Complexity and Computation of 3D Delaunay Triangulations

Nina Amenta (UC-Davis)

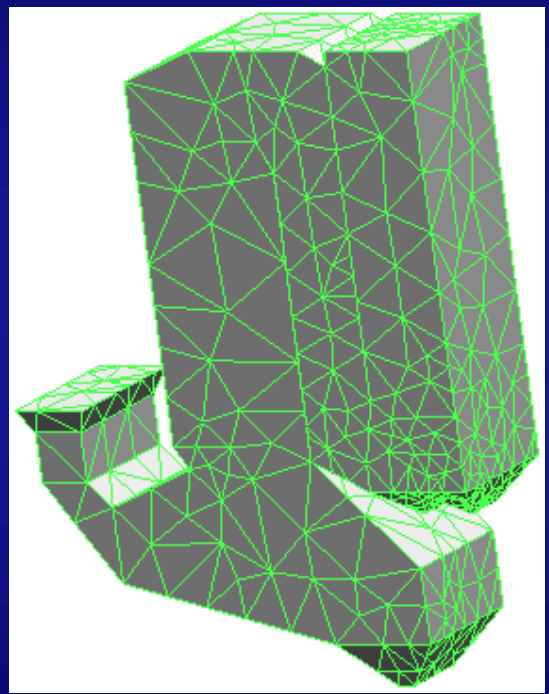
Large inputs in 3D

Surface reconstruction and medial axis construction inputs: 20,000 - 20,000,000 samples

Mesh generation

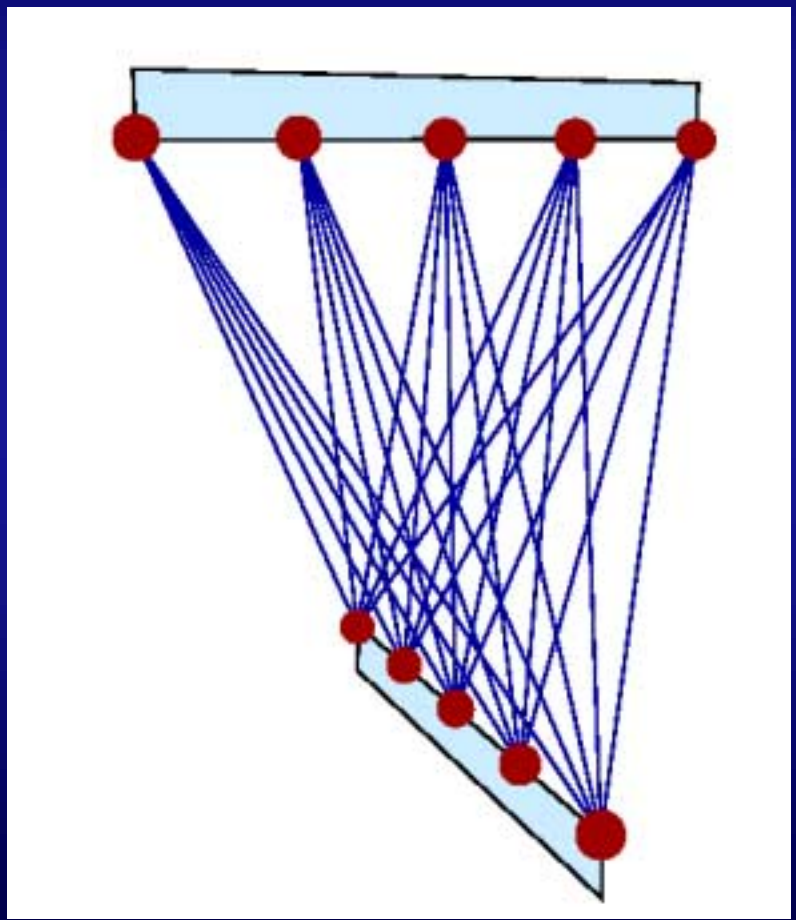
Needed for finite element simulations; very large inputs representing region boundary.

3D particularly important.



Shewchuk (98?)

3D Delaunay $O(n^2)$ ...

