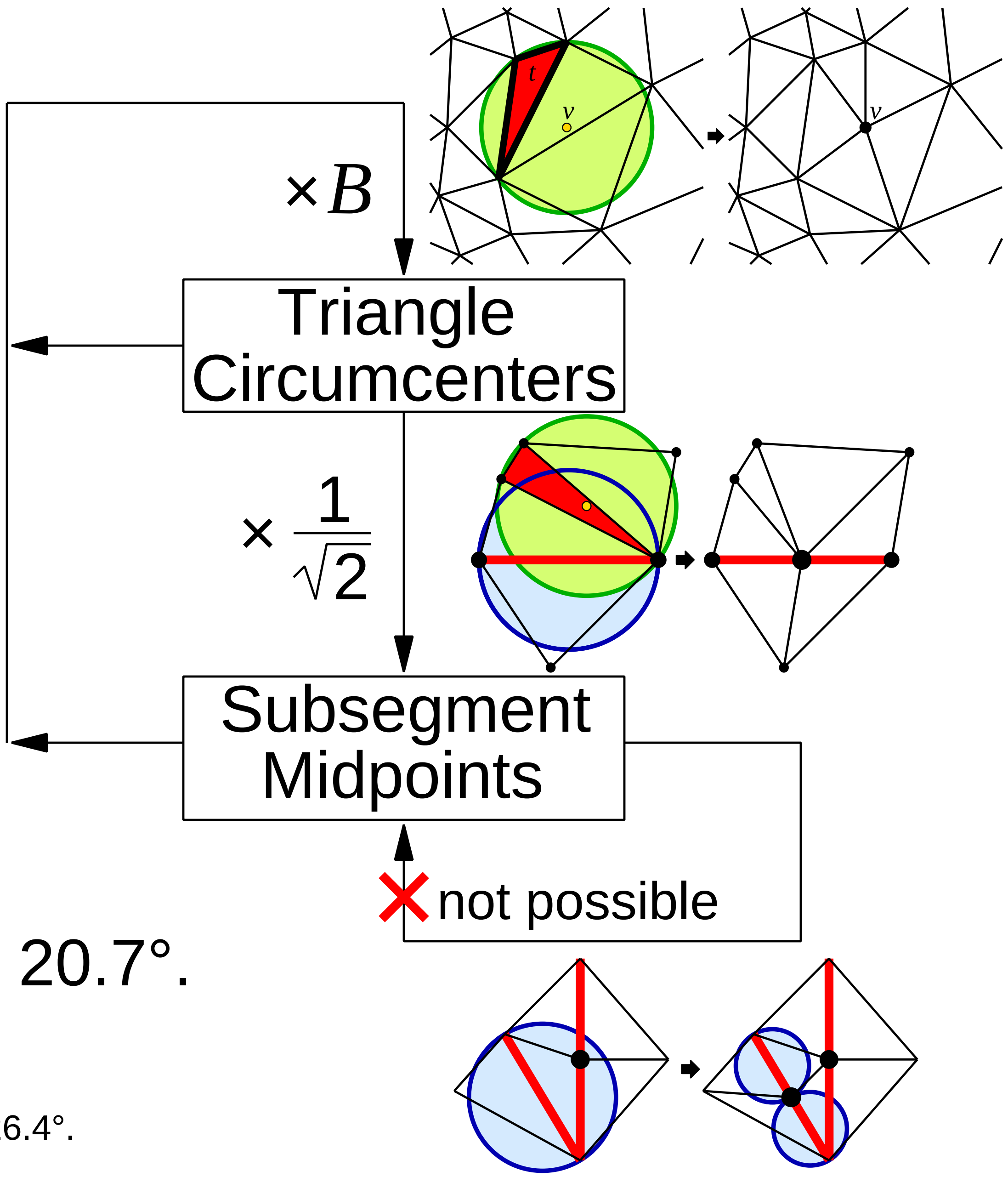
(This is the only step where the insertion radius can shrink. Fortunately, it can't shrink much.)

Goal: Avoid Cycle of Diminishing Edge Lengths

Multipliers (right) reflect smallest possible insertion radius of new vertex, relative to vertex that “caused” it.

Algorithm is guaranteed to terminate if no cycle exists with product less than 1.

We require $B \geq \sqrt{2}$.
Minimum angle can go up to 20.7°.

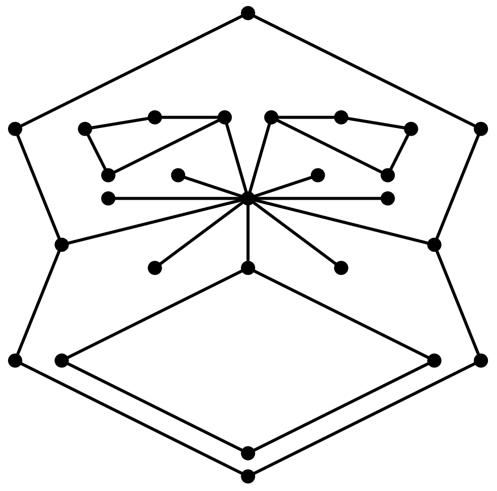
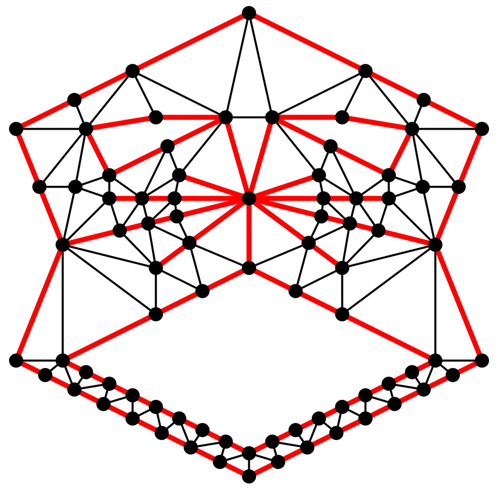


Miller, Pav, and Walkington improve this analysis to 26.4°.
For citation, see the Small Angles section.

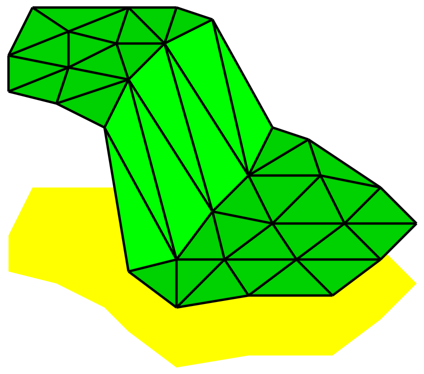
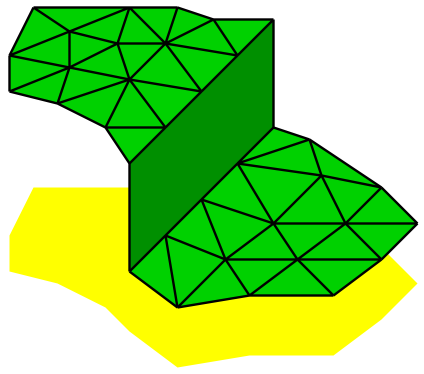
Constrained Delaunay

Delaunay triangulations are great, but sometimes you need to make sure edges or facets appear.

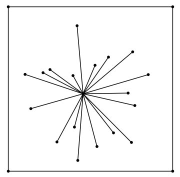
- Nonconvex shapes; internal boundaries

- Discontinuities in interpolated functions

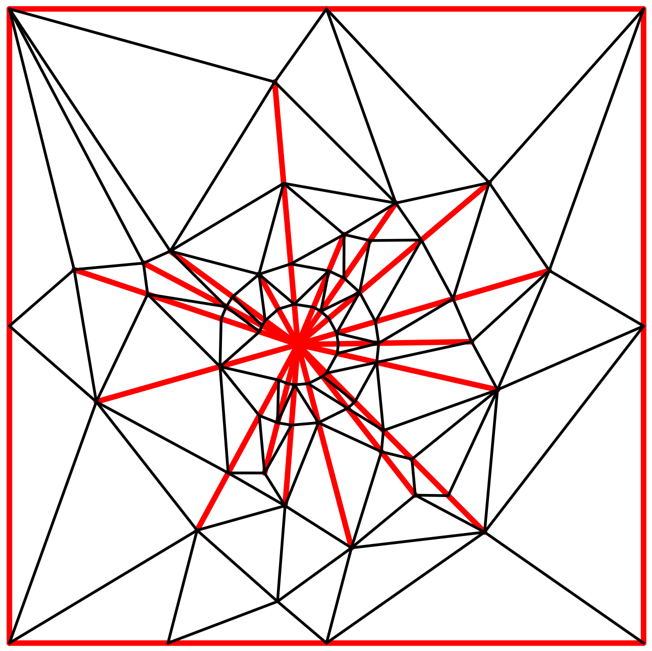



Three Alternatives for Recovering Segments



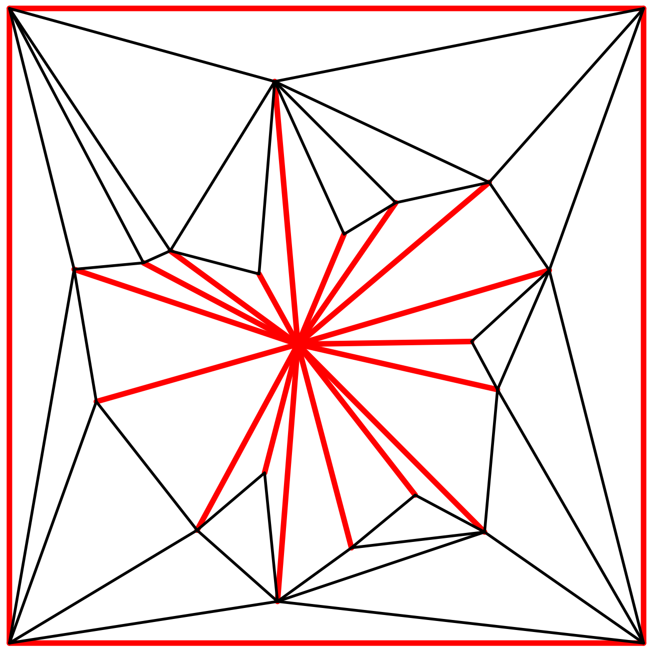
Conforming Delaunay triangulations

- Edges, triangles, and tetrahedra are all Delaunay.
- Worst case PSLG needs $\Omega(n^2)$ to $O(n^3)$ extra vertices.



"Almost Delaunay" triangulations

- Delaunay property compromised to recover boundary facets.
- What most heuristic 3D Delaunay meshing algorithms do.



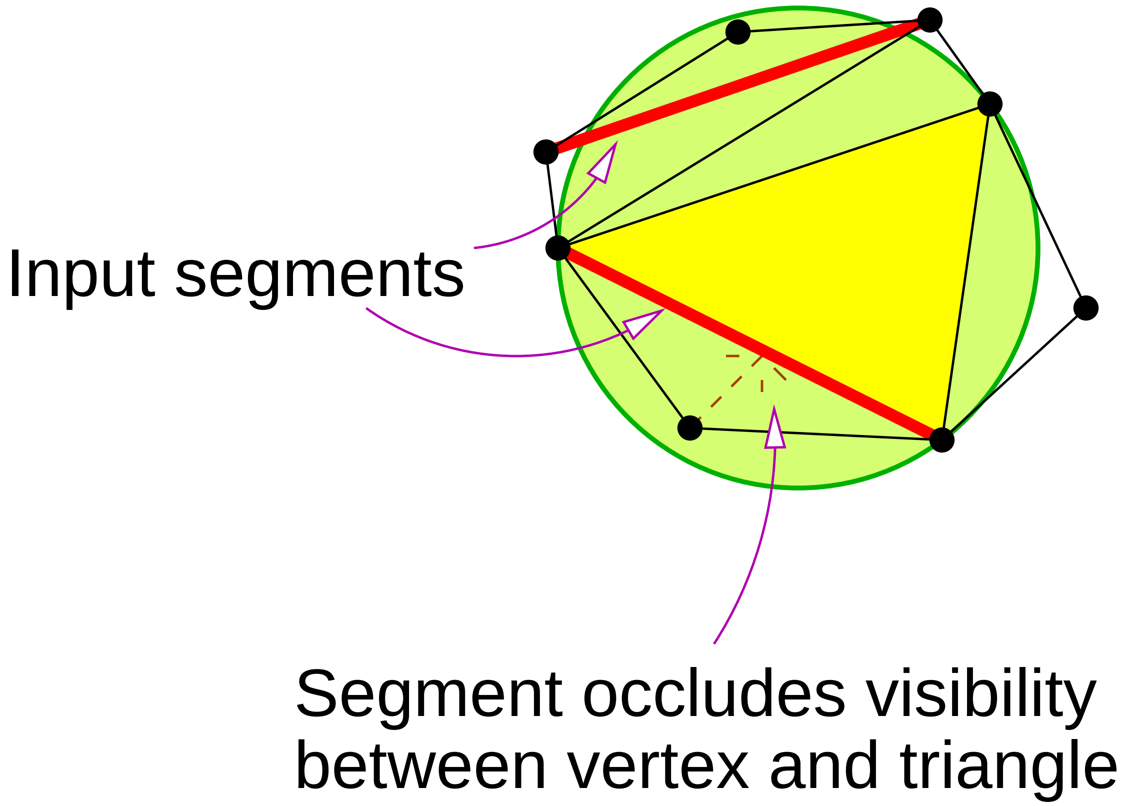
Constrained Delaunay triangulations (CDTs)

- Edges, triangles, and tetrahedra are constrained Delaunay or are domain boundaries.

Constrained Delaunay Triangle

A triangle is *constrained Delaunay* if

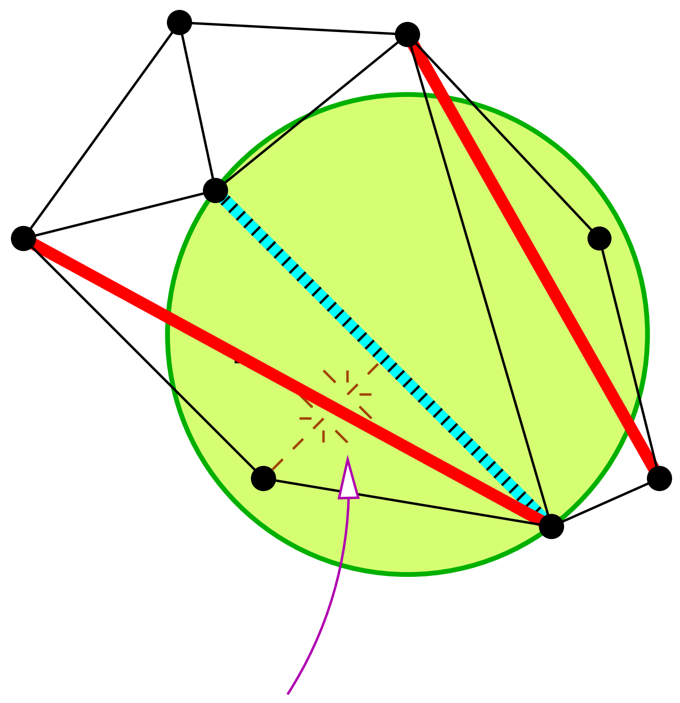
- its interior doesn’t intersect any input segment, and
- its circumcircle encloses no vertex visible from interior of triangle.



Constrained Delaunay Edge

An edge is *constrained Delaunay* if

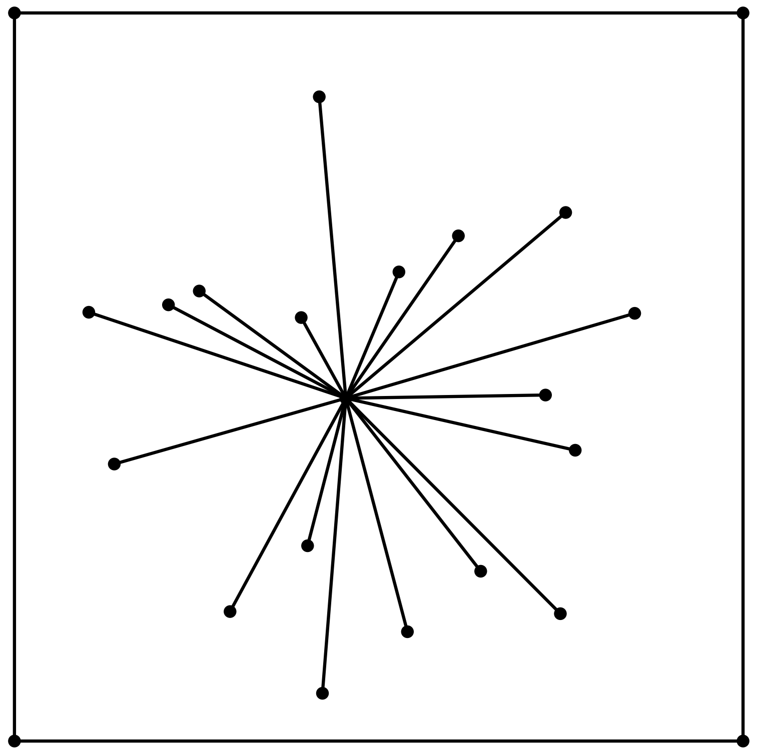
- it doesn't cross any input segment, and
- it has a circumcircle that encloses no vertex visible from interior of edge.

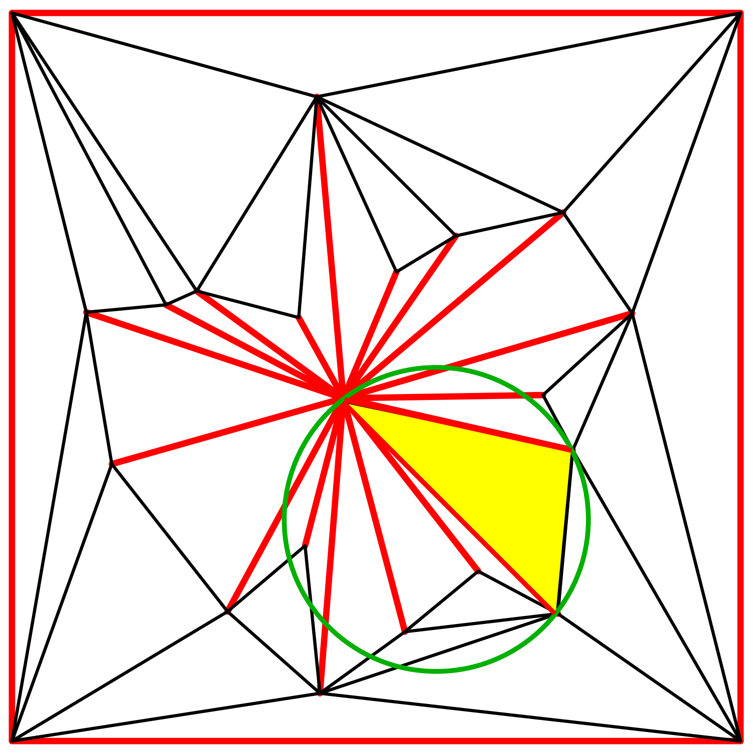


Segment occludes visibility between vertex and edge.

Constrained Delaunay Triangulations

...are triangulations entirely composed of constrained Delaunay triangles and edges, plus input segments.





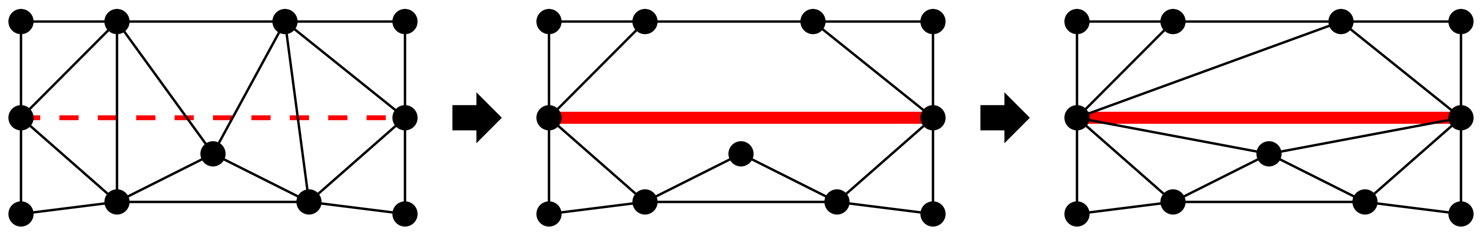
No need for stitching!

CDTs were introduced by Der−Tsai Lee and A. K. Lin, ‘‘Generalized Delaunay Triangulations for Planar Graphs,’’ Discrete & Computational Geometry **1**:201−217, 1986.

CDT Construction Algorithms

Folklore: start with DT; insert segments one by one. To insert a segment, delete the triangles it crosses; retriangulate the cavities by gift−wrapping. O(n^3) worst−case CDT construction time; usually faster in practice.

Marc Vigo Anglada, ‘‘An Improved Incremental Algorithm for Constructing Restricted Delaunay Triangulations,’’ Computers and Graphics **21**(2):215−223, March 1997. (Not the first person to think of this, but this paper is a good description.)



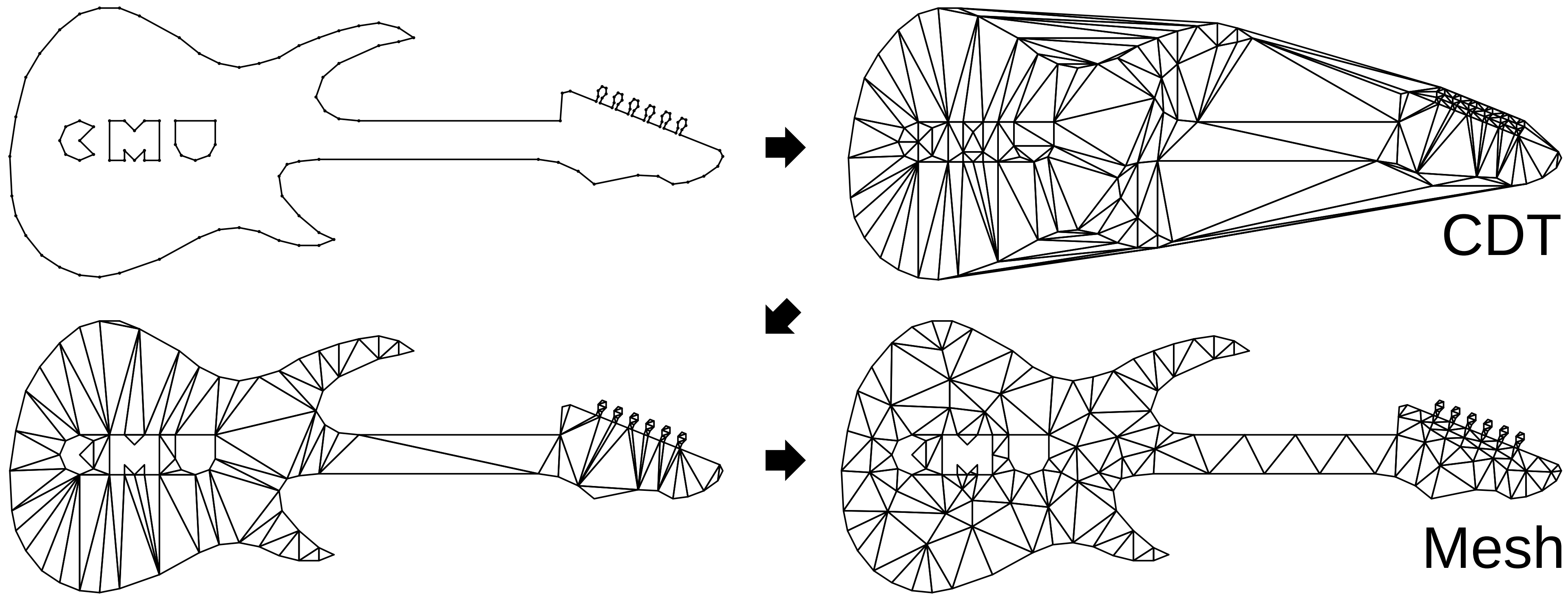
Faster: Optimal O(n log n) divide−and−conquer algorithms by Chew and Seidel. Harder to implement.

L. Paul Chew, ‘‘Constrained Delaunay Triangulations,’’ Algorithmica **4**(1):97−108,1989.

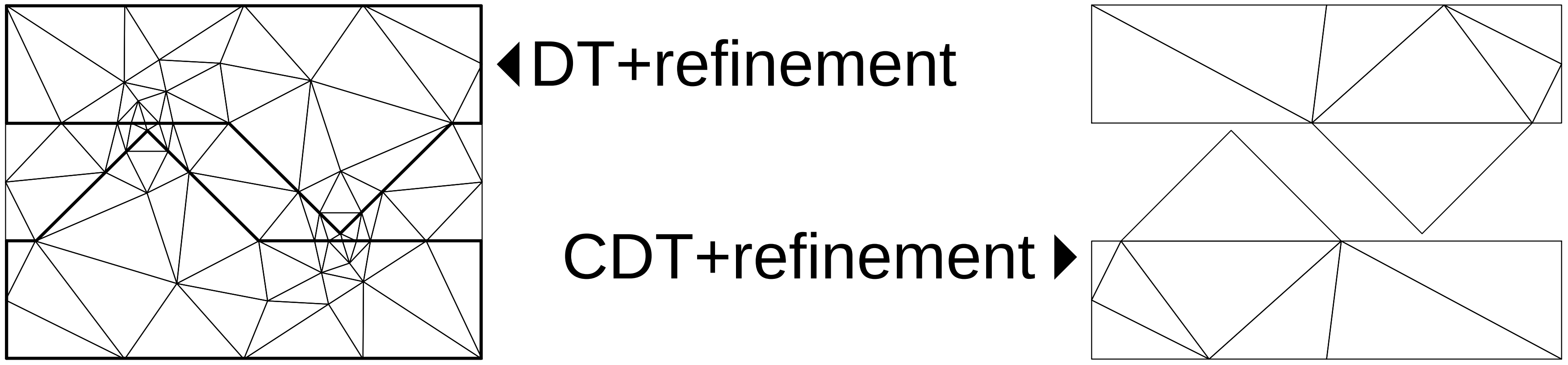
Raimund Seidel, ‘‘Constrained Delaunay Triangulations and Voronoi Diagrams with Obstacles,’’ *1978−1988 Ten Years IIG* (H. S. Poingratz and W. Schinerl, editors), pages 178−191, Institute for Information Processing, Graz University of Technology, 1988.

One Advantage of CDTs

Form CDT; remove triangles outside domain before refining.



Prevents overrefinement due to external features/small angles.

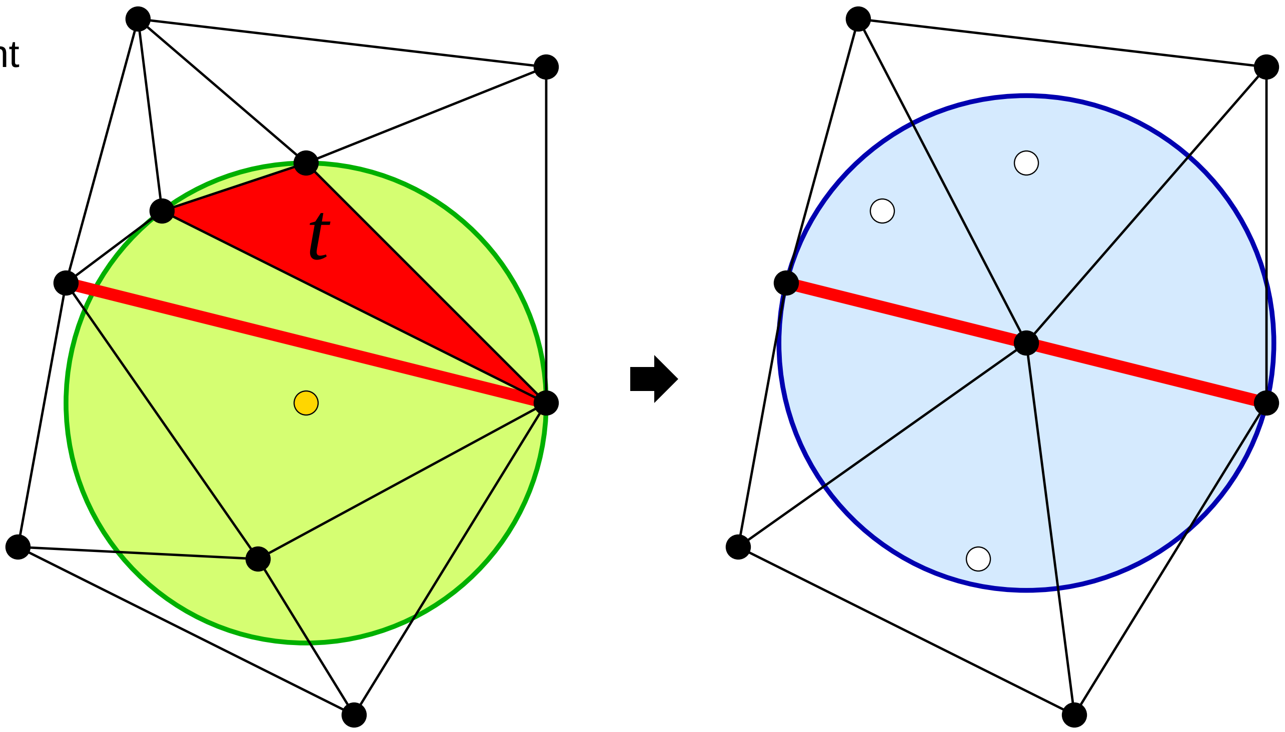


Chew's Second Delaunay Refinement Algorithm

L. Paul Chew, ''Guaranteed−Quality Mesh Generation for Curved Surfaces,'' Proceedings of the Ninth Annual Symposium on Computational Geometry, pages 274−280, May 1993. (Similar to Ruppert; developed independently.)

Uses CDTs. A subsegment is encroached (only) when it separates a skinny triangle from its circumcenter.

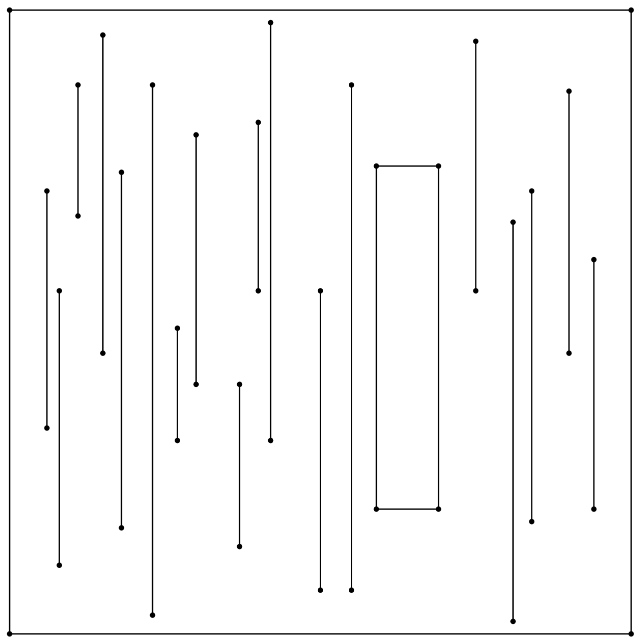
(Including when the circumcenter lies right on the subsegment.)

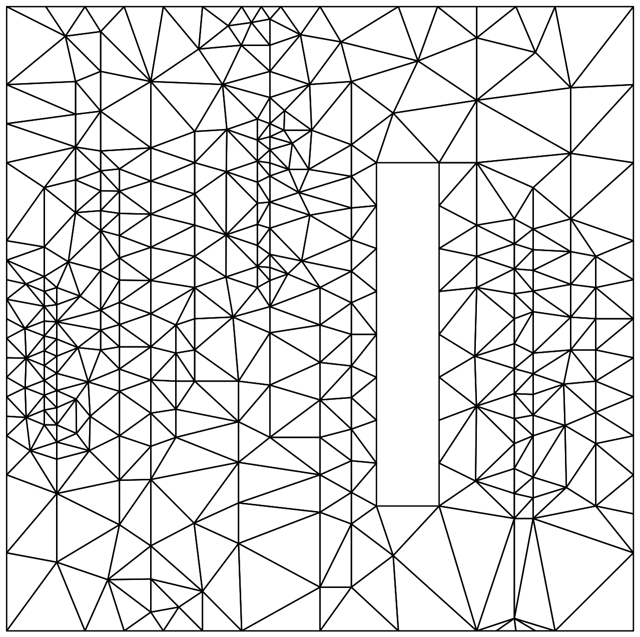


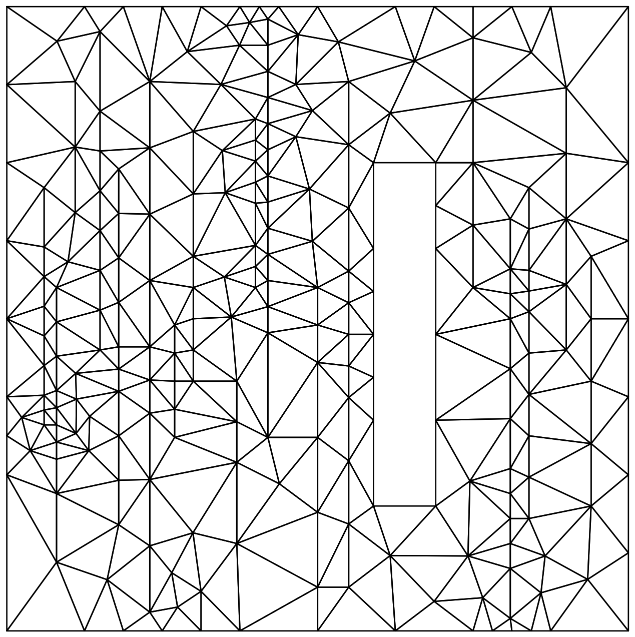
Delete all vertices from the encroached subsegment's diametral circle (except input vertices & vertices on segments). Split the subsegment.

Chew’s Second Delaunay Refinement Algorithm

L. Paul Chew, ‘‘Guaranteed−Quality Mesh Generation for Curved Surfaces,’’ Proceedings of the Ninth Annual Symposium on Computational Geometry, pages 274−280, May 1993.







Ruppert: 559 triangles. Chew: 423 triangles.

With an extra trick, Chew guarantees 30° minimum angle.

(Chew’s algorithm occasionally trisects a subsegment instead of bisecting it. Unnecessary in practice.)

If angle bound is reduced below 26.5°, good grading is theoretically guaranteed. (Compare to Ruppert’s 20.7°.)

This fact from Jonathan Richard Shewchuk, ‘‘Delaunay Refinement Algorithms for Triangular Mesh Generation,’’ Computational Geometry: Theory and Applications **22** (1−3):21−74, May 2002. An analysis technique of Miller, Pav, and Walkington improves this bound to 28.6°. (Compare to their bound of 26.4° for Ruppert’s algorithm.)

Curves

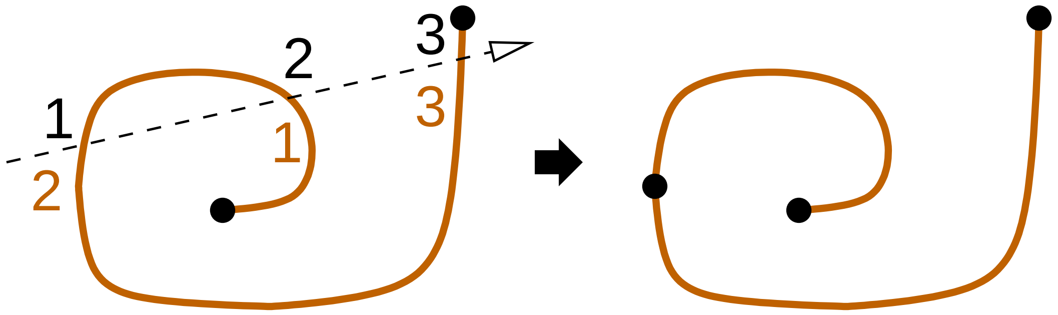
Curved Boundaries: Boivin and Ollivier−Gooch

Charles Boivin and Carl Ollivier−Gooch, ‘‘Guaranteed−Quality Triangular Mesh Generation for Domains with Curved Boundaries,’’ International Journal for Numerical Methods in Engineering **55** (10):1185−1213, 20 August 2002.

Warning: the following treatment is adapted and is not true to Boivin/Ollivier−Gooch, but the main ideas are theirs. The theoretical angle bound is slightly improved here.

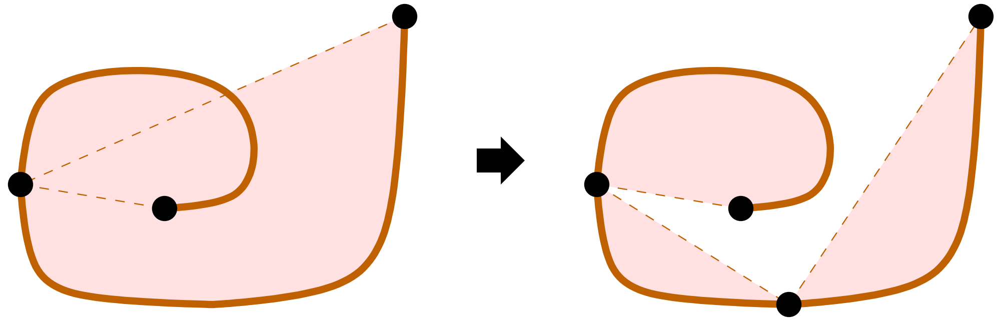
Preprocess curves, splitting them into subcurves, so that

- collinear points on any subcurve occur in linear order;



(Boivin and Ollivier−Gooch are more restrictive.
They require a subcurve’s tangent direction to vary by no more than 60°. The modified algorithm here does not.)

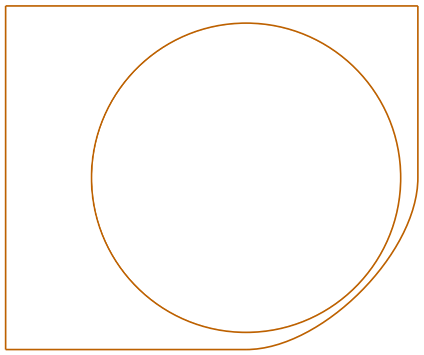
- no two subcurves’ convex hulls intersect, except at shared endpoints.



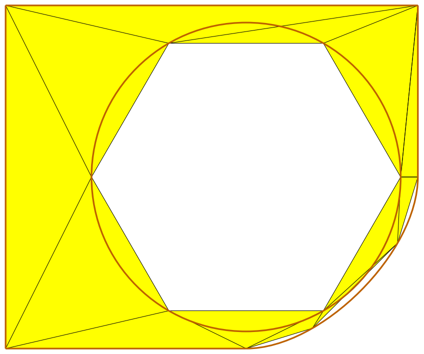
(Boivin and Ollivier−Gooch are less restrictive.
See their article for how to handle intersecting convex hulls during Delaunay refinement.)

Curved Boundaries: Boivin and Ollivier−Gooch

Charles Boivin and Carl Ollivier−Gooch, ‘‘Guaranteed−Quality Triangular Mesh Generation for Domains with Curved Boundaries,’’ International Journal for Numerical Methods in Engineering **55** (10):1185−1213, 20 August 2002.



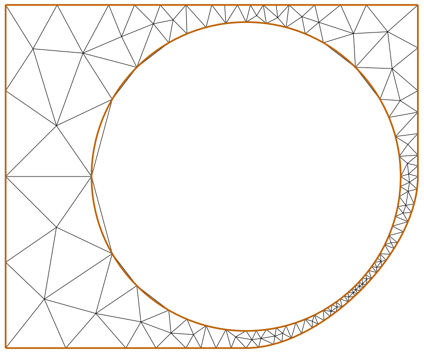
Domain.



Preprocess curves.

Approximate each subcurve with
a subsegment.

Construct CDT.

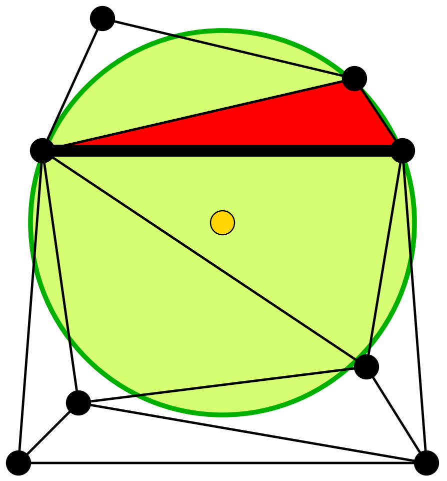


Delaunay refinement.

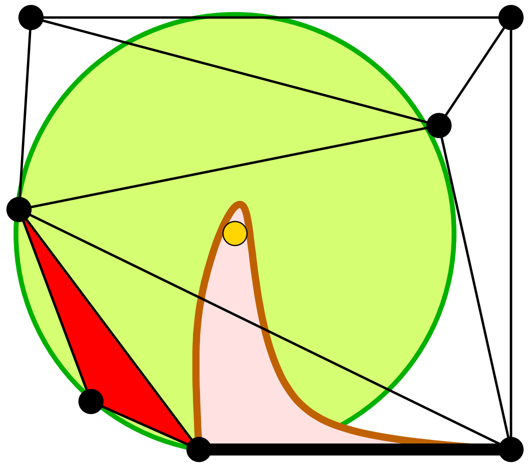
Curved Boundaries: Encroachment

Encroachment is like in Chew's second algorithm.
A subsegment is encroached if

- it separates a skinny triangle from its circumcenter, or



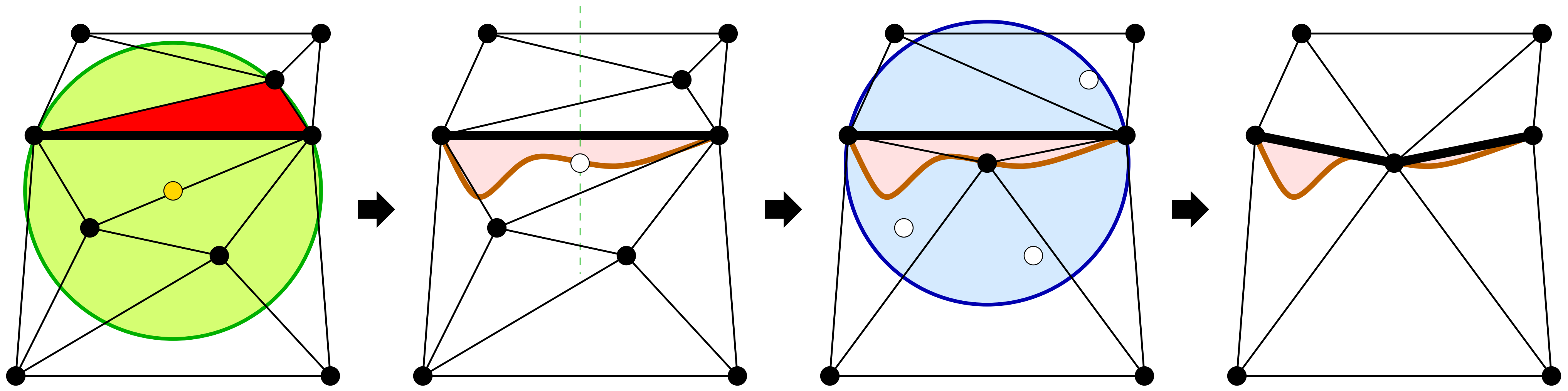
- a skinny triangle's circumcenter lies between the subsegment and its subcurve.
  (Or right on the subsegment or subcurve.)



Curved Boundaries: Encroachment

Case 1:

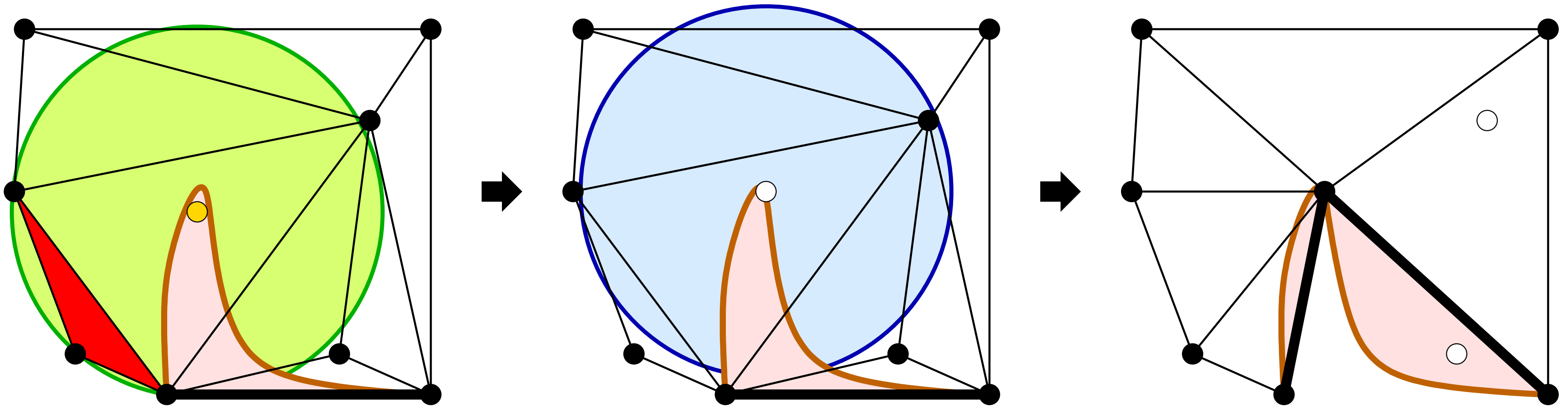
- Find a point where curve intersects segment bisector.
- Draw circle around new point through endpoints.
- Delete all vertices in circle (except input vertices & vertices on segments).
- Insert new vertex.
- Insert new subsegments.
- Unlock old segment; flip to constrained Delaunay.