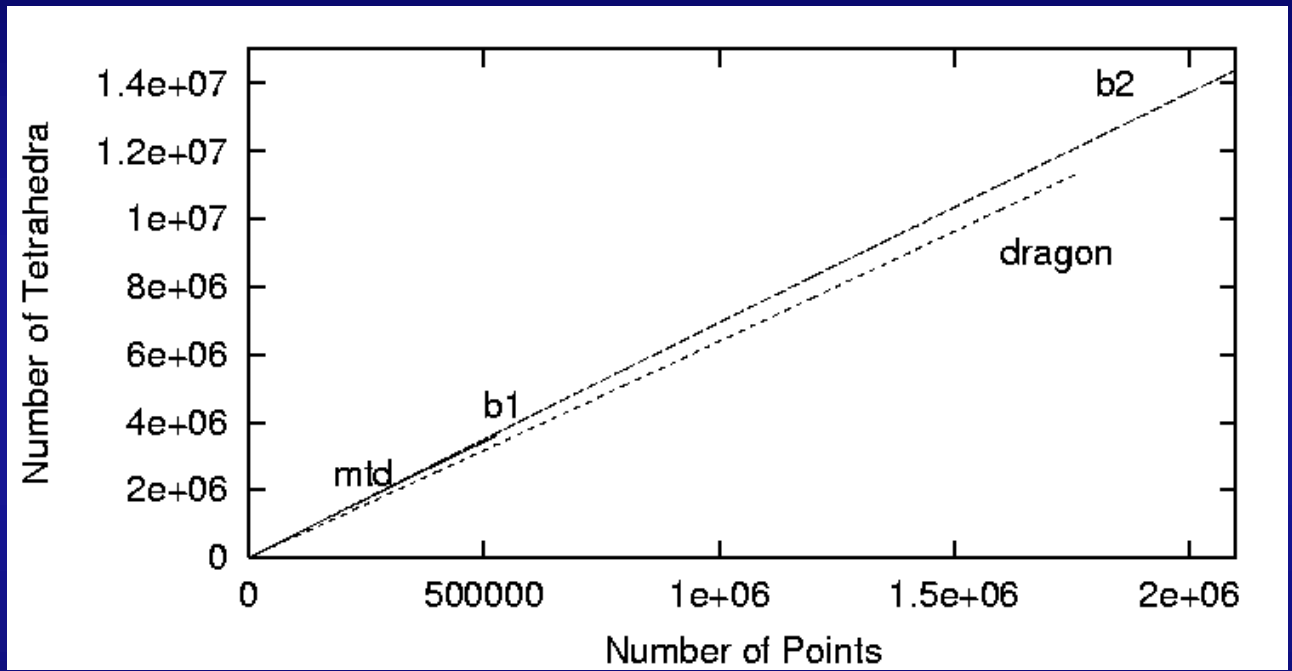


n/2 points on each of two skew lines.

Points on moment curve, etc.

All examples distributed on 1D curve?

… but linear in practice.



A & Choi, 01

Adding samples from surfaces in random order, # tetrahedra grows linearly.

Linear special cases

Dwyer 91 - Uniform random in ball (any constant dimension)

Golin & Na 00 - Uniform random on surface of convex polyhedron

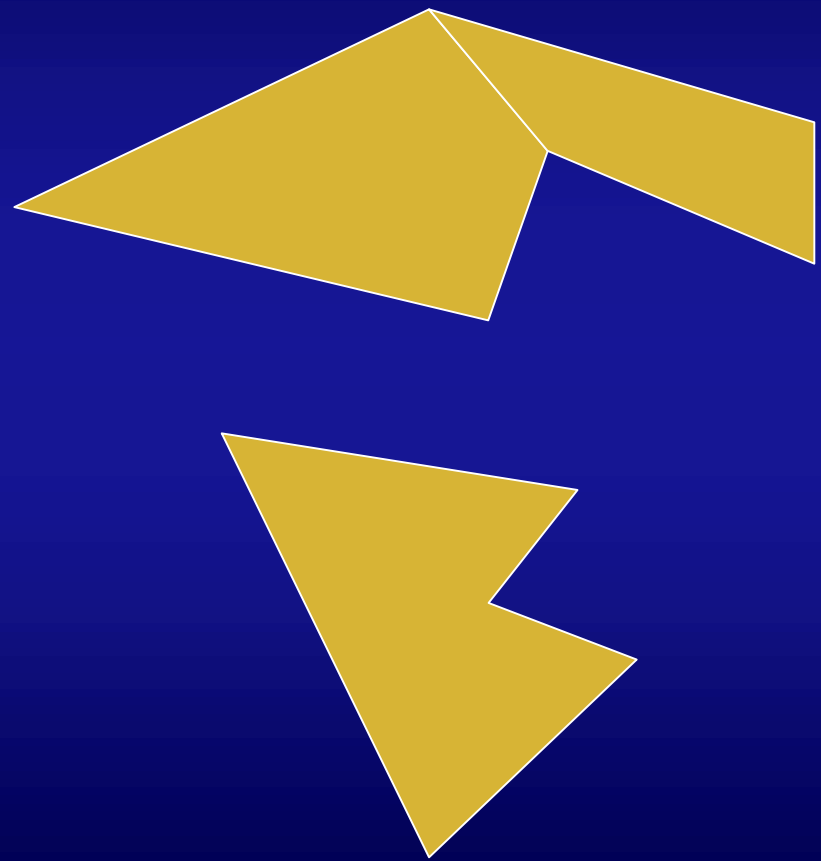
Attali & Boissonnat 02 - Nicely sampled polyhedral surface