Curved Boundaries: Encroachment

Case 2:

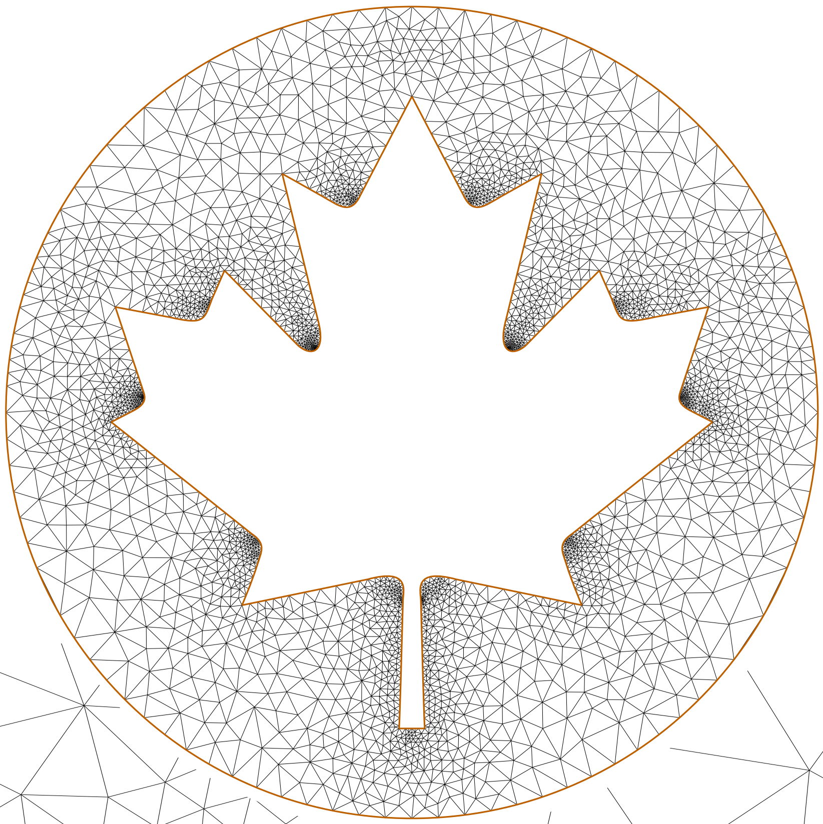
- Find a point on the curve no closer to either endpoint than the skinny triangle's circumcenter is.
- Draw circle around new point; radius = circumradius.
- Delete all vertices in circle (except...).
- Insert new vertex & subsegments.
- Delete all vertices inside the subsegment triangle.
- Unlock old subsegment.

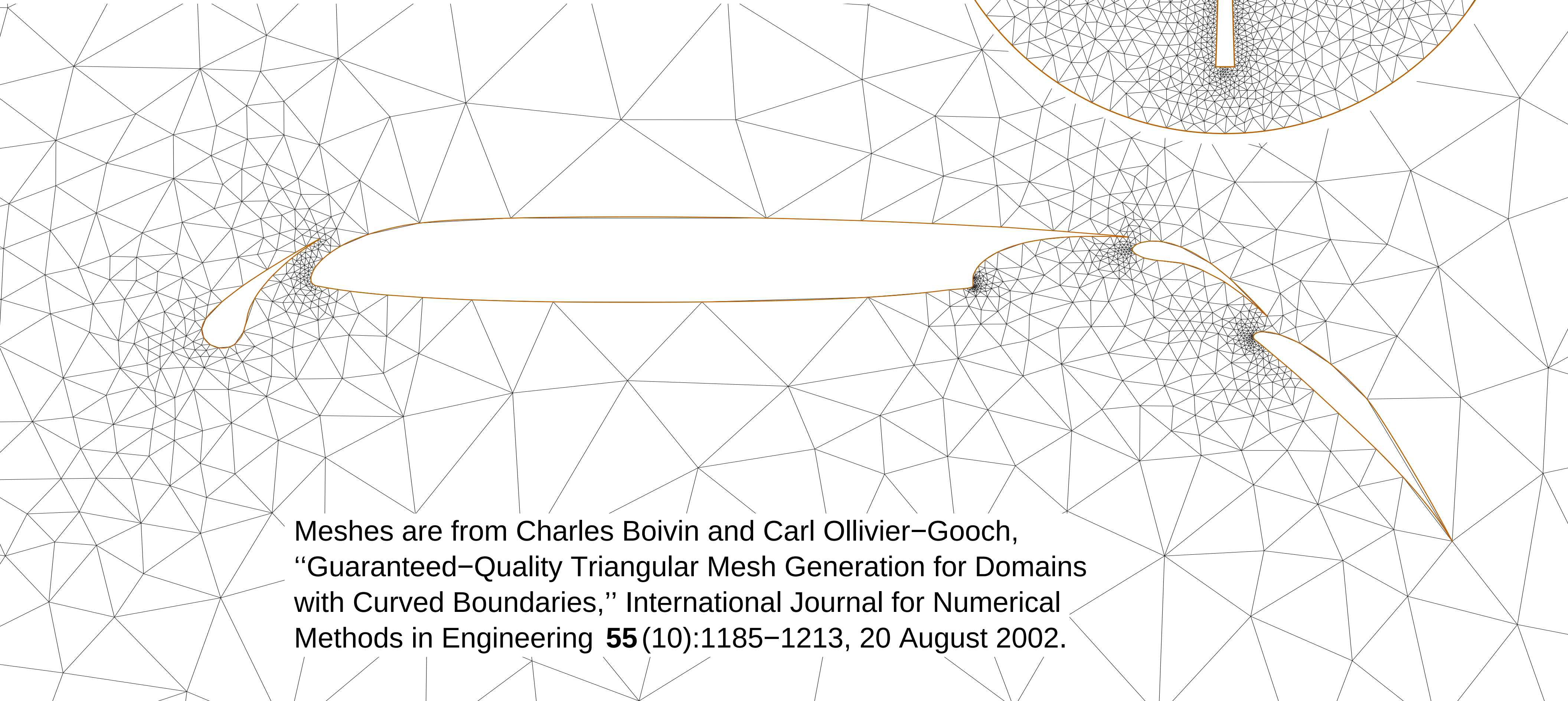


Curved Boundaries

Guaranteed termination for minimum angle bound up to 26.5° (like Chew’s second algorithm).

Guaranteed good grading.

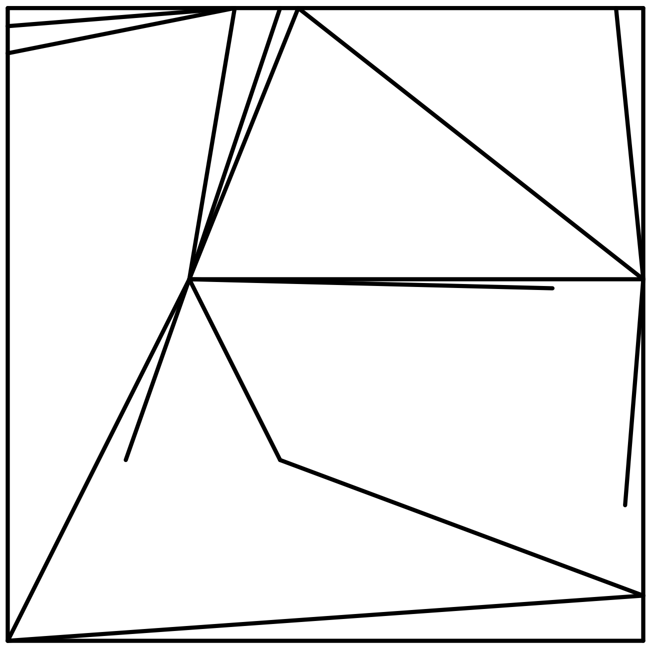




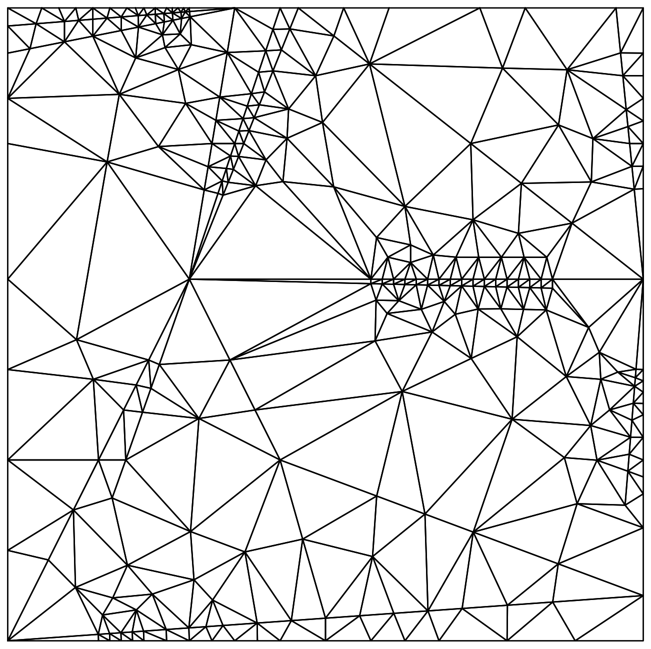
Meshes are from Charles Boivin and Carl Ollivier−Gooch, ‘‘Guaranteed−Quality Triangular Mesh Generation for Domains with Curved Boundaries,’’ International Journal for Numerical Methods in Engineering **55** (10):1185−1213, 20 August 2002.

Small Angles

Domains with Small Angles



Suppose the mesh must exactly fit the input. Small angles between adjoining input segments cannot be removed.



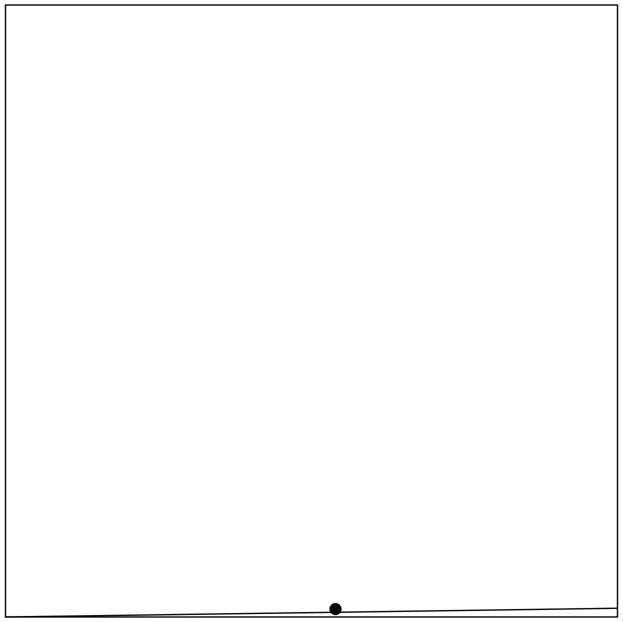
Problem: Create a triangular mesh that has no *new* angle less than θ. (For instance, 26°.)

A Negative Result

For a proof, see my ‘‘Delaunay Refinement Algorithms for Triangular Mesh Generation,’’ cited on page 67.

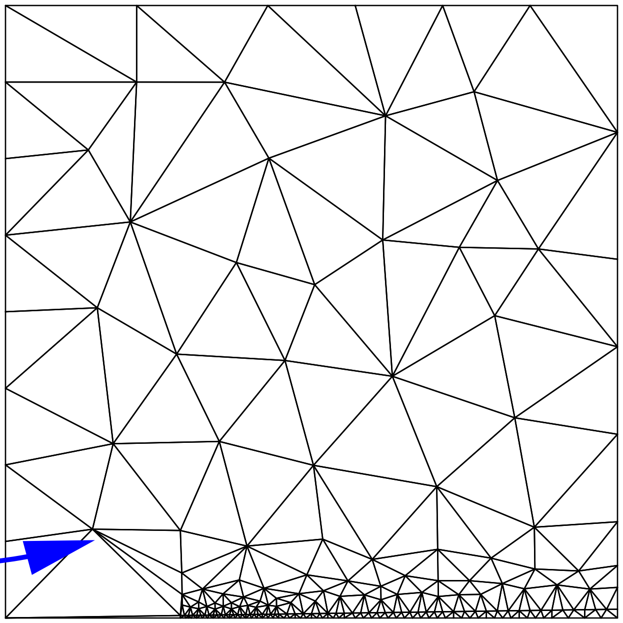
This problem has no solution!

Counterexample puts small input angle next to large input angle.



If the small angle is < 0.24°, **no** algorithm can mesh this PSLG without creating a new angle less than θ = 26°.

You can remove these small angles, but others will pop up to take their place!



An algorithm has to decide when and where to give up.

Goal: judge which skinny triangles are hopeless, and which skinny triangles you should attack.

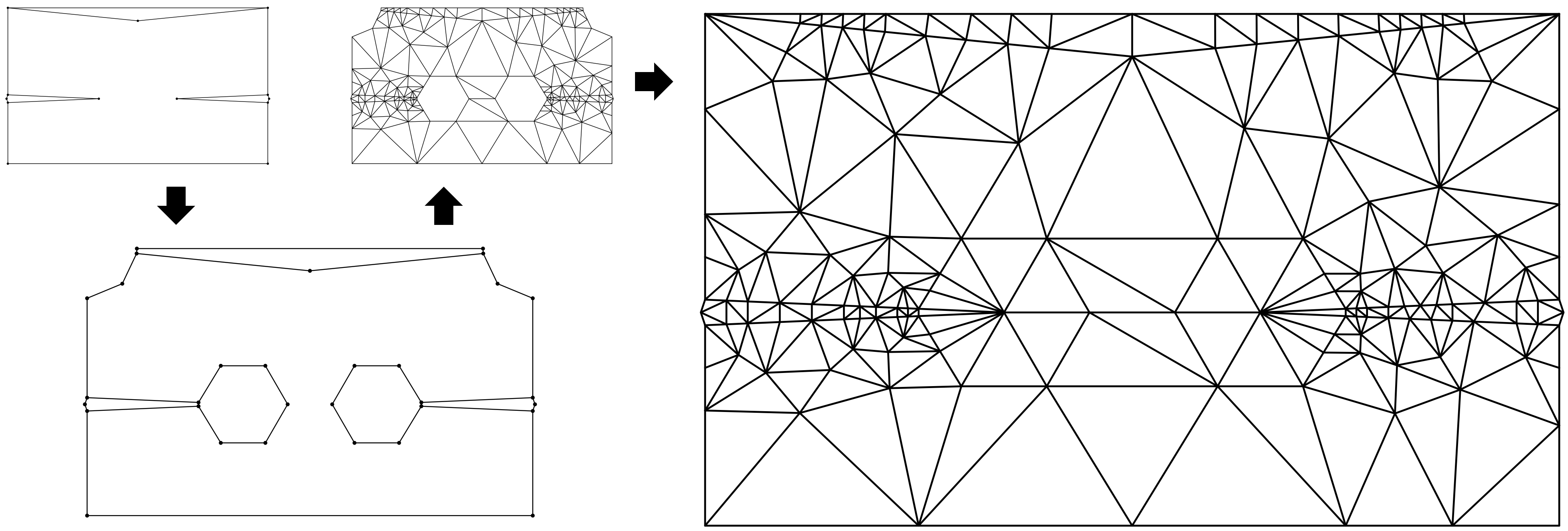
If we can’t demand no new angle less than θ, what can we demand?

- *Except* ‘‘near’’ small input angles, no angle is less than θ (say, 26.5°).
- No angle is less than the smallest nearby input angle.
- No angle is greater than 180°− 2 θ

Algorithms

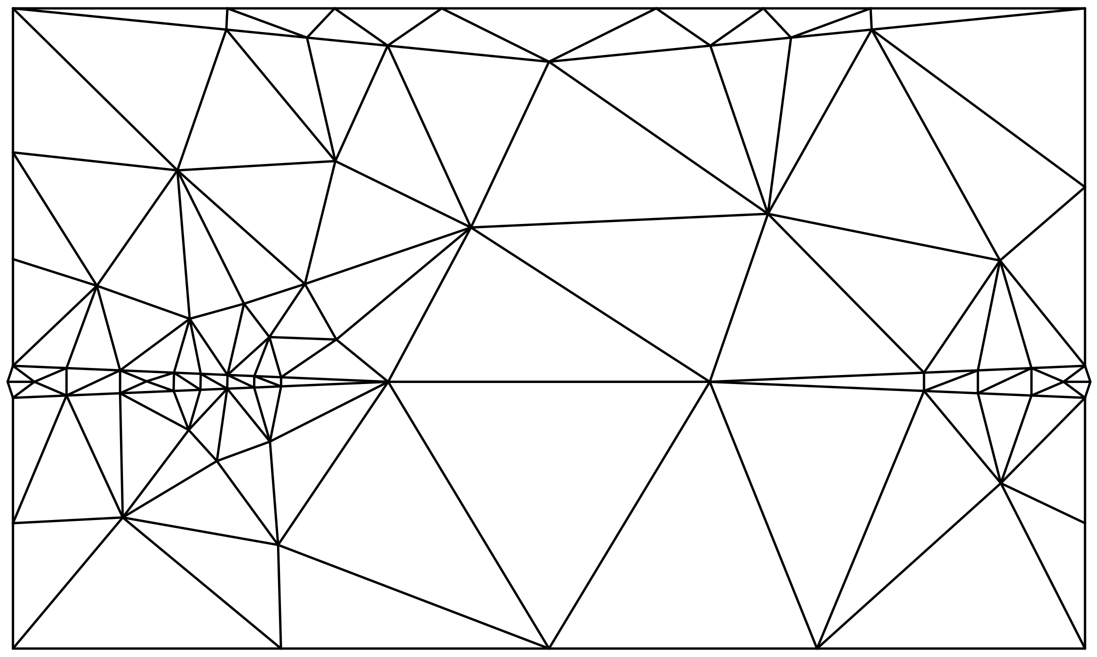
- Corner−lopping. See Bern−Eppstein−Gilbert and Ruppert.
- Better: Mine. Uses CDTs. No bound on max angle.
  Jonathan Richard Shewchuk, ‘‘Delaunay Refinement Algorithms for Triangular Mesh Generation,’’ Computational Geometry: Theory and Applications **22** (1−3):21−74, May 2002. (Also in 16th Symp. on Comp. Geometry, 2000.)
- Best: Miller−Pav−Walkington. Works with DTs or CDTs.
  No angle greater than 180°− 2 θ; better bound on minimum angle; easiest to implement; guaranteed good grading.
  Gary L. Miller, Steven E. Pav, and Noel J. Walkington, ‘‘When and Why Ruppert’s Algorithm Works,’’ Twelfth International Meshing Roundtable, pages 91−102, September 2003.

Corner–Lopping

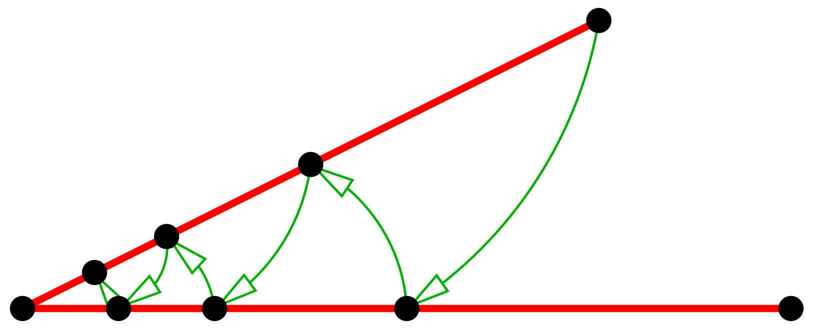


It’s a simple idea, but we can do better.

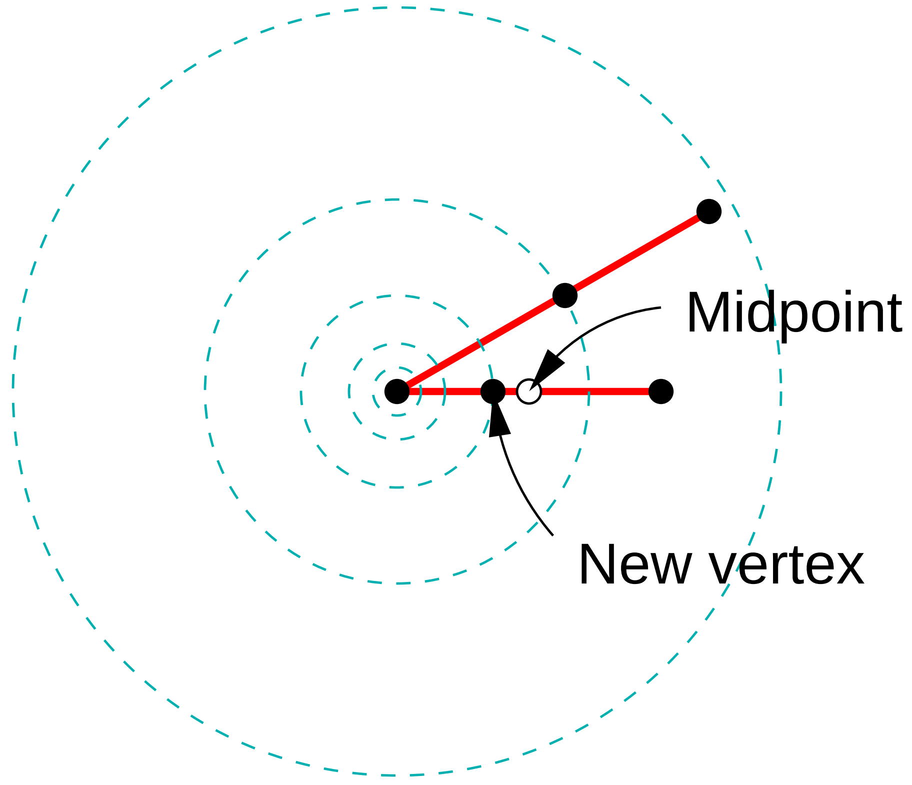
Generated by Miller–Pav–Walkington algorithm, as implemented in Triangle



Runaway Encroachment

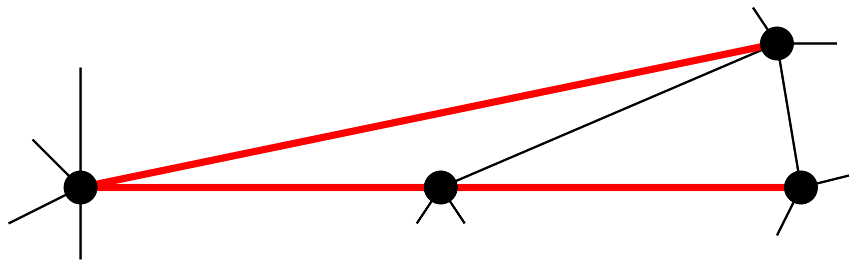


Problem: If angle < 45°, an endless cycle of mutual encroachment can occur.

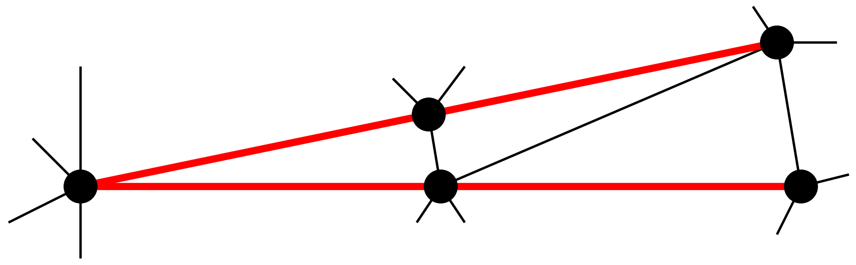


Ruppert’s solution: Split segments at concentric circular shells whose radii are powers of two.

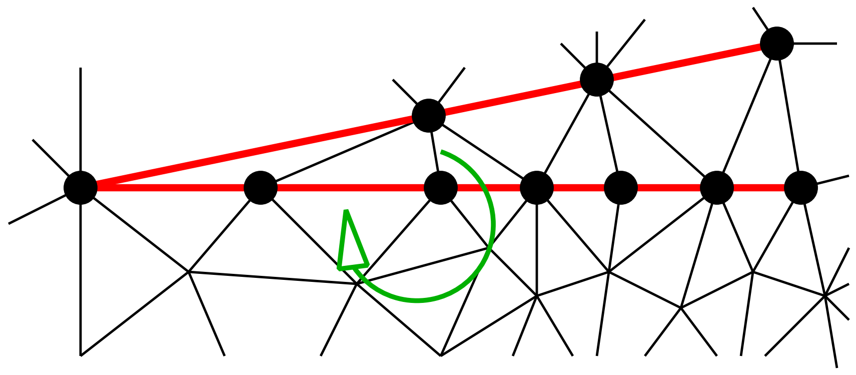
Small angles are "edge length reducers."