Golin & Na 02 - Uniform random on polyhedral surface, $O(n \log^4 n)$ (almost linear!)

Nicely sampled polygonal

Fixed set of polygons in R^3, only intersect at boundaries.

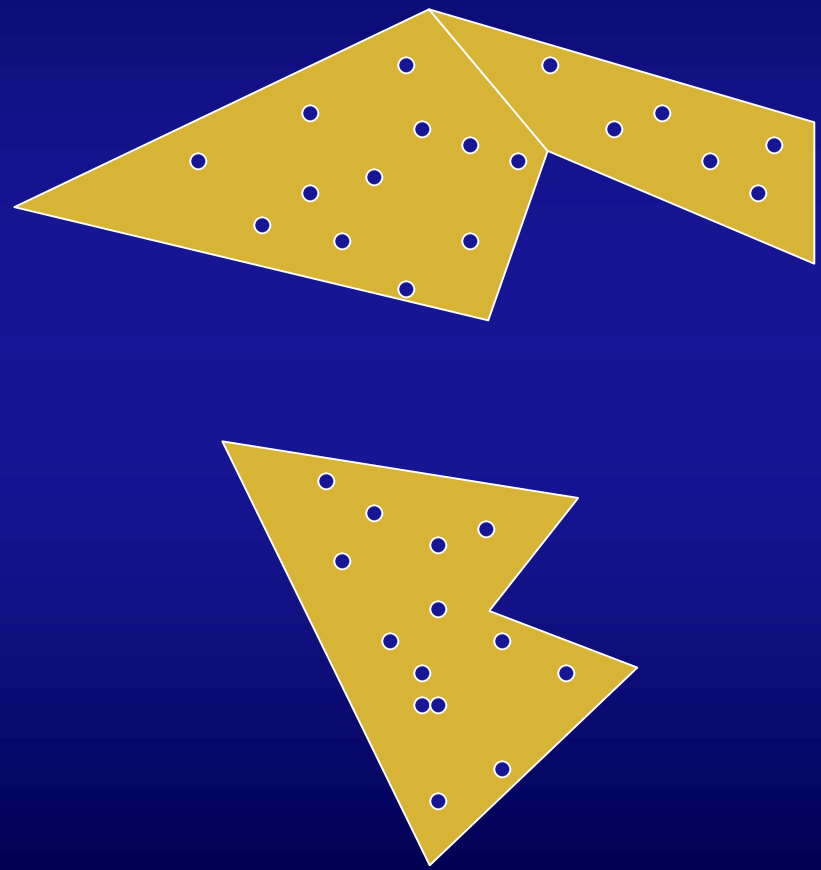
Area, boundary length, number of polygons constant.