A subsegment is split.
The new vertex encroaches upon the other subsegment.



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



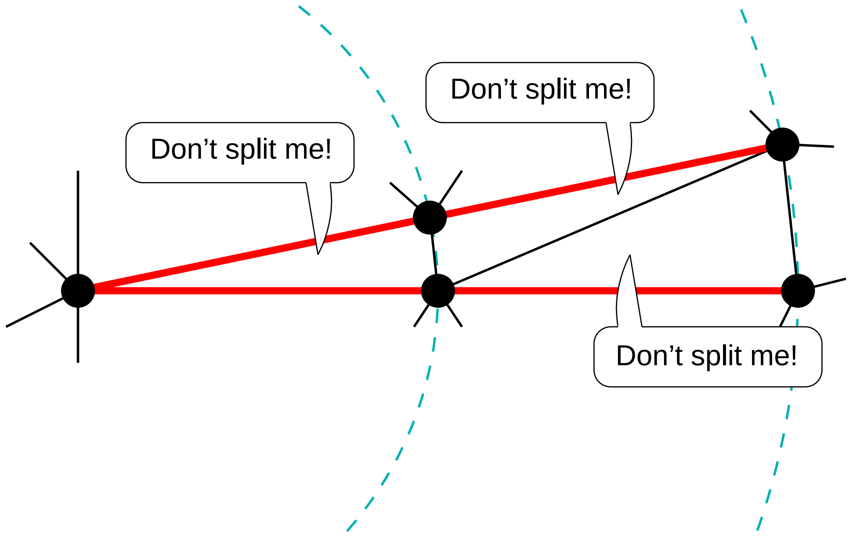
Skinny triangles engender more new vertices. Small edge lengths propagate around and split the subsegment again!

The Miller–Pav–Walkington Algorithm

Gary L. Miller, Steven E. Pav, and Noel J. Walkington, ''When and Why Ruppert's Algorithm Works,'' Twelfth International Meshing Roundtable, pages 91−102, September 2003. See also Pav's Ph.D. dissertation.

Make one tiny adjustment to Delaunay refinement with concentric circular shells:

Never attack a skinny triangle whose shortest edge subtends a small input angle and has both endpoints on circular shells.

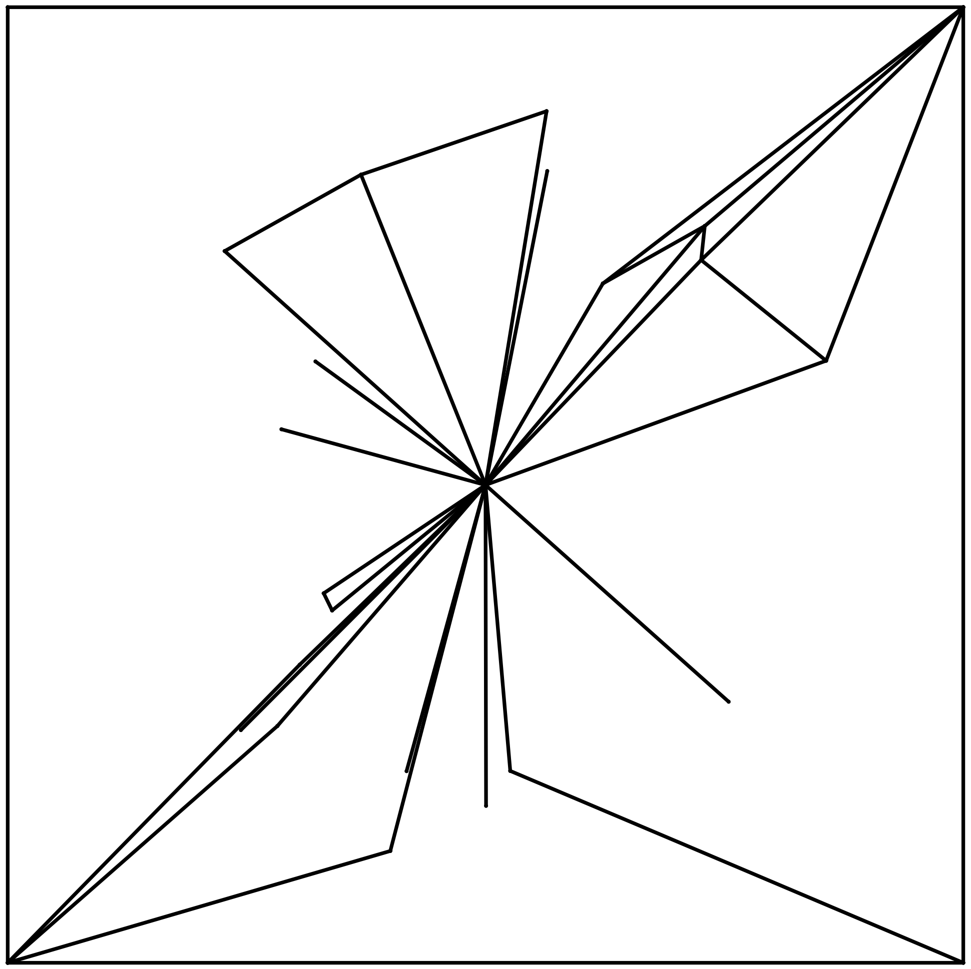


Guaranteed to terminate with no *other* skinny triangles (< 26.45°); no large angles (> 127.1°); good grading.

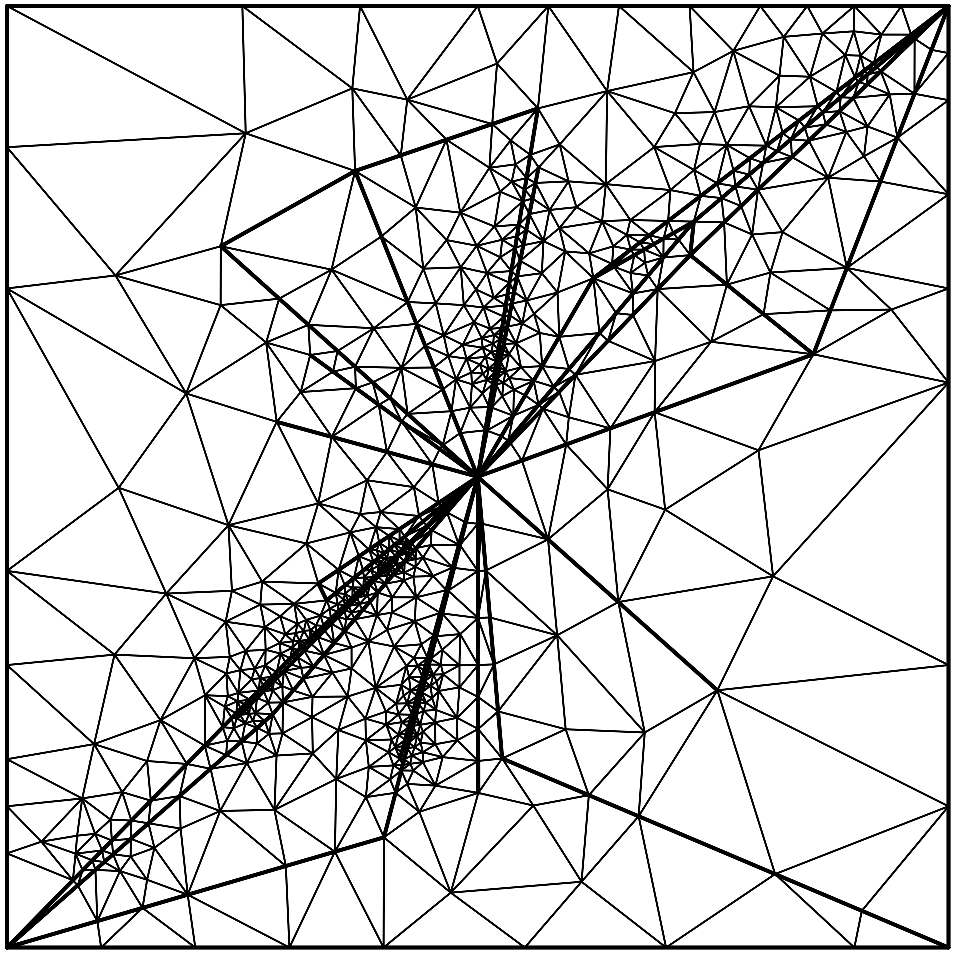
Combined with Chew's algorithm, we can demand most angles be > 28.6° and all angles be < 122.8°.

The Miller–Pav–Walkington Algorithm

Gary L. Miller, Steven E. Pav, and Noel J. Walkington, ‘‘When and Why Ruppert’s Algorithm Works,’’ Twelfth International Meshing Roundtable, pages 91−102, September 2003.



Most angles > 26.45°. All angles < 127.1°.



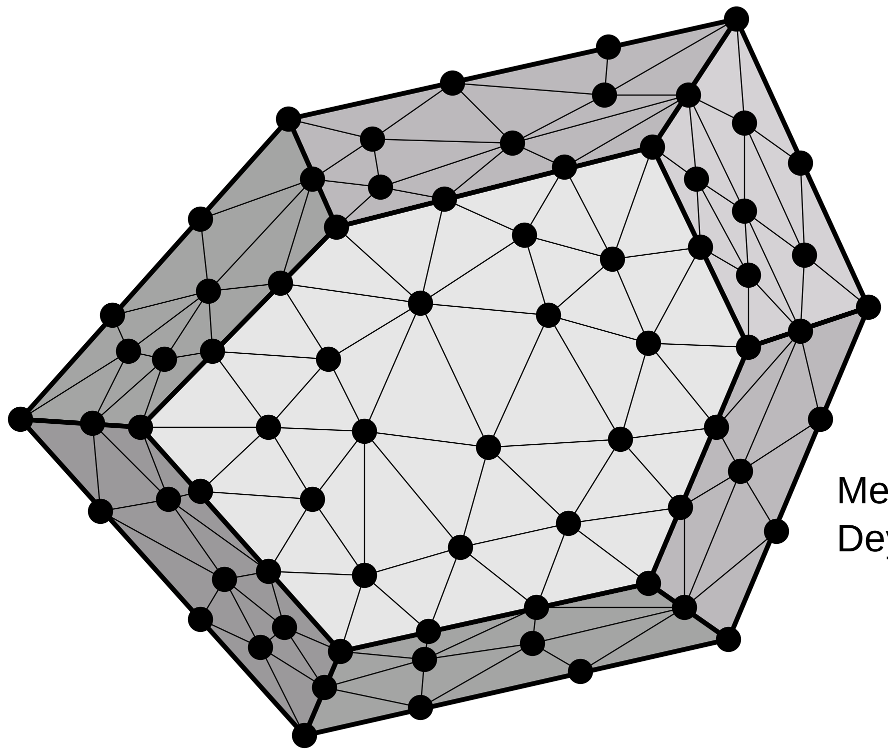
Generated by Triangle v. 1.6.

3D Delaunay Refinement

3D Delaunay Refinement

Tamal Krishna Dey, Chanderjit L. Bajaj, and Kokichi Sugihara, ‘‘On Good Triangulations in Three Dimensions,’’ International Journal of Computational Geometry and Applications **2**(1):75−95, 1992.

The first 3D Delaunay refinement algorithm works only for convex polyhedra. Like Chew’s first algorithm, it pre−discretizes the boundary so Delaunay refinement will work.



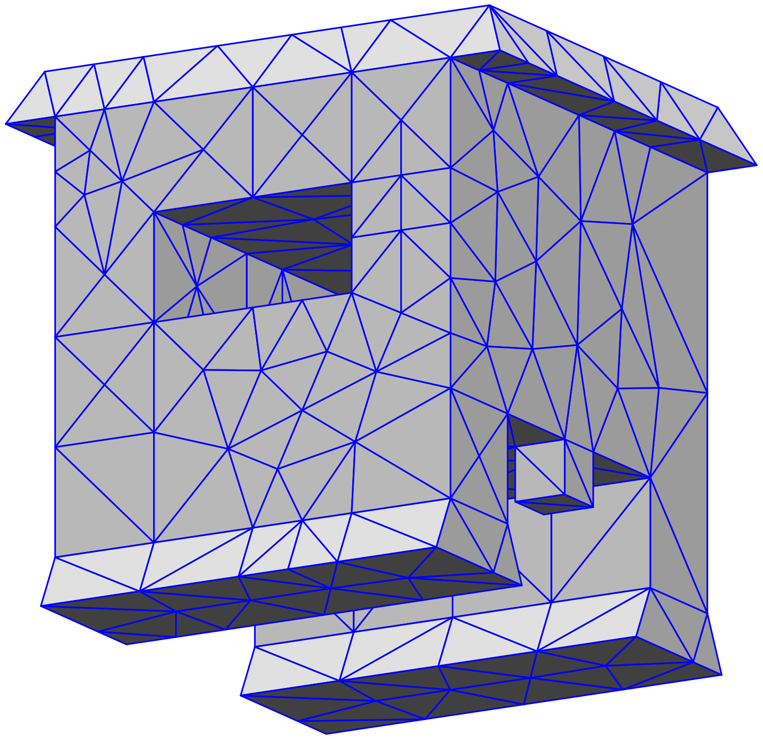
Mesh transcribed from Dey−Bajaj−Sugihara article.

3D Delaunay Refinement

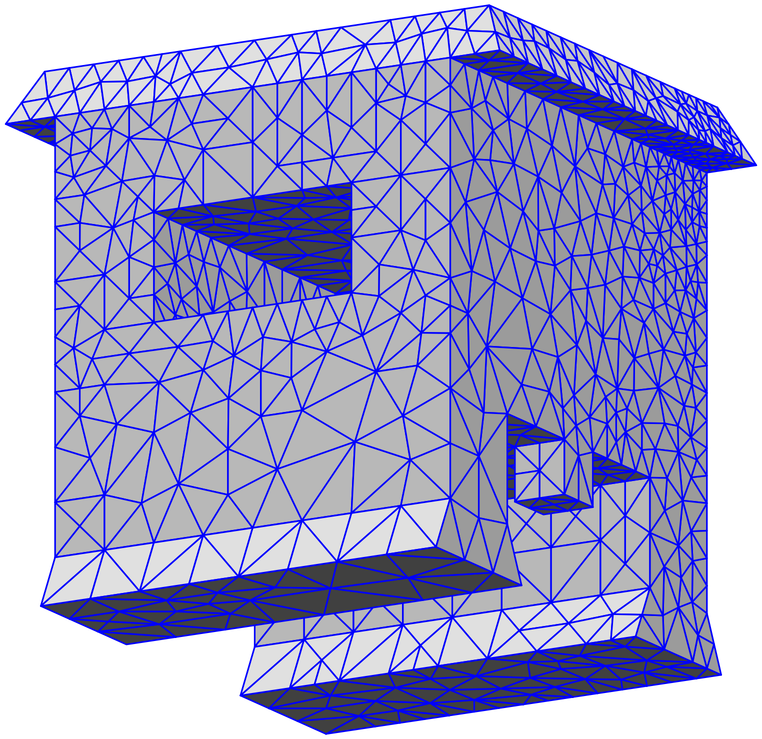
Jonathan Richard Shewchuk, ‘‘Tetrahedral Mesh Generation by Delaunay Refinement,’’ Proceedings of the Fourteenth Annual Symposium on Computational Geometry, pages 86−95, June 1998.

Here’s an algorithm that works on non−convex domains with non−manifold boundaries.

- **Restriction:** Input domain has no angle < 90°. (Neither a plane angle nor a dihedral angle.)
- You choose the maximum acceptable circumradius−to−shortest edge ratio B, as low as 2. Can go lower in practice.



B = 1.2, 334 vertices, 1009 tetrahedra.

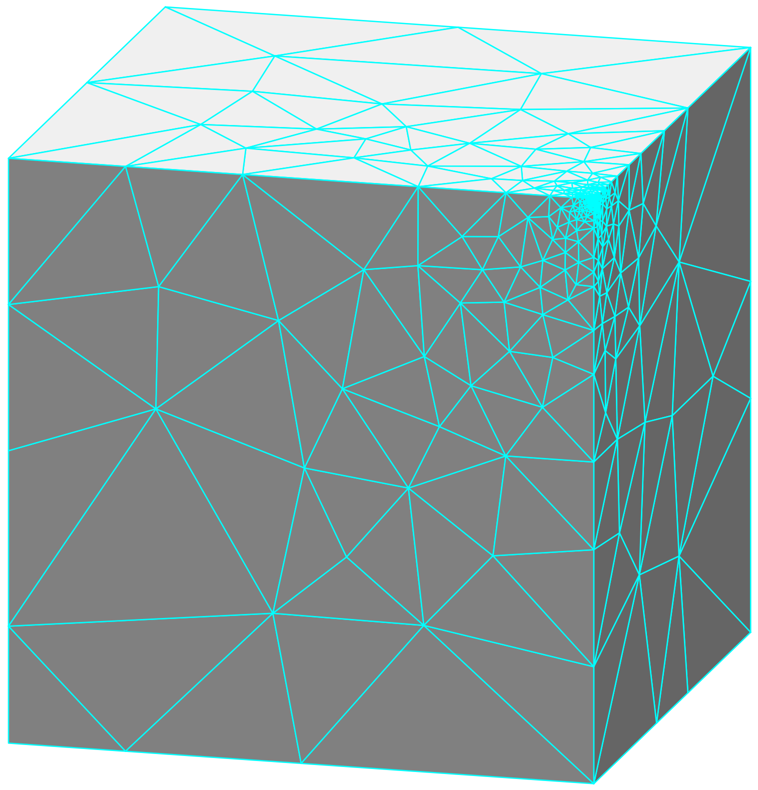


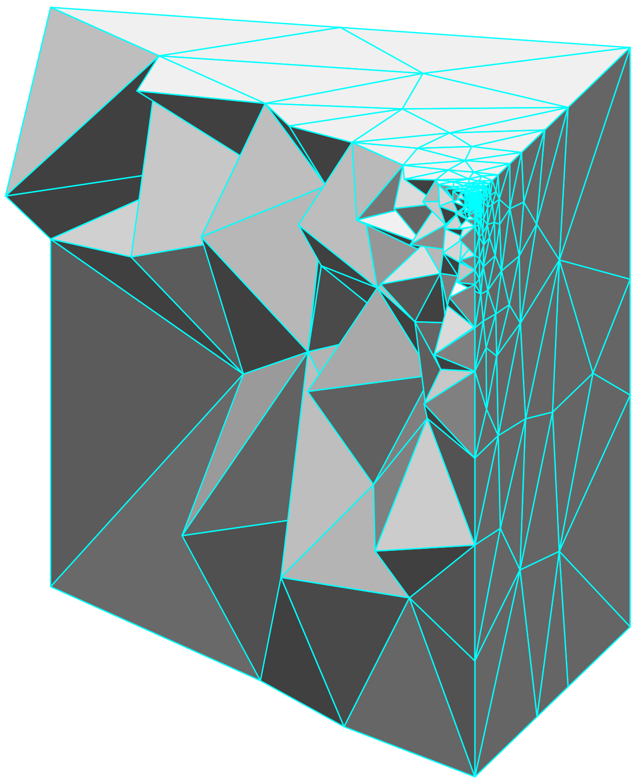
B = 1.041, 3144 vertices, 13969 tetrahedra.

3D Delaunay Refinement

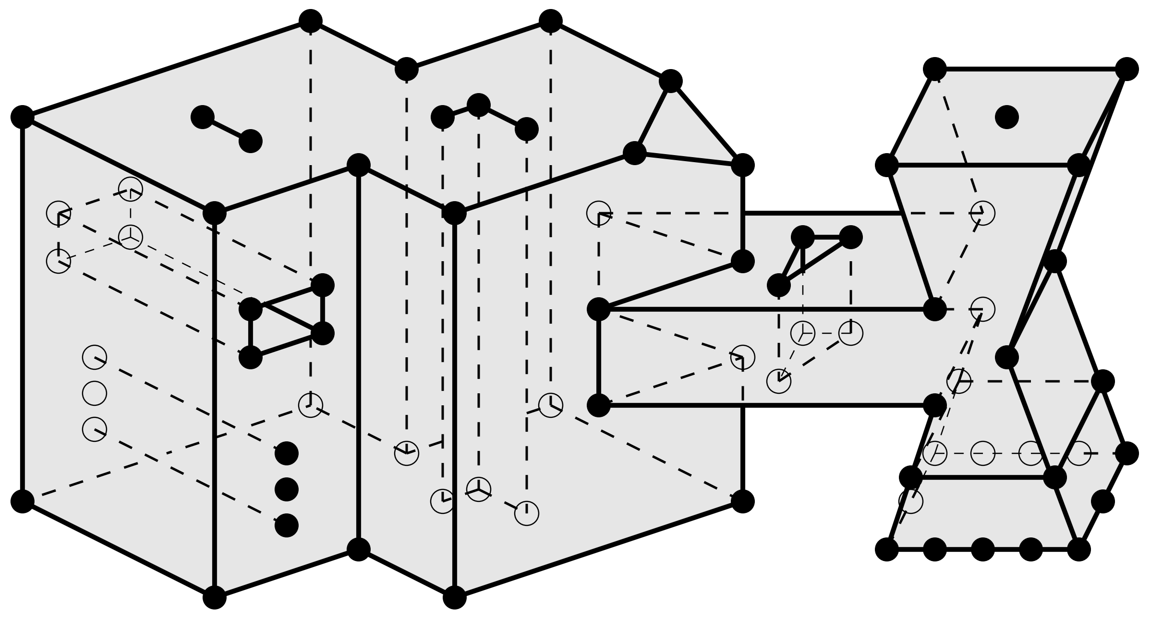
Jonathan Richard Shewchuk, ‘‘Tetrahedral Mesh Generation by Delaunay Refinement,’’ Proceedings of the Fourteenth Annual Symposium on Computational Geometry, pages 86−95, June 1998.

Provably good grading: all edge lengths are proportional to the ‘‘local feature size.’’ Theoretical grading guarantee deteriorates as $B \rightarrow 2$ (your maximum acceptable circumradius−to−shortest edge ratio), but grading remains good in practice.



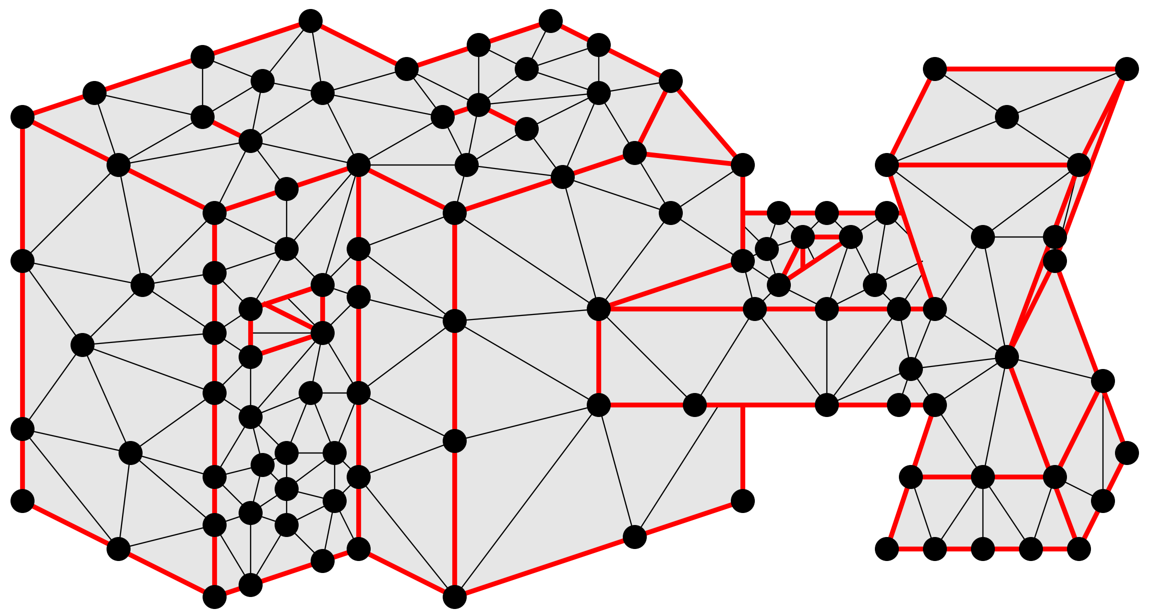


Input: A Piecewise Linear Complex



PLC

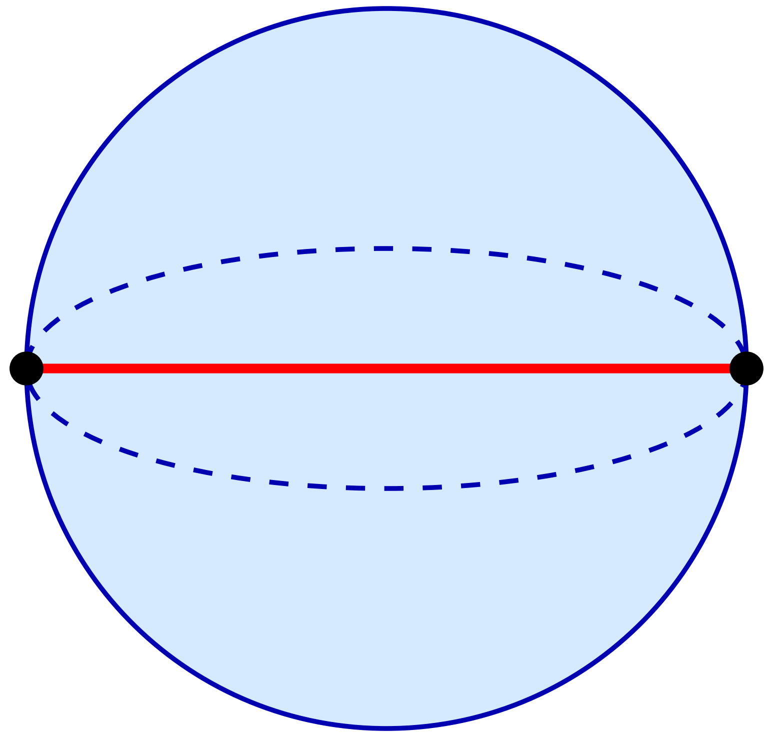
Set of vertices, segments, and *facets.*



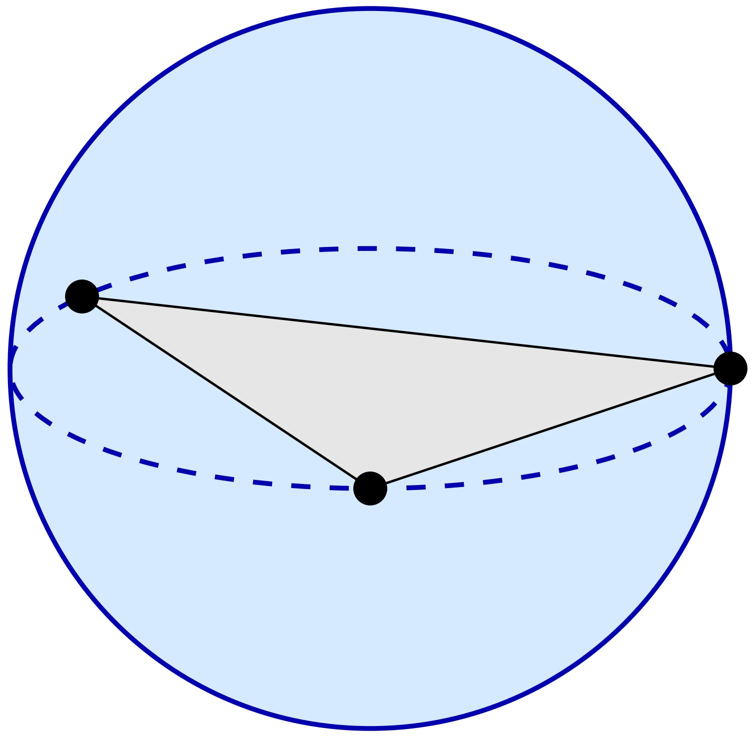
Mesh

The segments are divided into *subsegments,* and the facets into *subfacets.*

Definitions



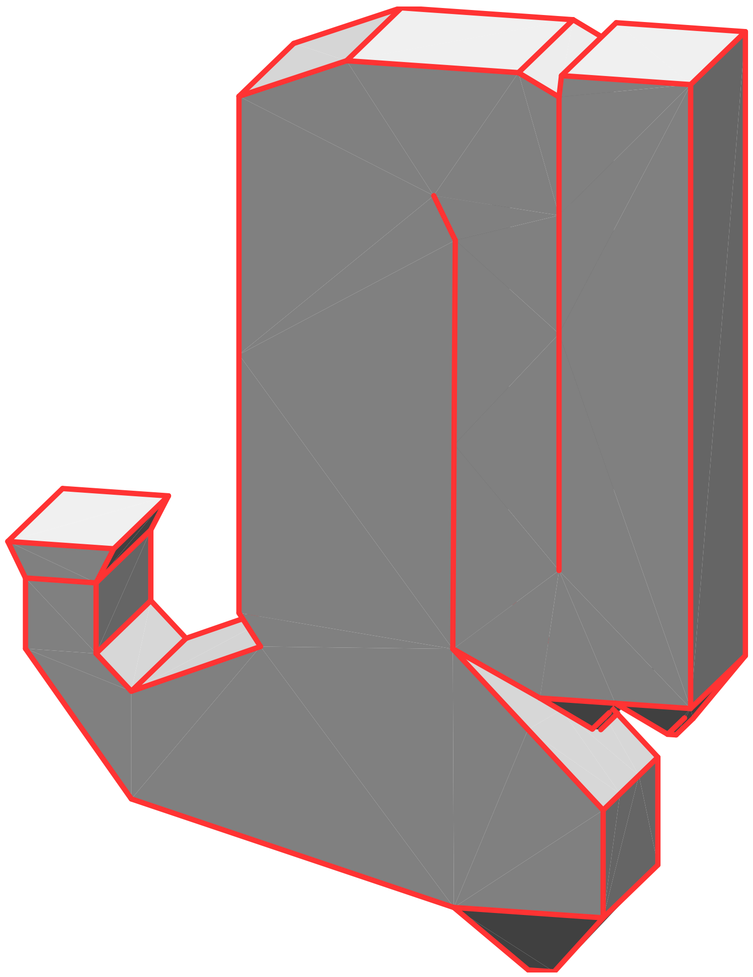
The *diametral sphere* of a subsegment

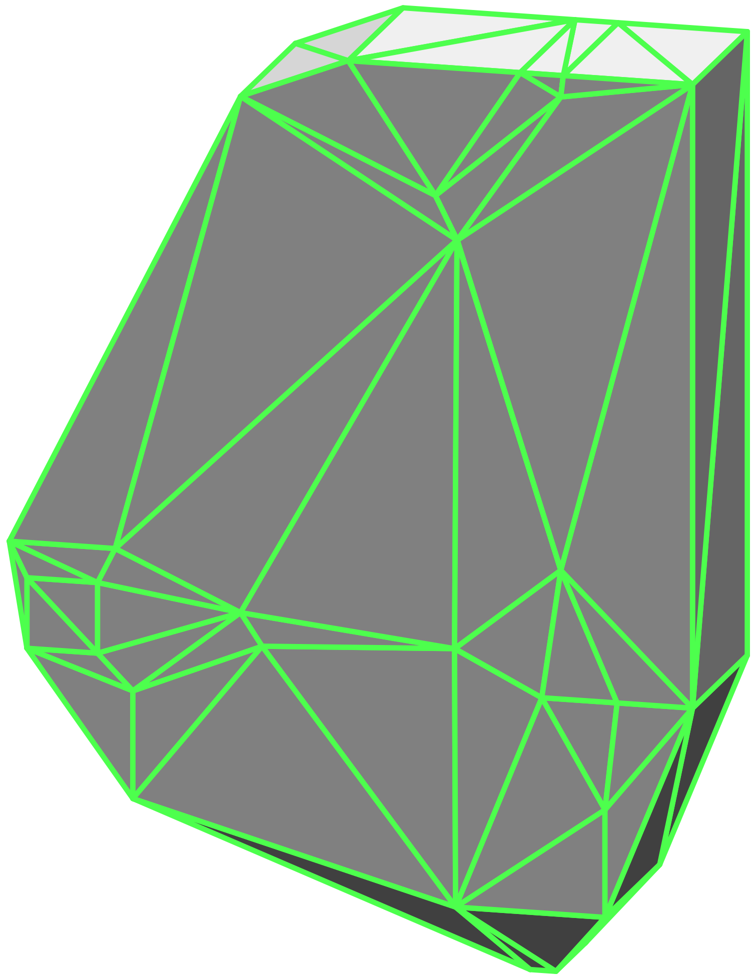


The *equatorial sphere* of a subfacet

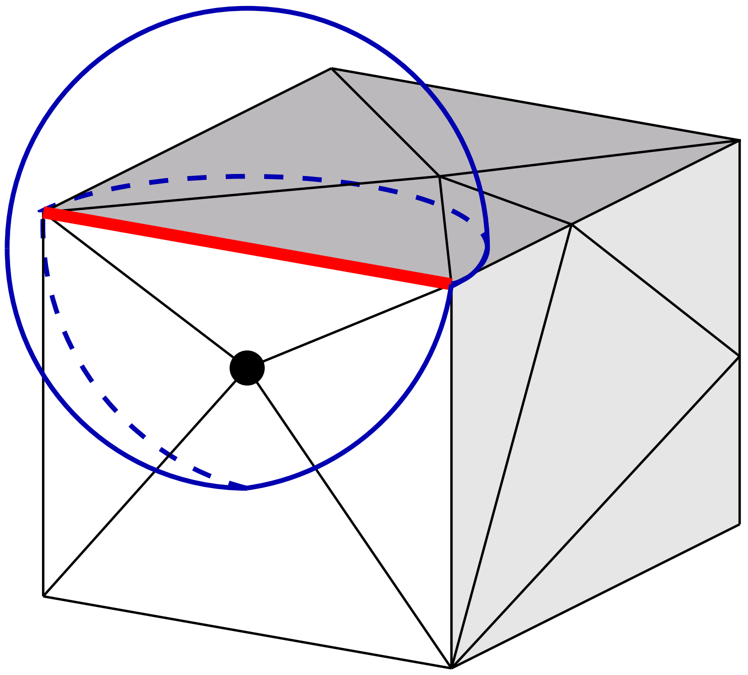
(The smallest sphere that passes through all its vertices.)

Begin with the Delaunay tetrahedralization of the vertices of the PLC.

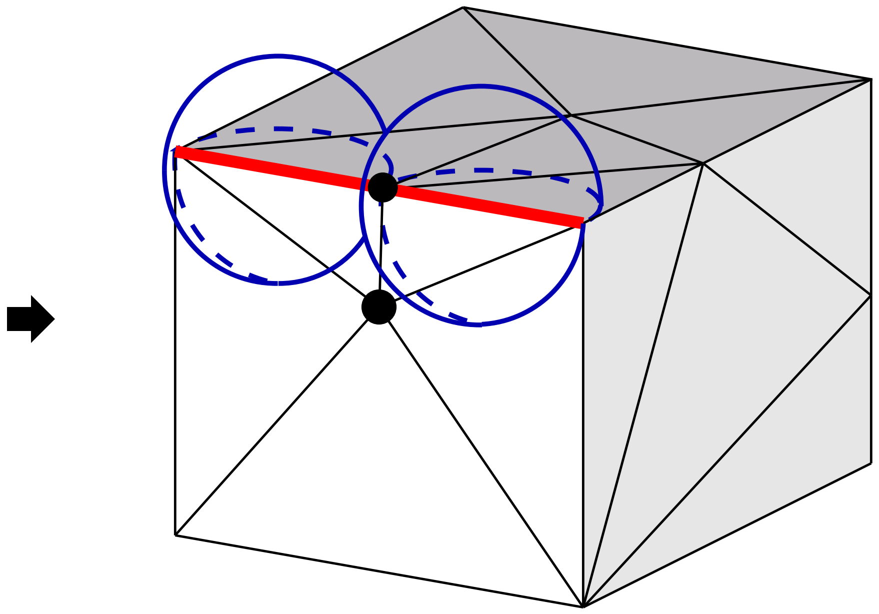

This PLC courtesy Carl−Ollivier Gooch.



Rule #1
Splitting an Encroached Subsegment

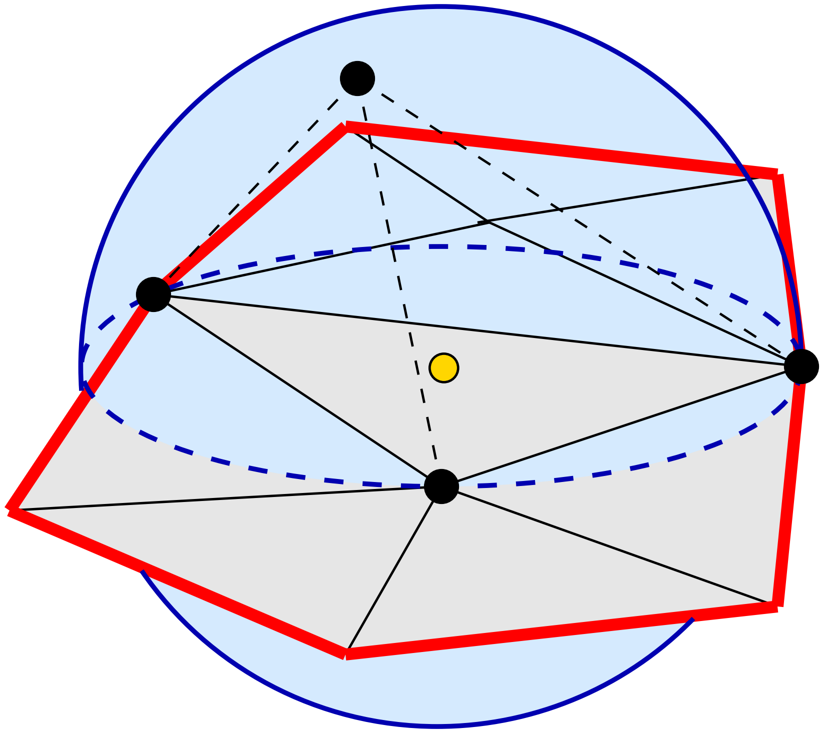


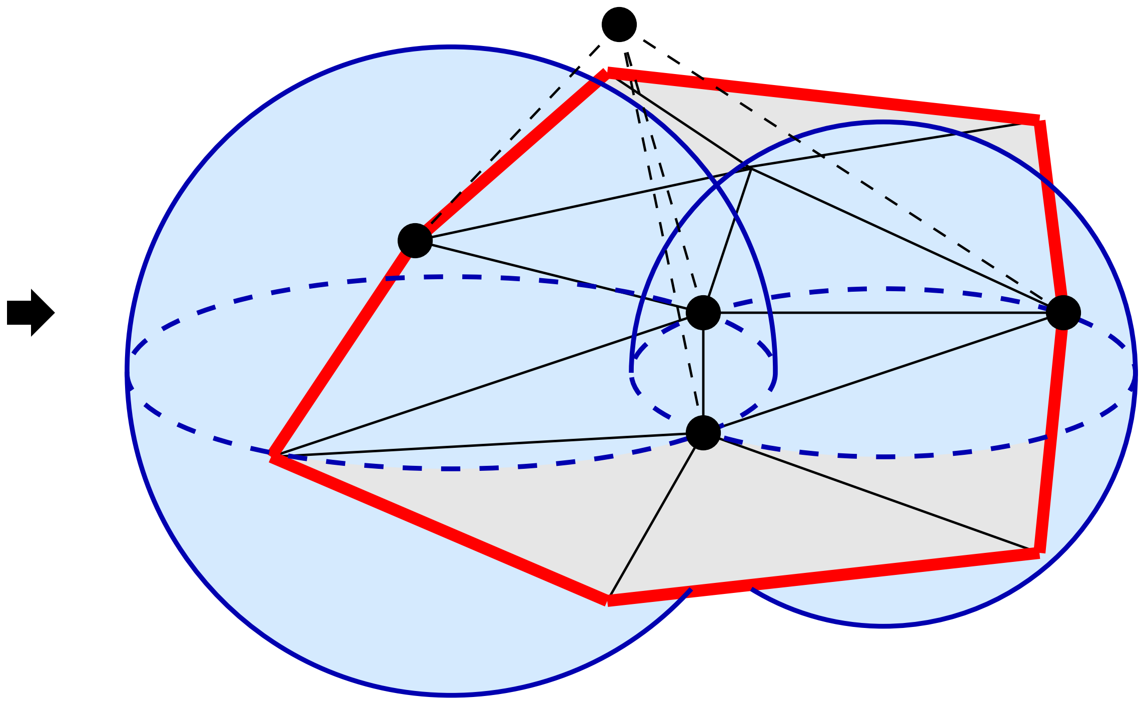
The diametral sphere of this subsegment is encroached.



Split the encroached subsegment by inserting a new vertex at its midpoint and maintaining the Delaunay property.

Rule #2
Splitting an Encroached Subfacet



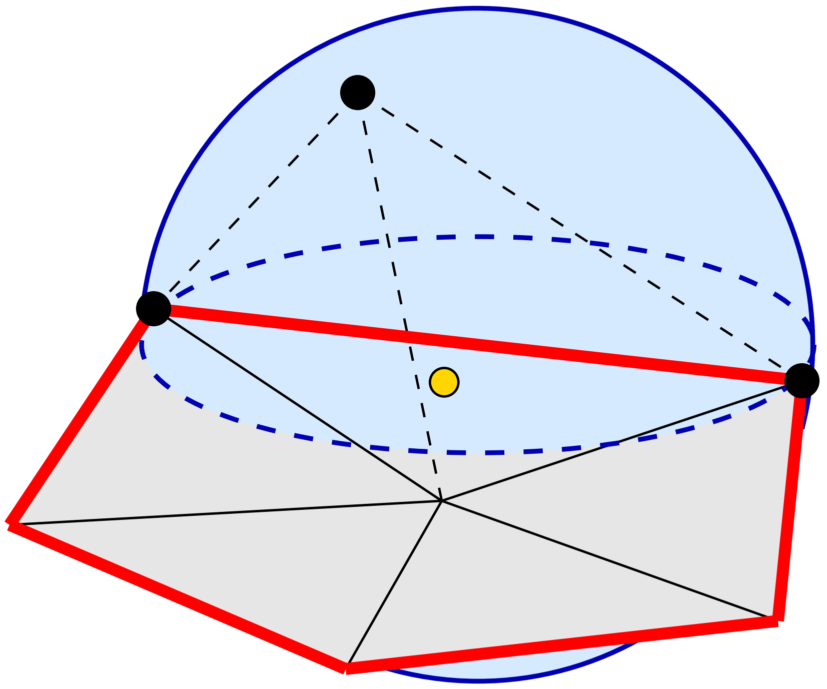


The equatorial sphere of this subfacet is encroached.

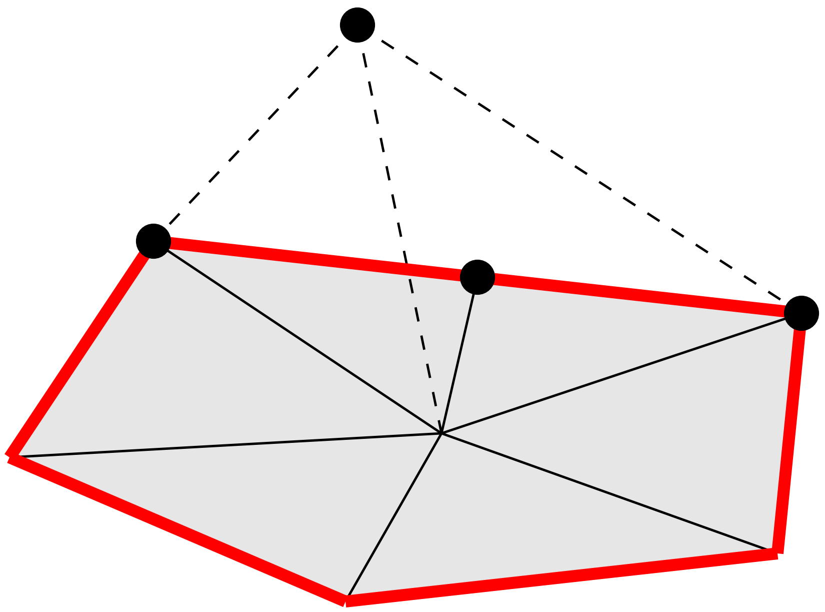
(For best results, you must choose the right subfacet to split first. Orthogonally project the encroaching vertex onto the facet. Split the subfacet that contains the projected point.)

Split the encroached subfacet by inserting a new vertex at its circumcenter and maintaining the Delaunay property.

But . . .



If the new vertex would encroach upon a subsegment, reject the vertex.



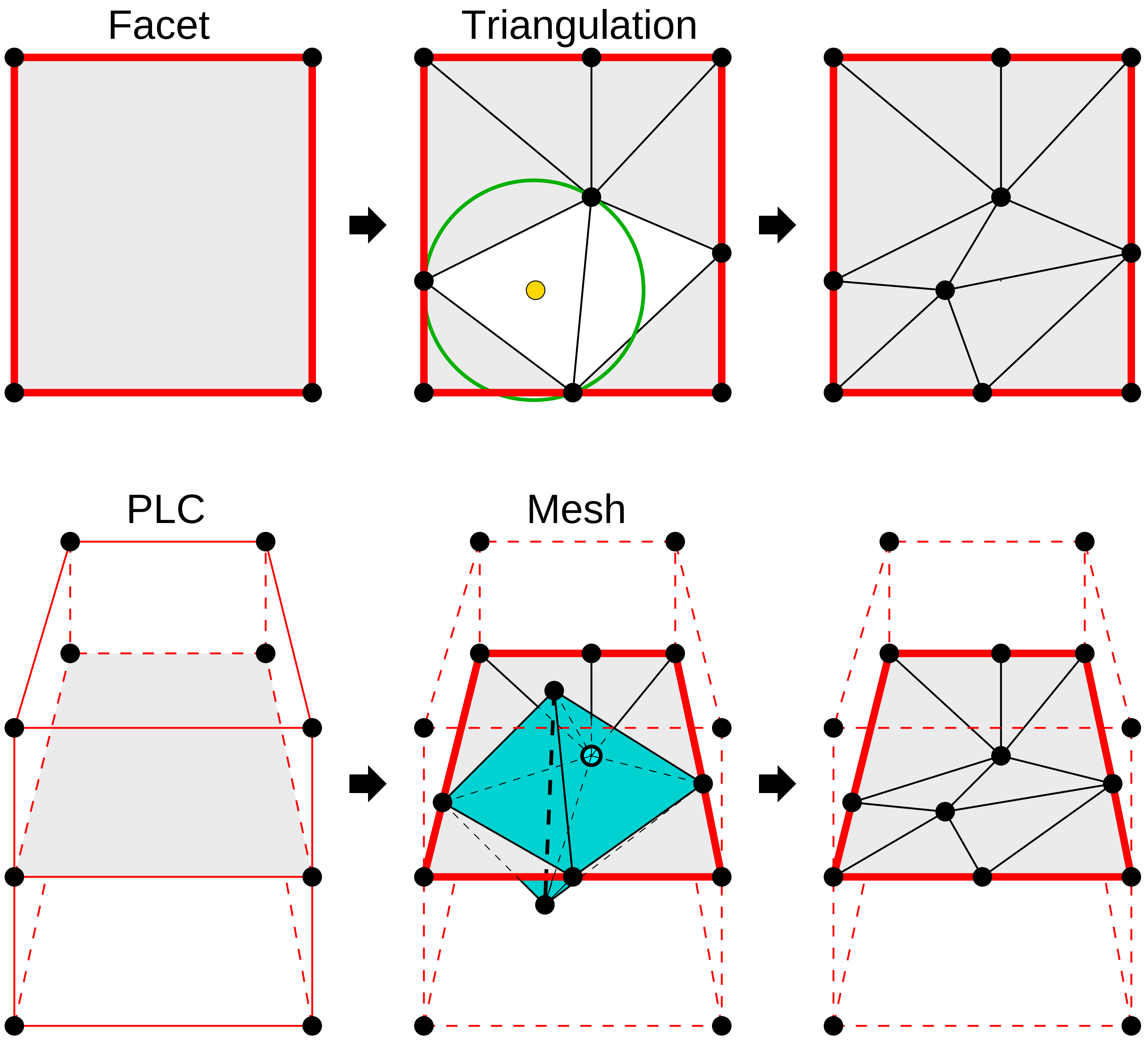
Split the encroached subsegment(s) instead.