Sampling model

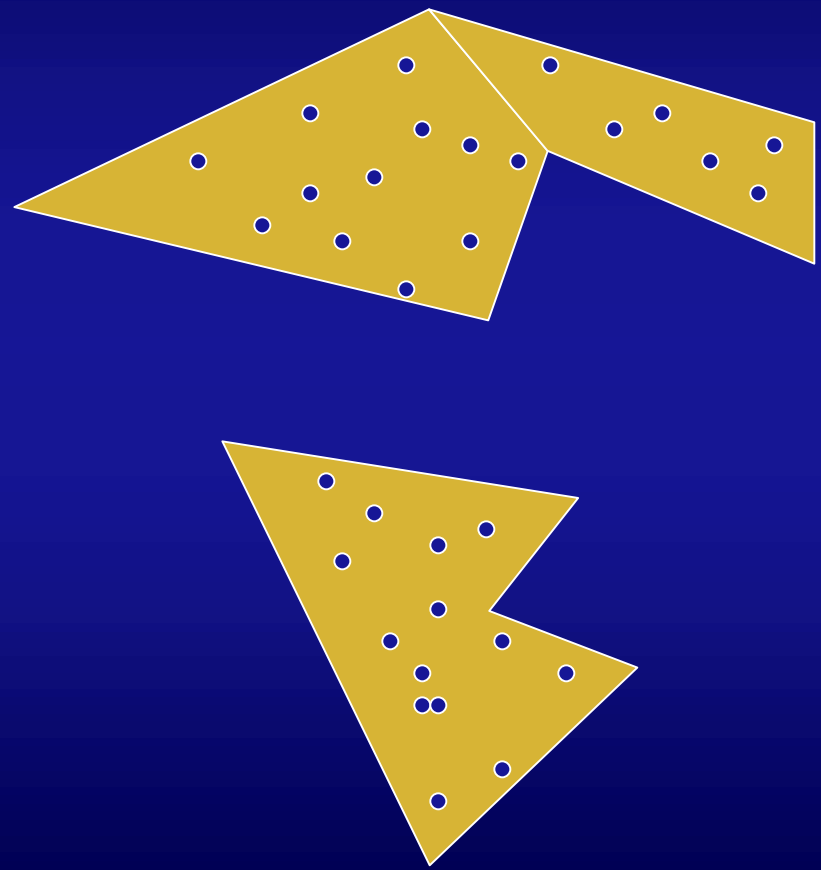
Every point has at least one and at most k samples within distance ε. Consider behavior as ε->0.

$n = O(1/\varepsilon^2)$



Attali & Boissonnat 03

Complexity of Delaunay triangulation is O(n).



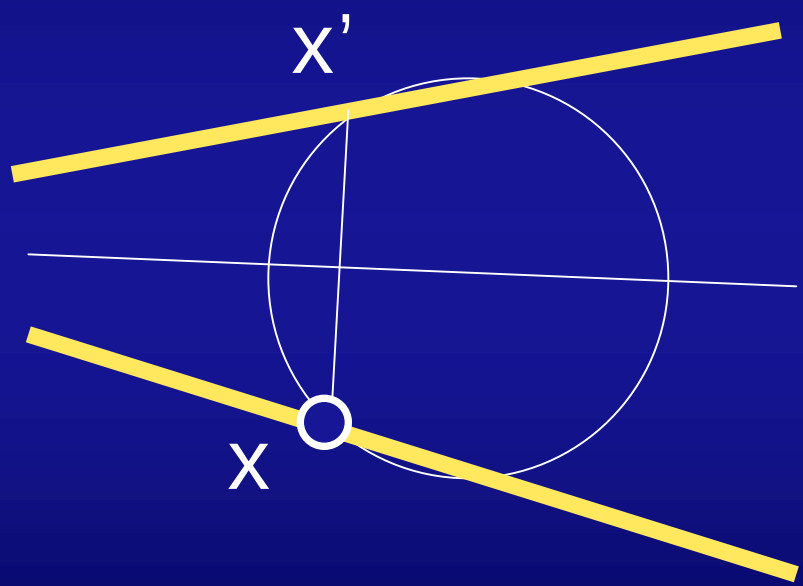
Other sampling models

Every sample has a nearest sample at distance at most ε and at least $\varepsilon/2$.

Every point has sample within distance ε —– not appropriate!

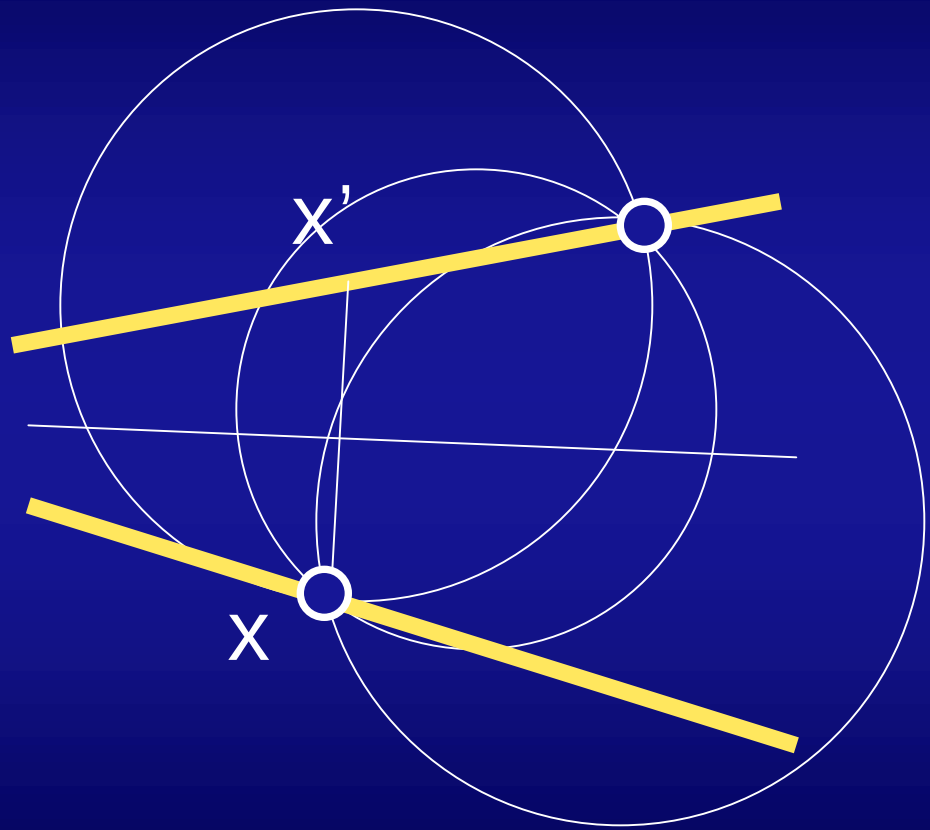
Easy modulo edge effects …

x' is reflection of x across planar bisector. Voronoi balls nearly tangent to face at x are close to x'.



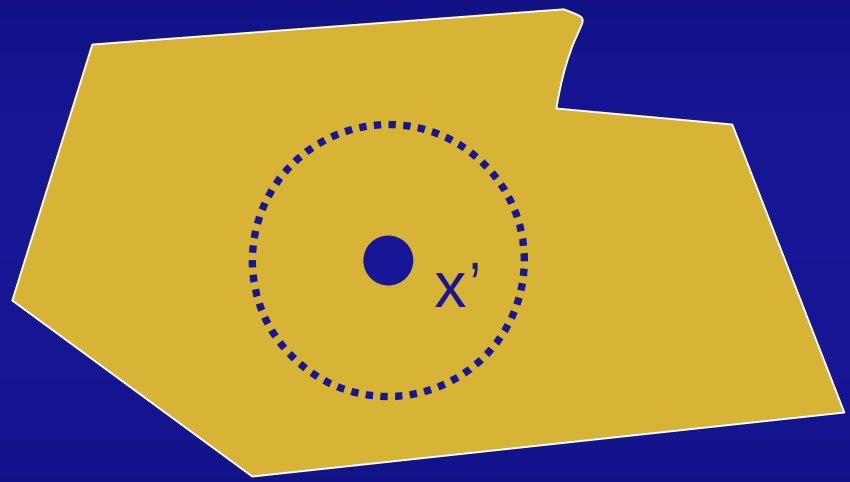
Modulo edge effects …

Any ball touching a point far from x' on the opposite plane contains too much area to be empty.