Missing Facet Recovery by “Stitching” (Rule #2)

Maintain a 2D DT of each facet *separately* from the 3D mesh.

Split any subfacet that is present in a facet DT but not in the 3D mesh.

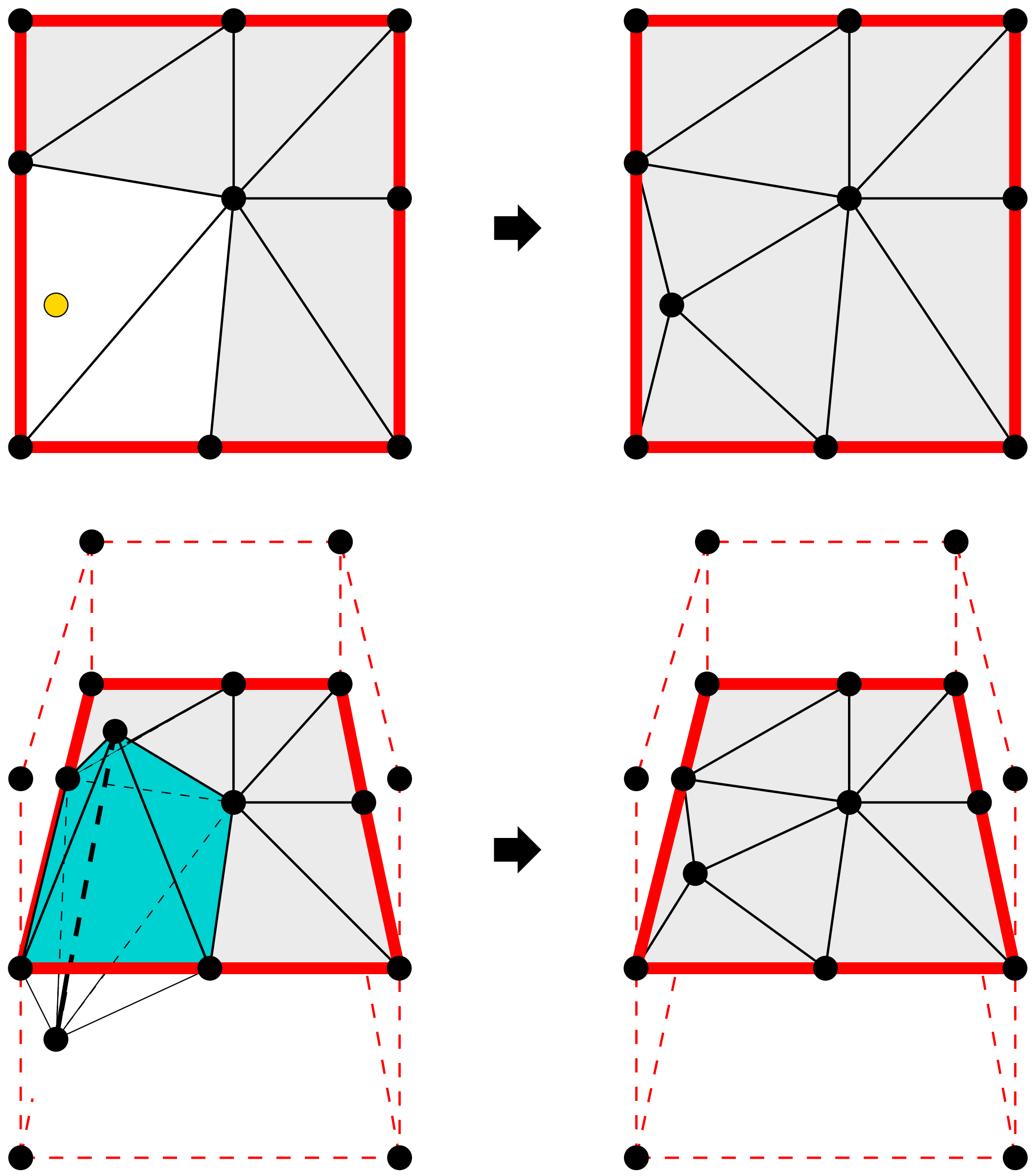
When you split a subfacet, insert new vertex into the 2D facet DT and the 3D mesh simultaneously.



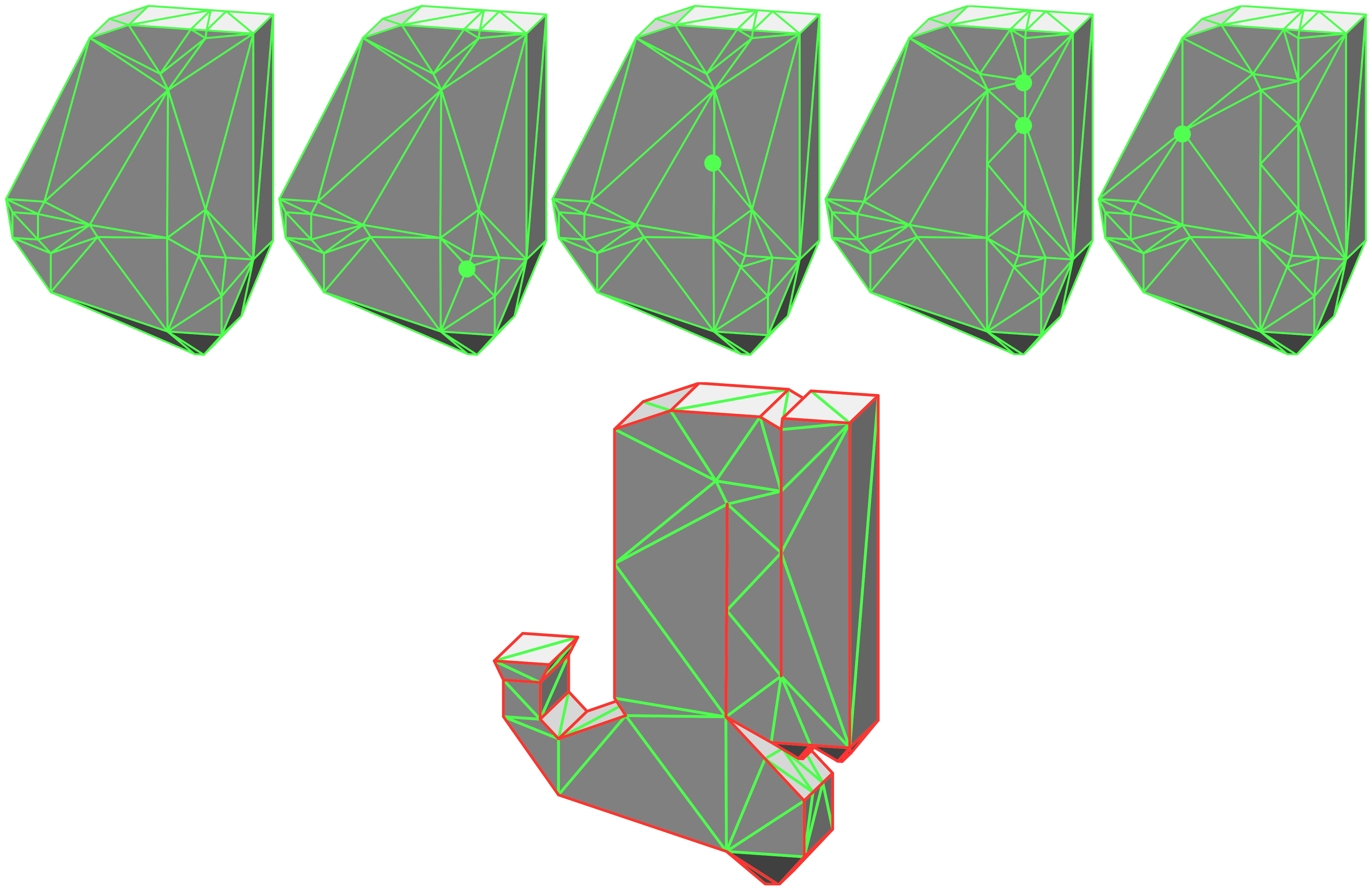
Missing Facet Recovery

By contrast, a popular method in the heuristic meshing literature inserts a vertex at the intersection of a missing facet and an edge of the 3D mesh.

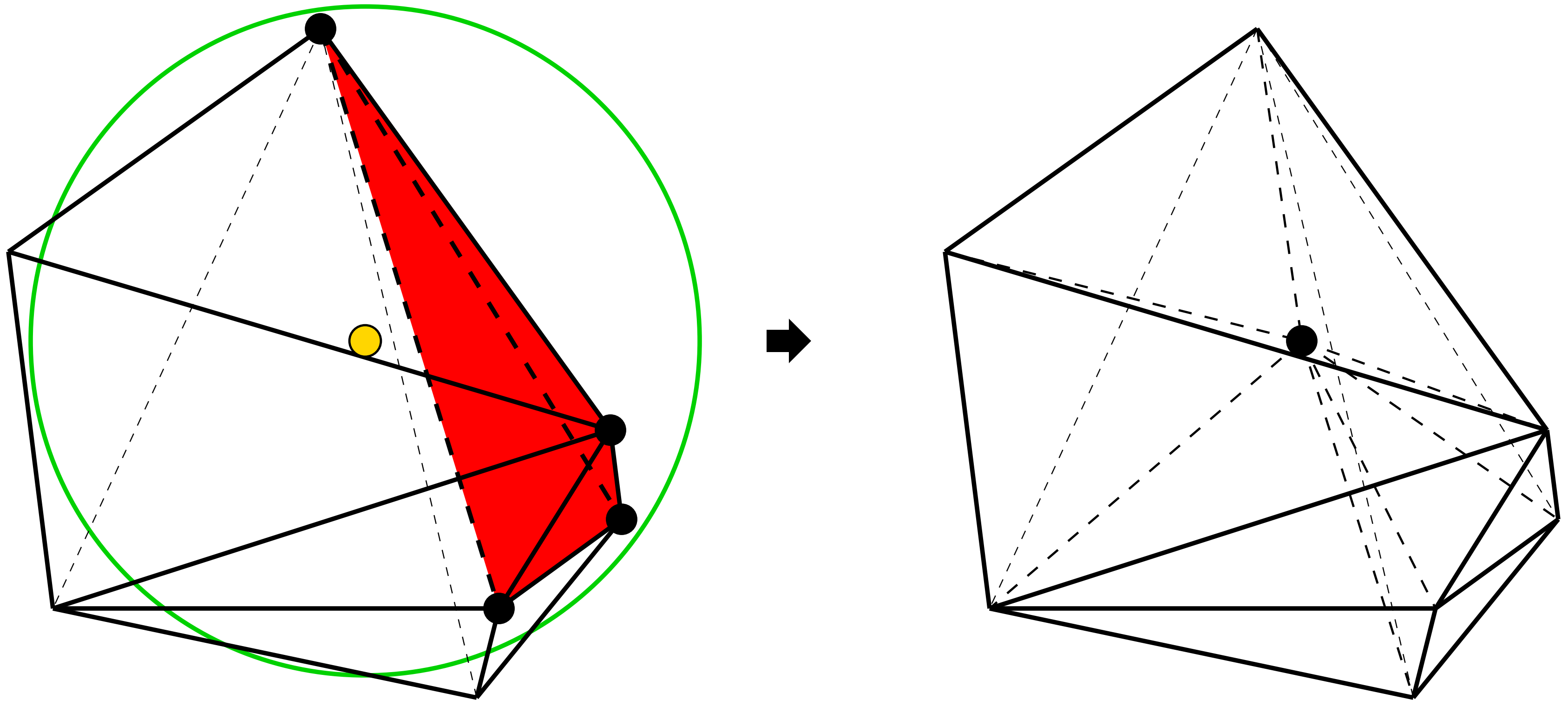
Unfortunately, this approach can place a vertex very close to a subsegment.



Delaunay refinement in action.

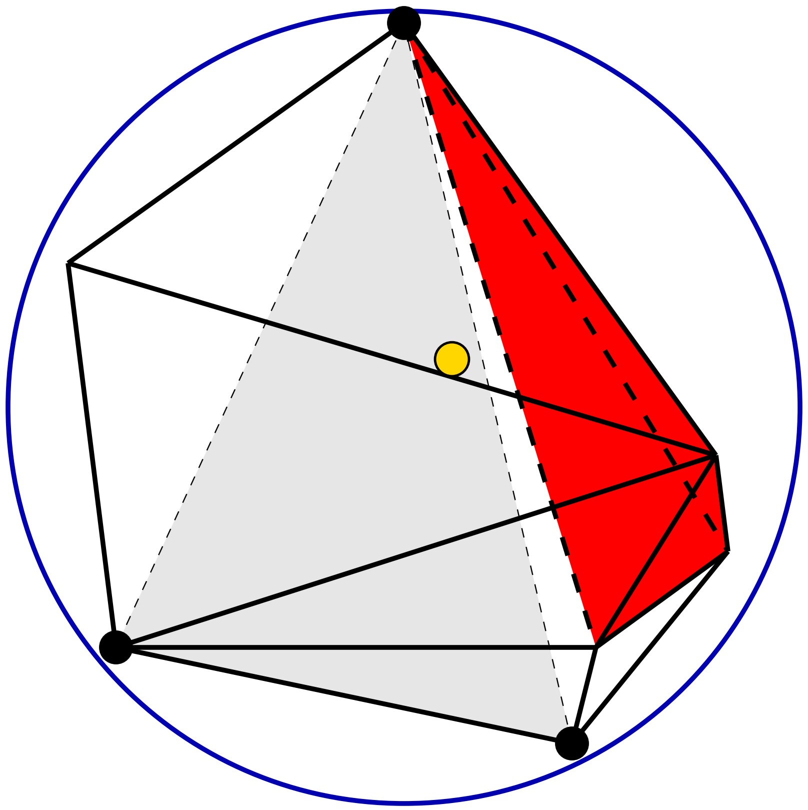


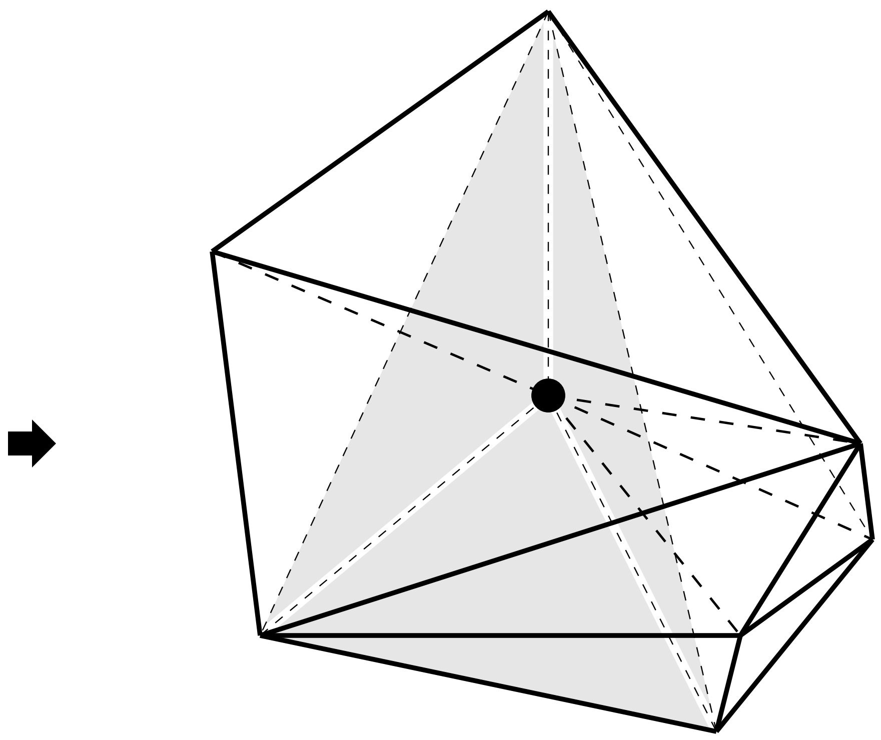
Rule #3
Splitting a Skinny Tetrahedron



Split a skinny tetrahedron by inserting a new vertex at its circumcenter and maintaining the Delaunay property.

But . . .



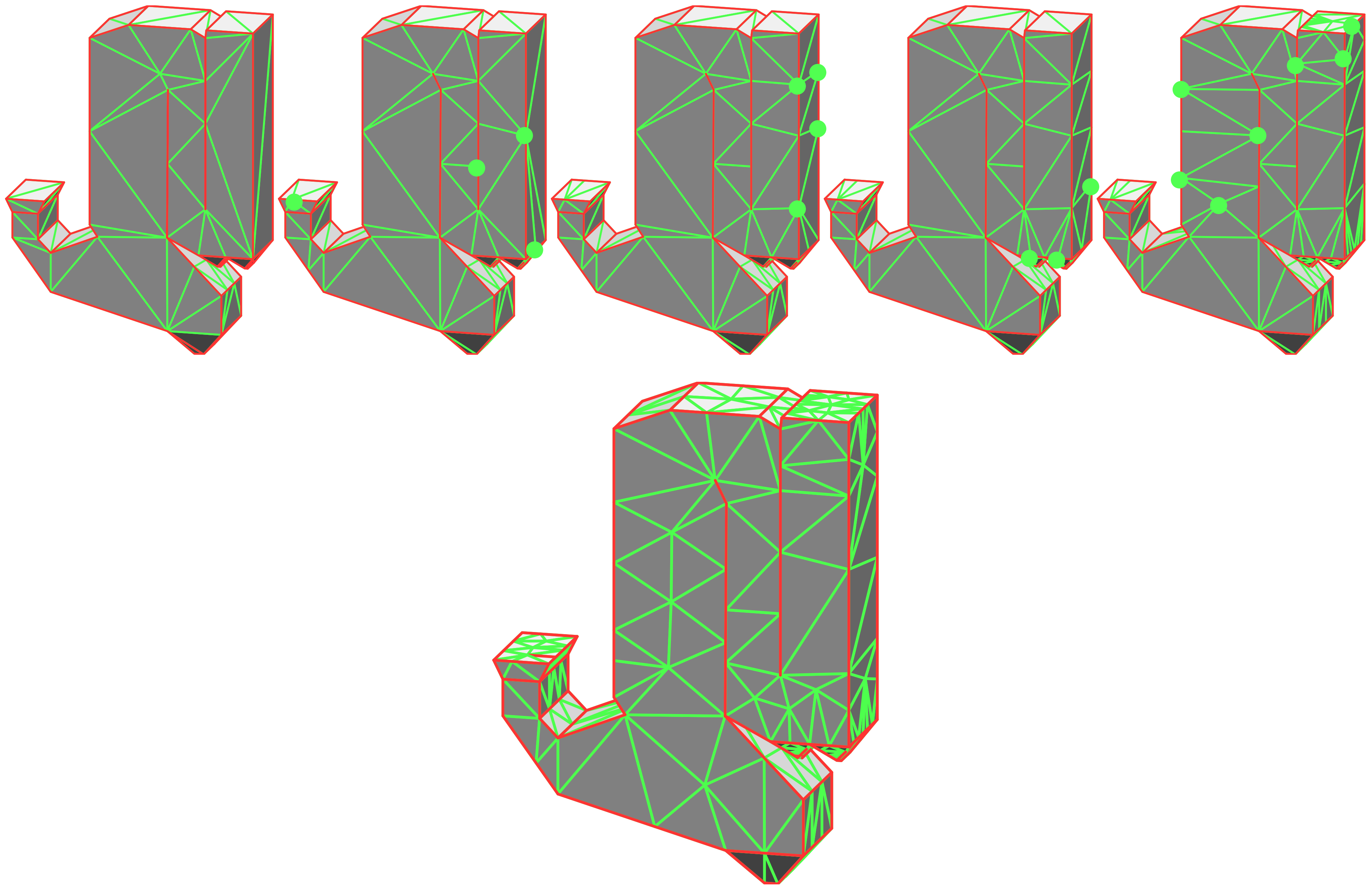


If the new vertex would encroach upon a subfacet or subsegment, reject the vertex.

Split the encroached subfacet(s) or subsegment(s) instead.

(Subsegments first. Split encroached subfacets only if the skinny tetrahedron survives after you split all subsegments that its circumcenter encroaches upon.)

Delaunay refinement in action.

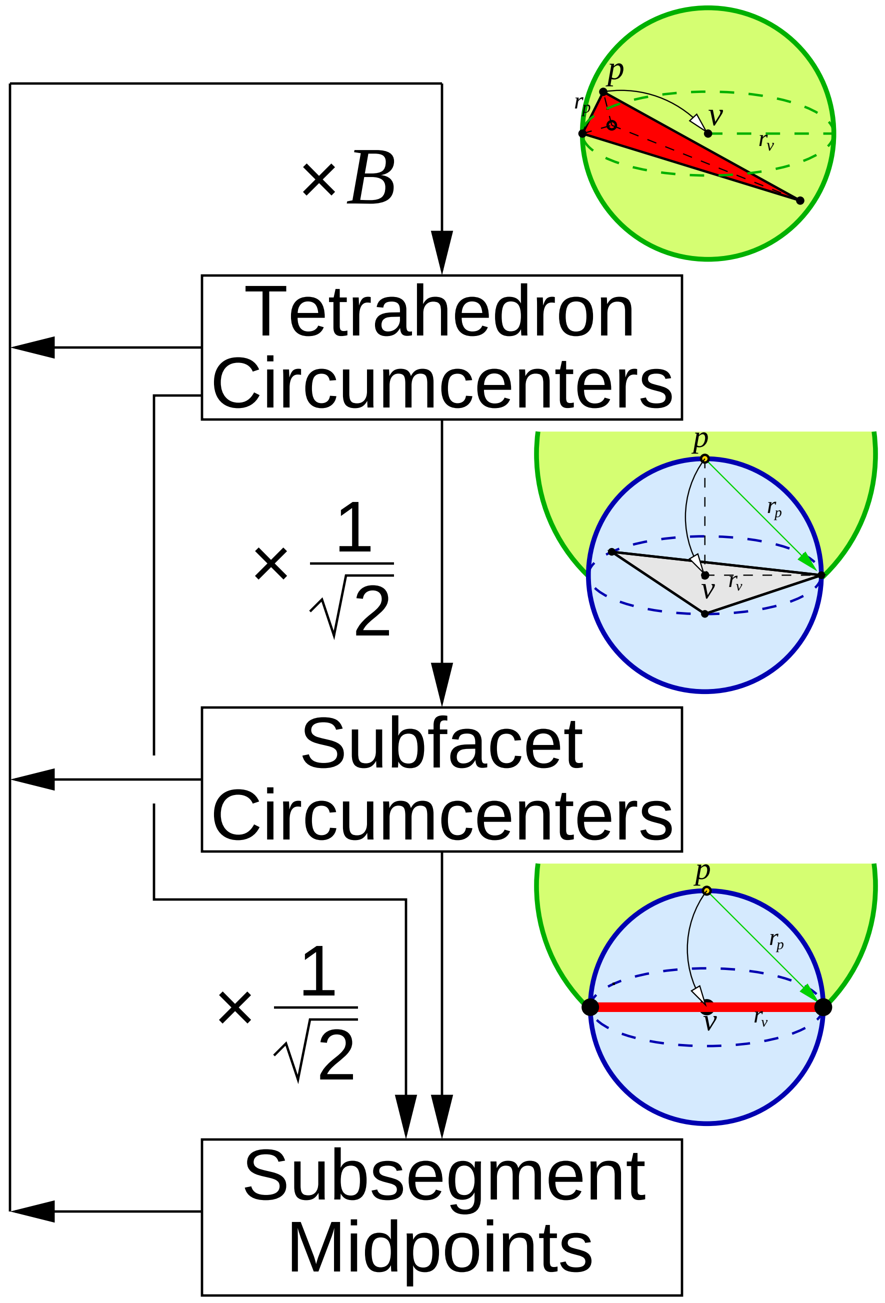


Goal: Avoid Cycle of Diminishing Edge Lengths

Multipliers (right) reflect smallest possible insertion radius of new vertex, relative to vertex that “caused” it.

Algorithm is guaranteed to terminate if no cycle exists with product less than 1.

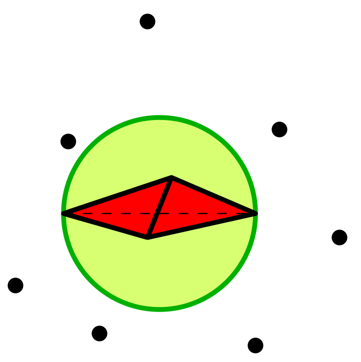
We require $B \geq 2$.


$\times B$
Tetrahedron Circumcenters
$\times \frac{1}{\sqrt{2}}$
Subfacet Circumcenters
$\times \frac{1}{\sqrt{2}}$
Subsegment Midpoints
p
v
r_p
r_v

Slivers

Sliver Elimination

The theoretical bound allows slivers to survive if their circumradius−to−shortest edge ratios are less than 2. How do we get rid of them?



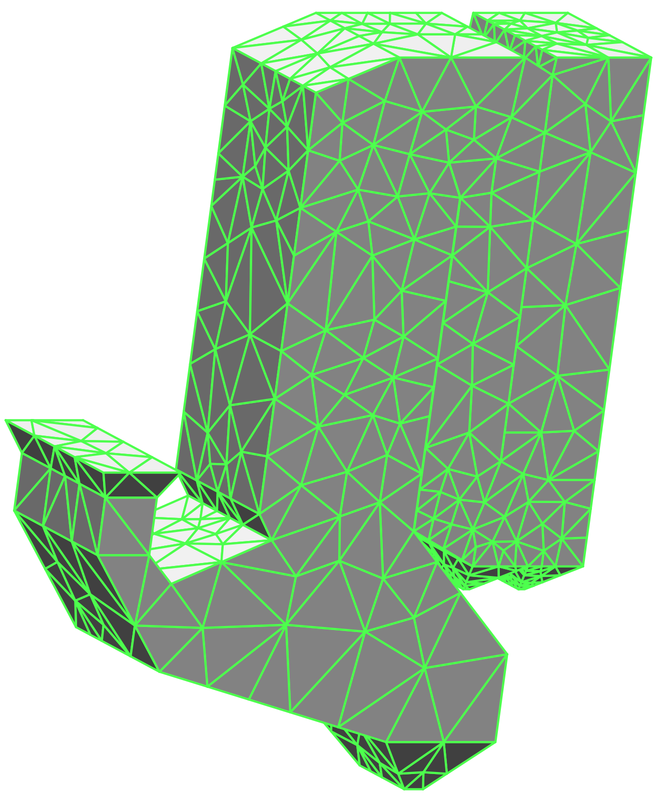
- Delaunay refinement. A sliver can always be eliminated by a vertex at its circumcenter. There’s just no guarantee that refinement will terminate.

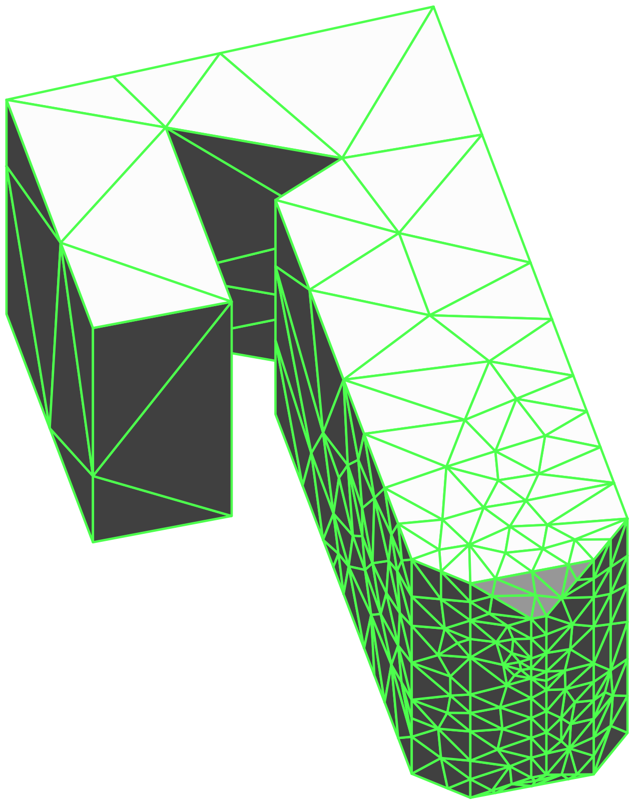
- Randomized Delaunay refinement (Chew). Insert a random off−center vertex. If you don’t like the result, undo and try again with different random vertex. Has a “guarantee.”

- “Sliver Exudation” (Cheng et al.). Has a “guarantee.”

Slivers and Delaunay Refinement

Fortunately, sliver removal by Delaunay refinement works well in practice, even without a termination guarantee.





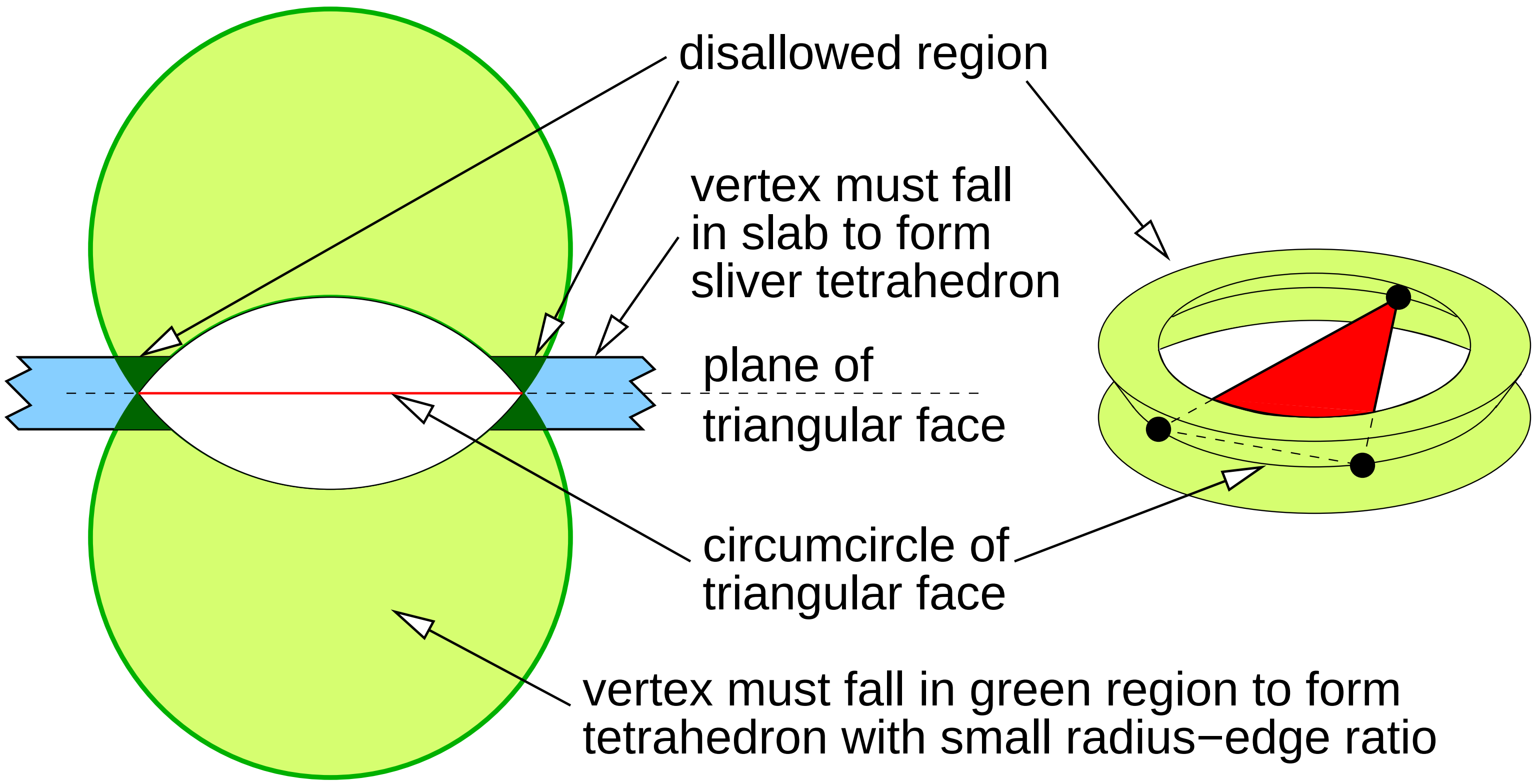
minimum dihedral: 22°   minimum dihedral: 22.8°

Jonathan Richard Shewchuk, ‘‘Tetrahedral Mesh Generation by Delaunay Refinement,’’ Proceedings of the Fourteenth Annual Symposium on Computational Geometry, pages 86−95, June 1998.

Chew's Third Delaunay Refinement Algorithm

L. Paul Chew, ''Guaranteed−Quality Delaunay Meshing in 3D,'' Proceedings of the Thirteenth Annual Symposium on Computational Geometry, pages 391−393, June 1997.

Chew observes that a new vertex must fall in a small region to create a sliver with good circumradius−to−shortest edge ratio (with a pre−existing triangular face, like the red one below).

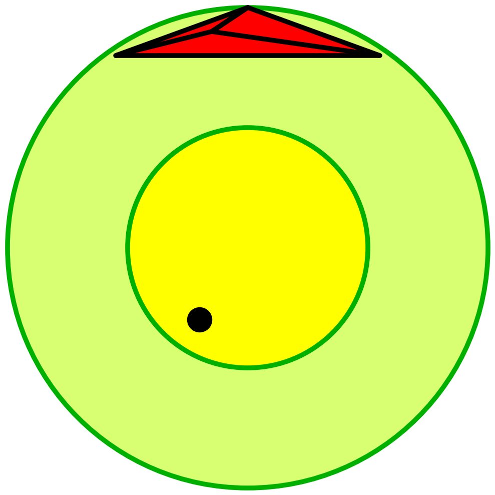

disallowed region
vertex must fall in slab to form sliver tetrahedron
plane of triangular face
circumcircle of triangular face
vertex must fall in green region to form tetrahedron with small radius−edge ratio

Idea: If a circumcenter falls in a face's disallowed region, perturb it randomly and try again. (Never implemented, to my knowledge.)

Chew’s Third Delaunay Refinement Algorithm

L. Paul Chew, ‘‘Guaranteed−Quality Delaunay Meshing in 3D,’’ Proceedings of the Thirteenth Annual Symposium on Computational Geometry, pages 391−393, June 1997.

New vertex may go anywhere in the inner half of the circumsphere of a skinny tetrahedron.



If ‘‘sliver’’ is defined as having an extremely small dihedral angle, Chew can prove that the union of the forbidden regions does not fill the inner sphere. A random search eventually finds a good spot.

Unfortunately, the bound on dihedral angle is too minuscule to bother computing. Still, the first provably good sliver eliminator!

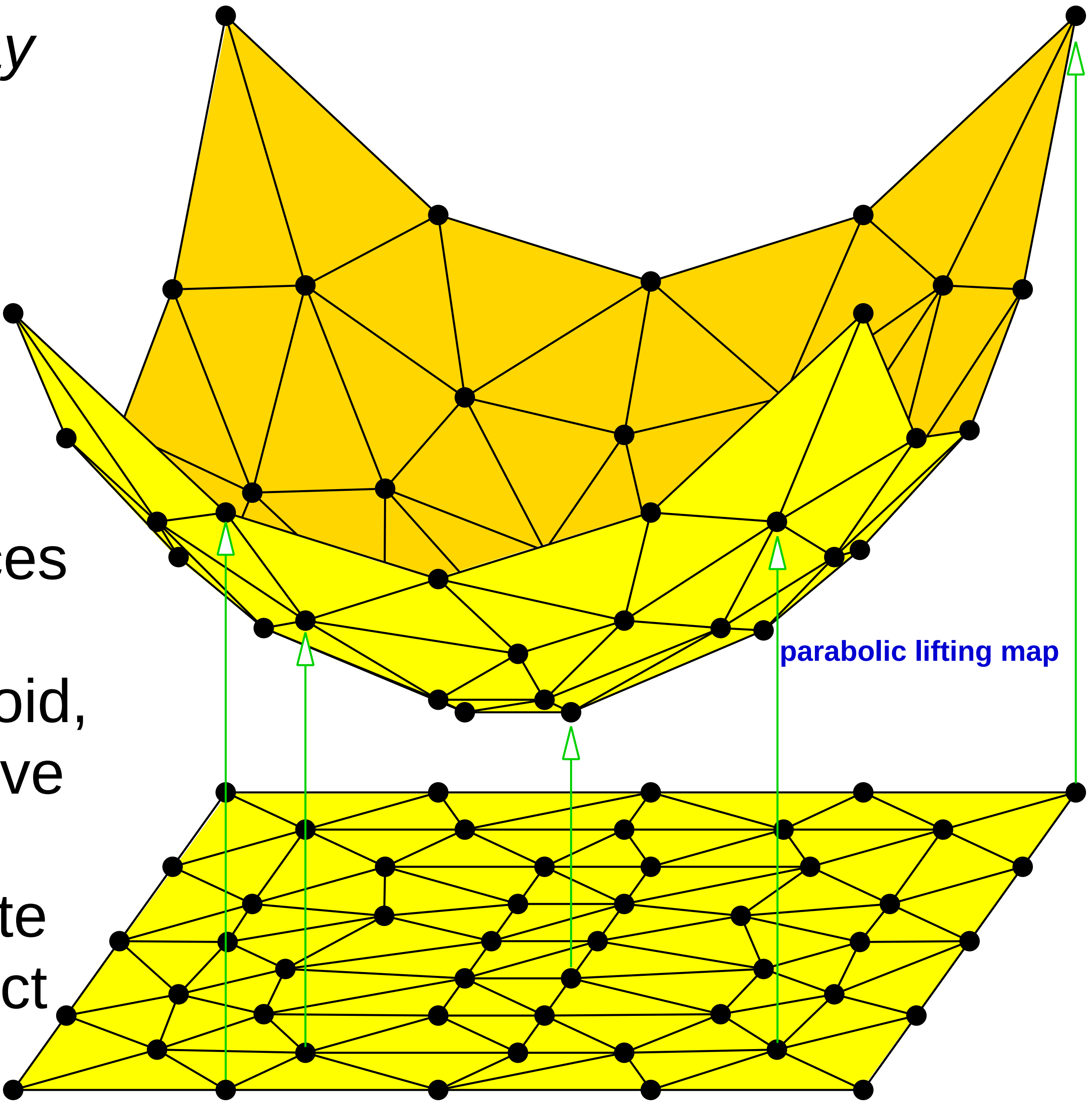
Sliver Exudation

Siu−Weng Cheng, Tamal Krishna Dey, Herbert Edelsbrunner, Michael A. Facello, and Shang−Hua Teng, ‘‘Sliver Exudation,’’ Journal of the ACM **47** (5):883−904, September 2000.

Uses *weighted Delaunay triangulations.*

The 3D DT matches the lower convex hull of the vertices lifted onto a paraboloid in E^4.

In a weighted DT, vertices with positive weight are lifted below the paraboloid, and vertices with negative weight are lifted above the paraboloid. Compute lower convex hull; project tetrahedra down to E^3.

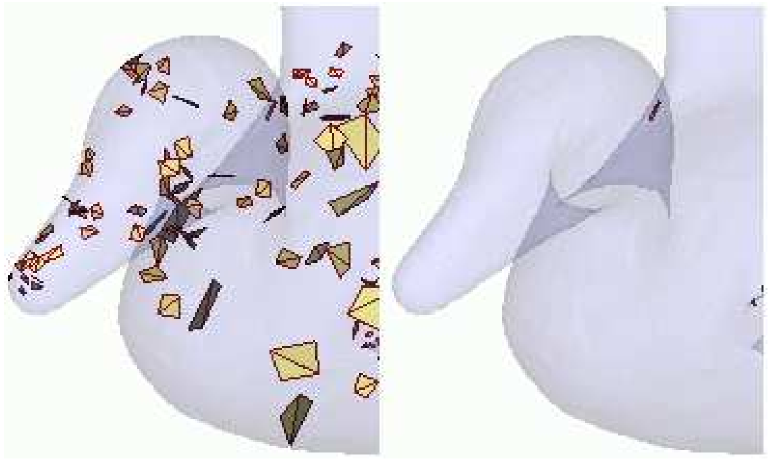


Sliver Exudation

Siu−Weng Cheng, Tamal Krishna Day, Herbert Edelsbrunner, Michael A. Facello, and Shang−Hua Teng, ‘‘Sliver Exudation,’’ Journal of the ACM **47** (5):883−904, September 2000.

As Chew shows, slivers with good circumradius−to−shortest edge ratios are fragile: small perturbations eliminate them.

Idea: fiddle with the *weights* of the vertices until the slivers disappear. Weights must stay within a small range, lest a vertex disappear into the convex hull. Search within that range for a sliver−free configuration.

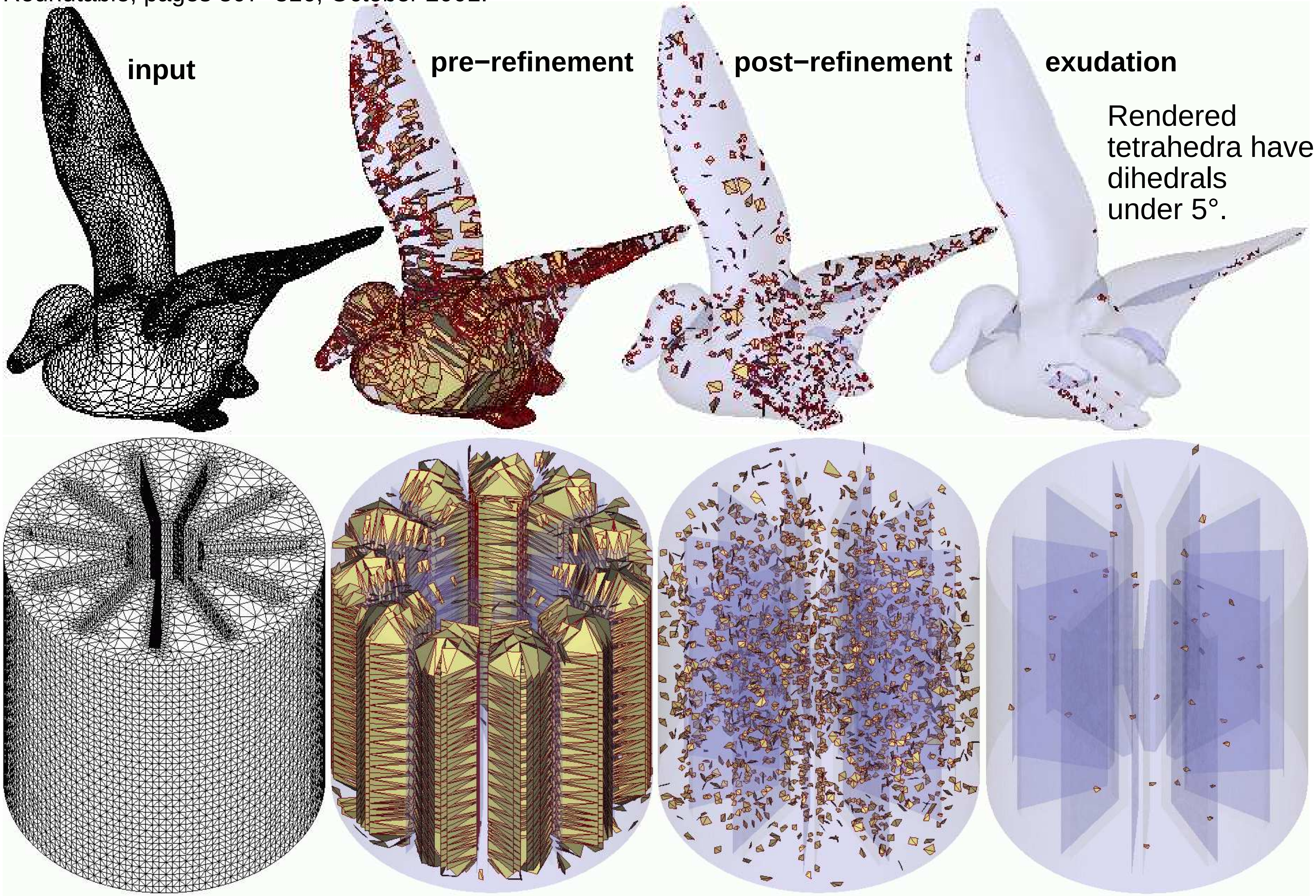


Courtesy
Damrong Guoy

Provably good sliver elimination. Unfortunately, the bound on dihedral angle is too minuscule to bother computing.

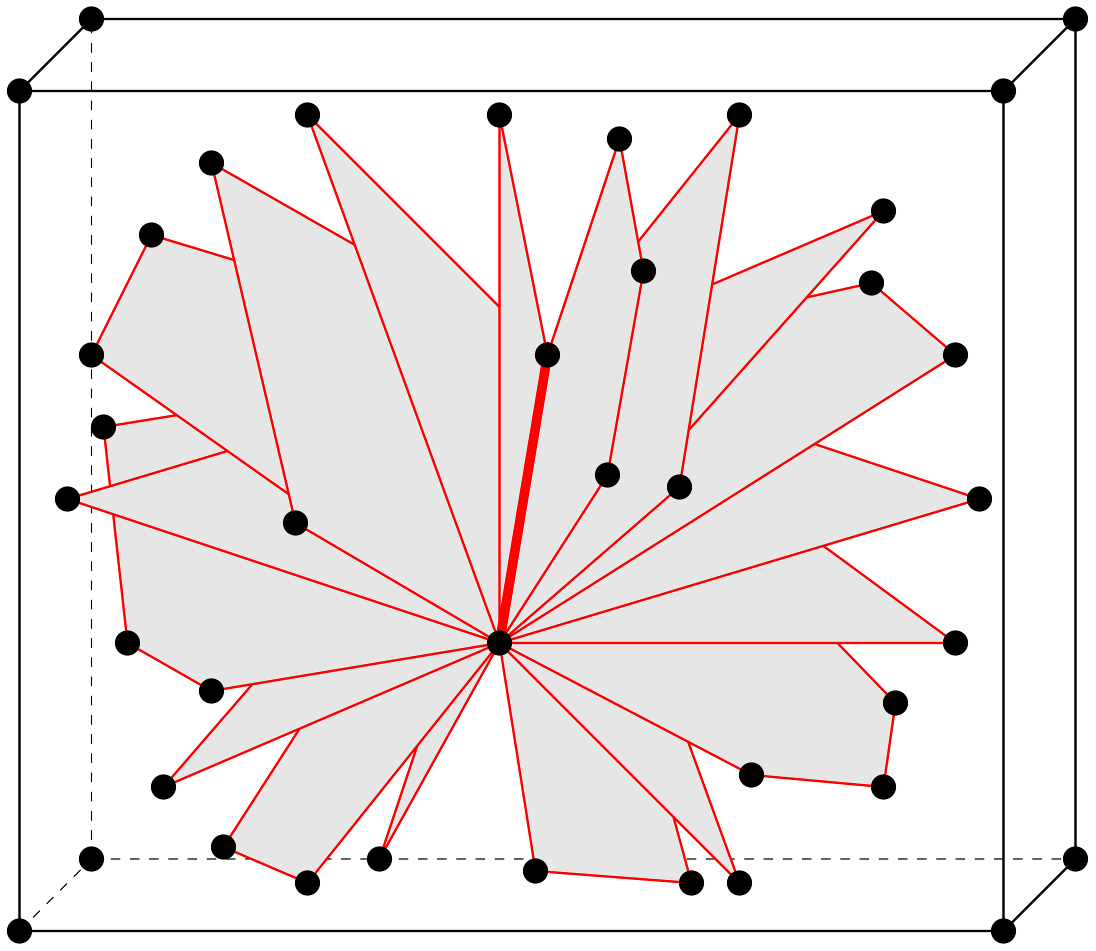
Experiments show sliver exudation to be reasonably (but not perfectly) effective.

Herbert Edelsbrunner and Damrong Guoy, ‘‘An Experimental Study of Sliver Exudation,’’ Tenth International Meshing Roundtable, pages 307−316, October 2001.



Small Angles

A Hard Example for Tetrahedral Meshing



It is difficult to mesh the interior of this box with Delaunay tetrahedra. A new vertex inserted in one facet tends to knock out triangular subfacets in adjacent facets.

Provably Good Meshing for 3D Domains with Small Angles

Earliest algorithm: Mine...but the paper omits the proof. Uses 3D CDTs. Handles non−manifold boundaries.

Jonathan Richard Shewchuk, ‘‘Mesh Generation for Domains with Small Angles,’’ Proceedings of the Sixteenth Annual Symposium on Computational Geometry, pages 1−10, June 2000.

Cheng−Dey−Ramos−Ray. Uses DTs, but handles only manifold boundaries (polyhedra with holes).

Siu−Weng Cheng, Tamal Krishna Dey, Edgar A. Ramos, and Tathagata Ray, ‘‘Quality Meshing for Polyhedra with Small Angles,’’ Proceedings of the Twentieth Annual Symposium on Computational Geometry, June 2004.

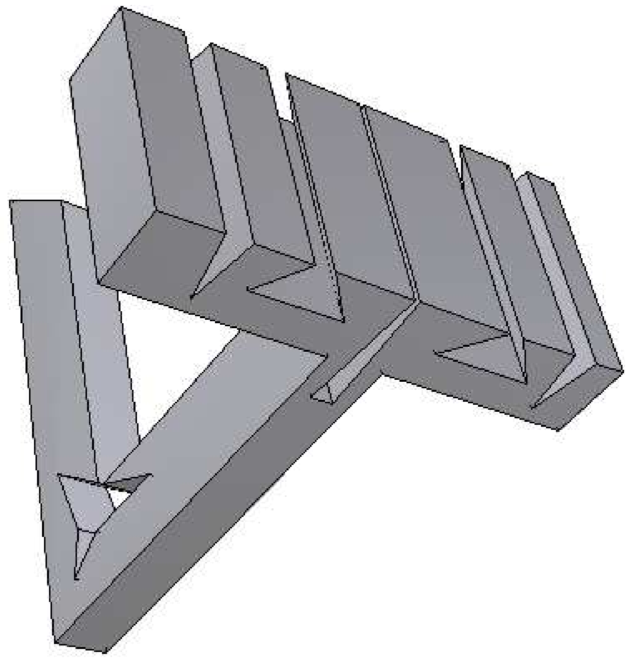
Pav−Walkington. Uses DTs, handles non−manifold boundaries. Claims provably good grading!

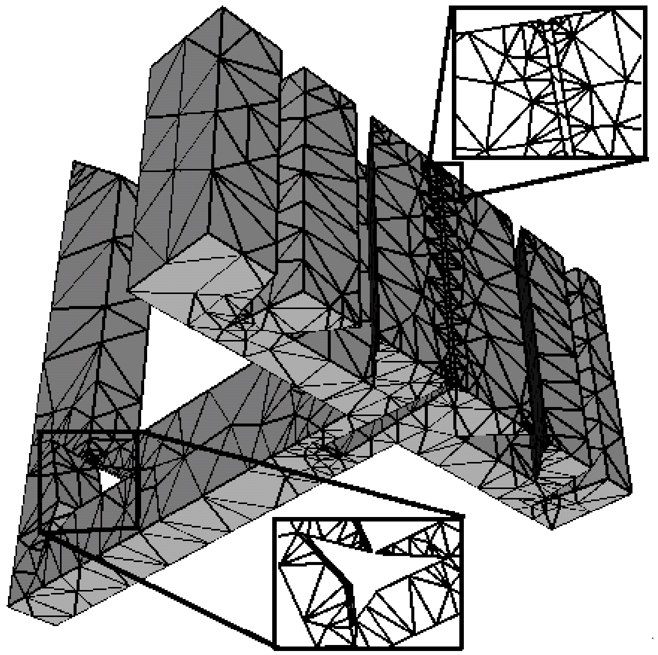
Steven Pav and Noel Walkington, ‘‘Robust Three Dimensional Delaunay Refinement,’’ Thirteenth International Meshing Roundtable, September 2004.

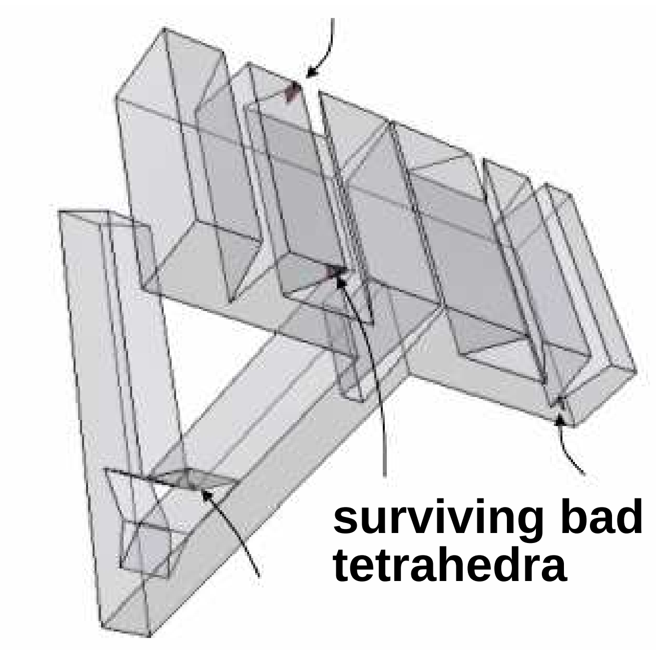
All guarantee good circumradius−to−shortest edge ratios except near small input angles.

3D Small Angles: Cheng−Dey−Ramos−Ray

Siu−Weng Cheng, Tamal Krishna Dey, Edgar A. Ramos, and Tathagata Ray, ‘‘Quality Meshing for Polyhedra with Small Angles,’’ Proceedings of the Twentieth Annual Symposium on Computational Geometry, June 2004.







Courtesy Tamal Dey

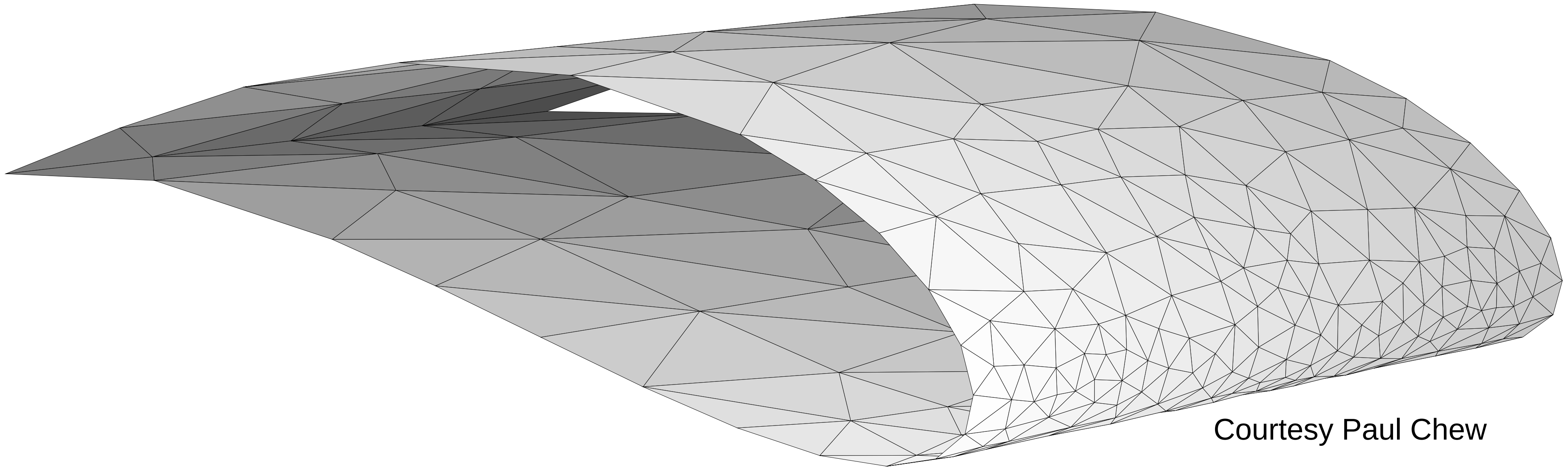
The only one of these algorithms implemented so far.

Conclusions

Things I Don’t Have Time to Discuss

Provably good triangular mesh generation for curved surfaces. (Chew’s second Delaunay refinement algorithm was designed for this purpose.)

L. Paul Chew, ‘‘Guaranteed−Quality Mesh Generation for Curved Surfaces,’’ Proceedings of the Ninth Annual Symposium on Computational Geometry, pages 274−280, May 1993.



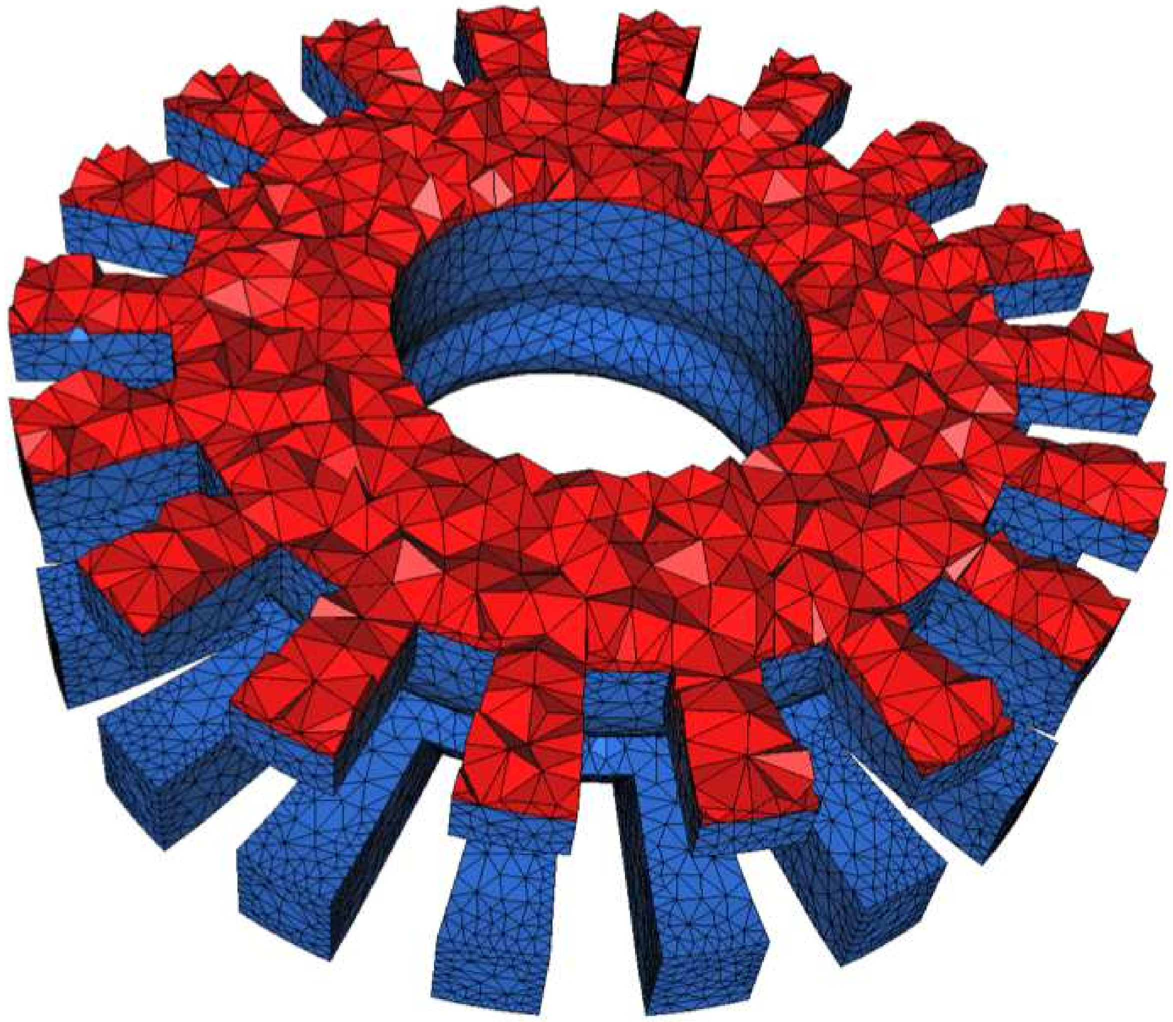
Courtesy Paul Chew

This work is continued by Jean−Daniel Boissonnat and Steve Oudot. ‘‘Provably Good Surface Sampling and Approximation,’’ Symposium on Geometry Processing 2003, pages 9−19, June 2003.

Things I Don’t Have Time to Discuss

Provably good tetrahedral mesh generation for domains with curved boundaries.

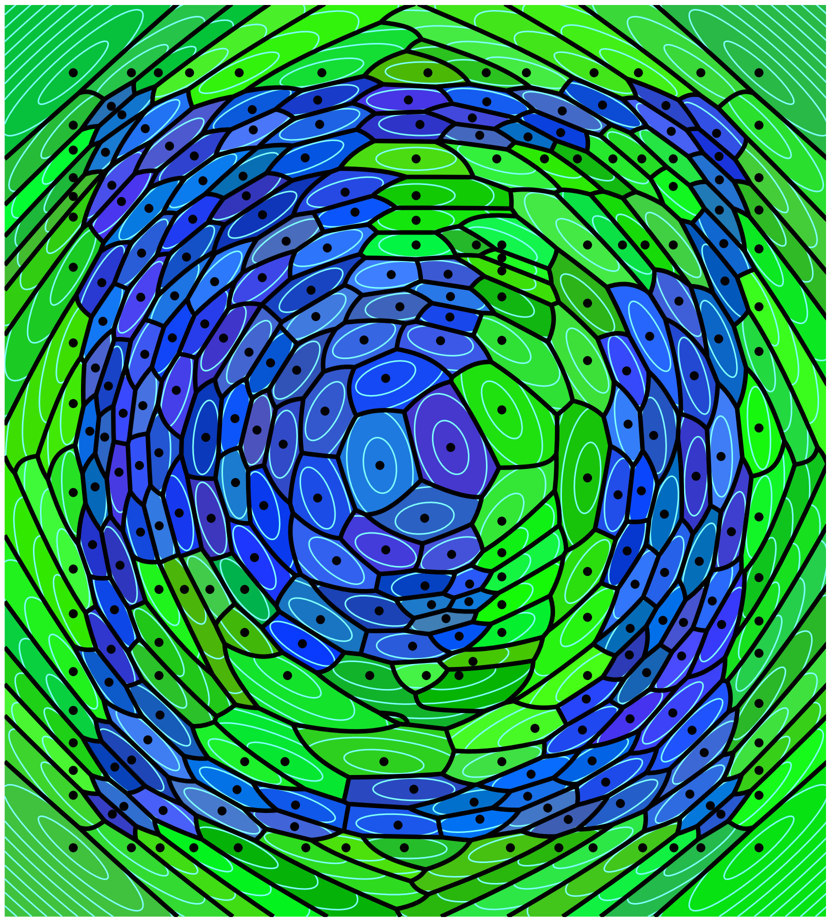
Laurent Rineau and Mariette Yvinec, ‘‘Meshing 3D Domains Bounded by Piecewise Smooth Surfaces,’’ Sixteenth International Meshing Roundtable, pages 443−460, October 2007.

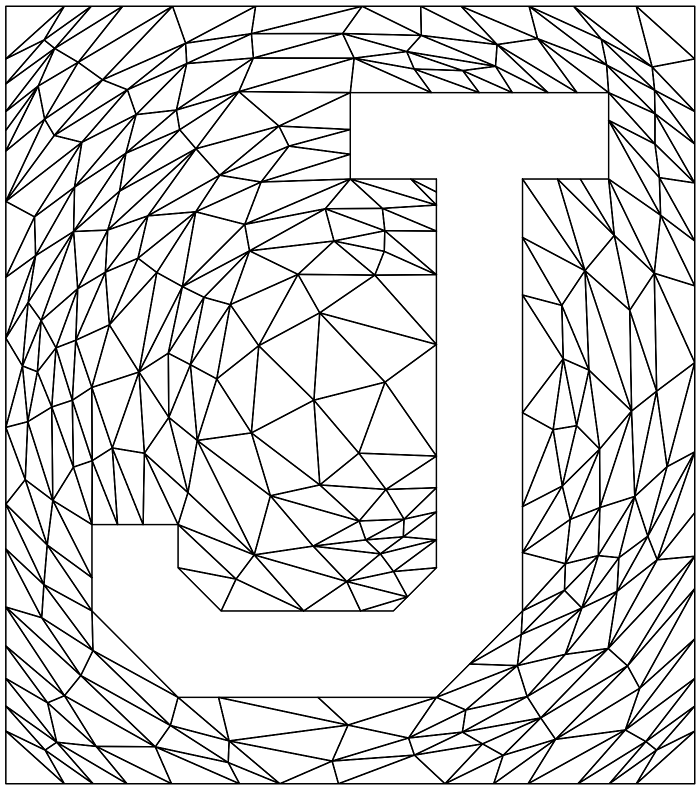


Things I Don’t Have Time to Discuss

Provably good anisotropic mesh generation (using anisotropic Voronoi diagrams).

François Labelle and Jonathan Richard Shewchuk, ‘‘Anisotropic Voronoi Diagrams and Guaranteed−Quality Anisotropic Mesh Generation,’’ Proceedings of the Nineteenth Annual Symposium on Computational Geometry, pages 191−200, June 2003.

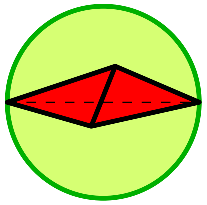




Courtesy François Labelle

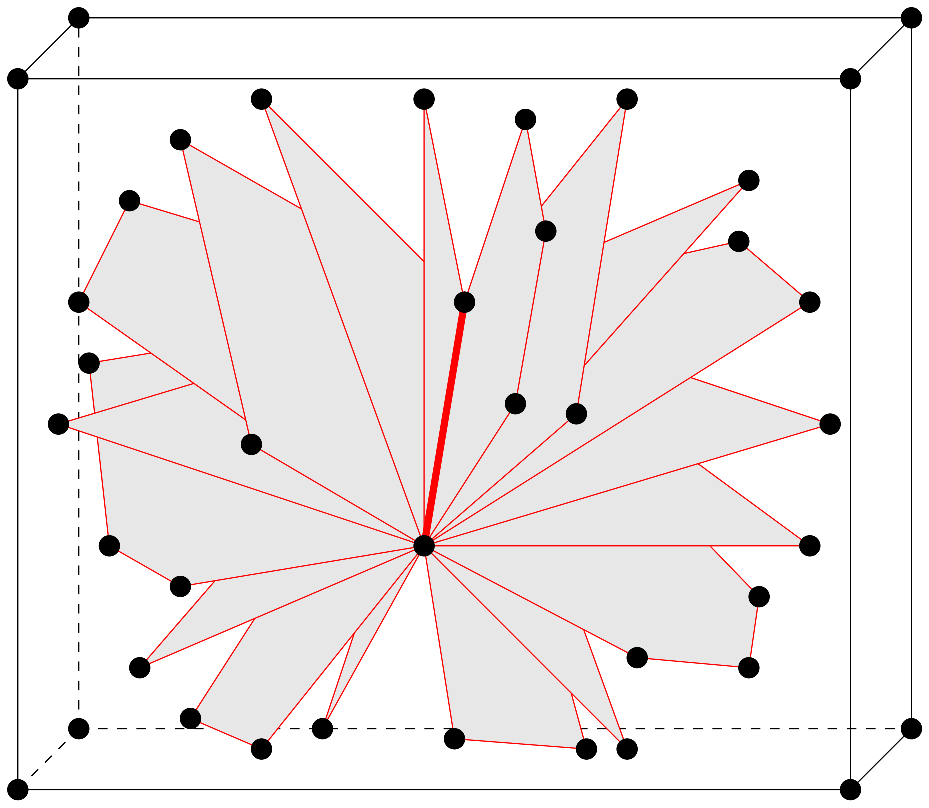
Open Problems

- Tetrahedral meshing of PLCs that guarantees a *meaningful* bound on the smallest dihedral angle.



- Anisotropic surface meshing that's both practical and provably good.
- There's still room for improvement in meshing 3D domains with small angles. Mesh this domain with guaranteed quality *and* without adding many new vertices in practice.

My biased opinion: I think 3D CDTs will be part of the best−performing algorithm of the future.



Download these course notes!

http://www.cs.berkeley.edu/~jrs/jgatalk.pdf

Fin

Provably good software! **Triangle**

http://www.cs.cmu.edu/~quake/triangle.html