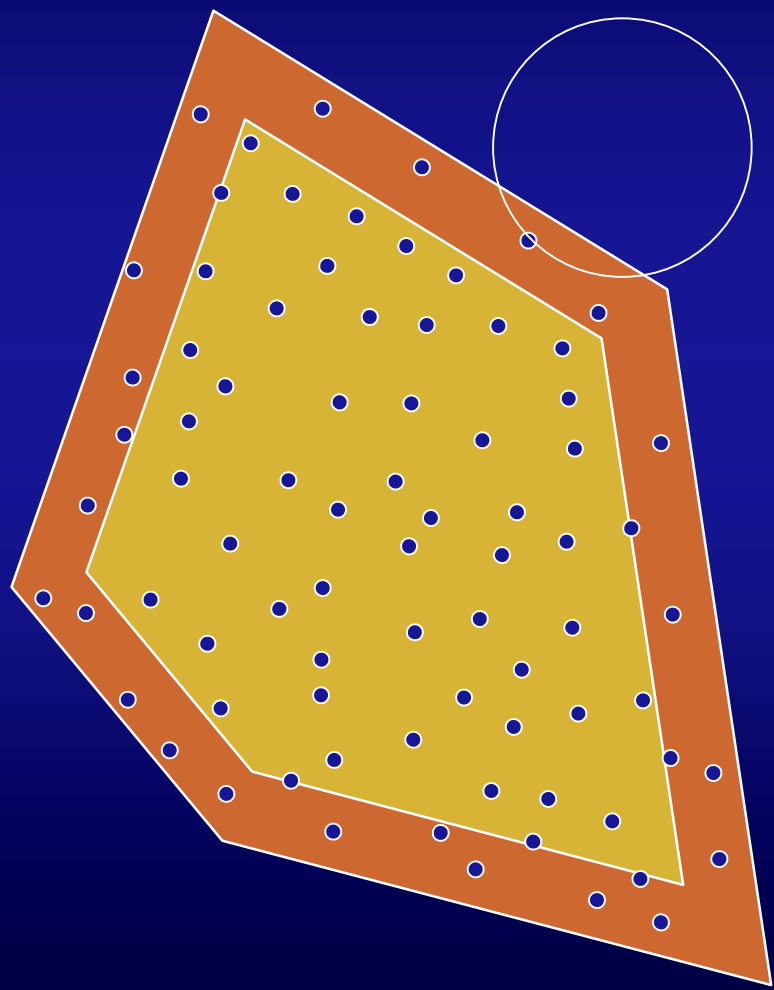


Modulo edge effects …

All Delaunay edges from x into interior of another polygon have to end within O(ε) ball around x', so only O(1) such.



Singular region



Singular region of width $O(\varepsilon)$ surrounding edges.

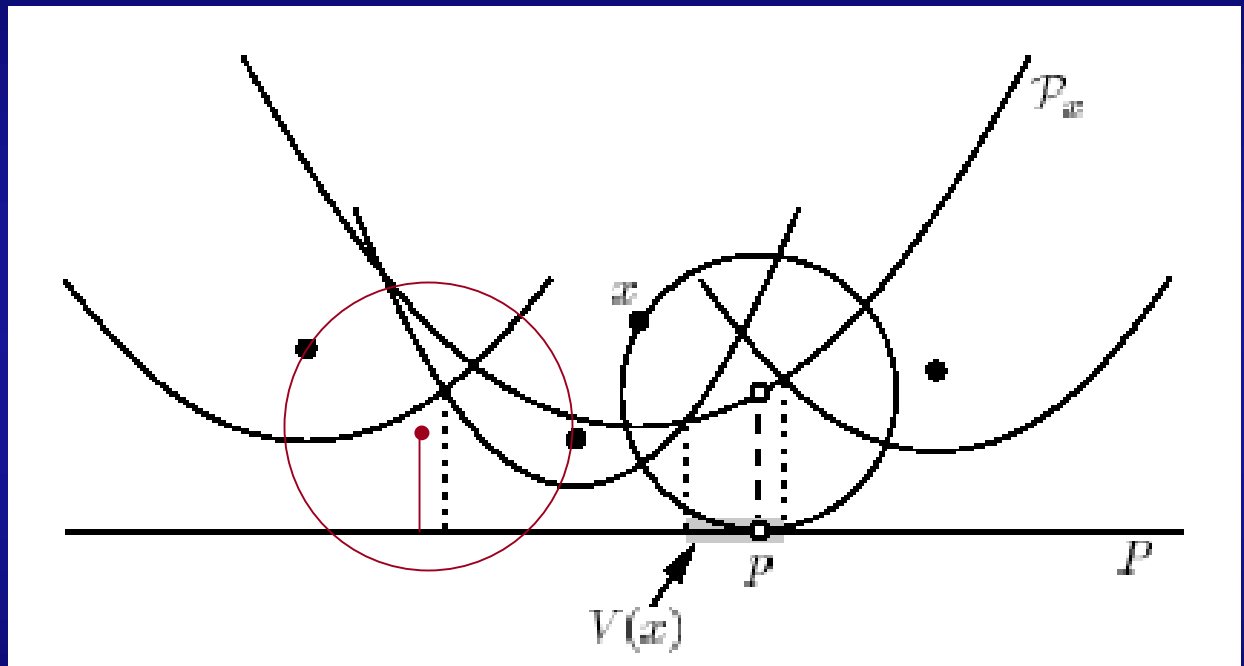
Singular to singular

Only O(sqrt n) samples total in all singular regions so O(n) singular-to-singular Delaunay edges.

Singular to non-singular

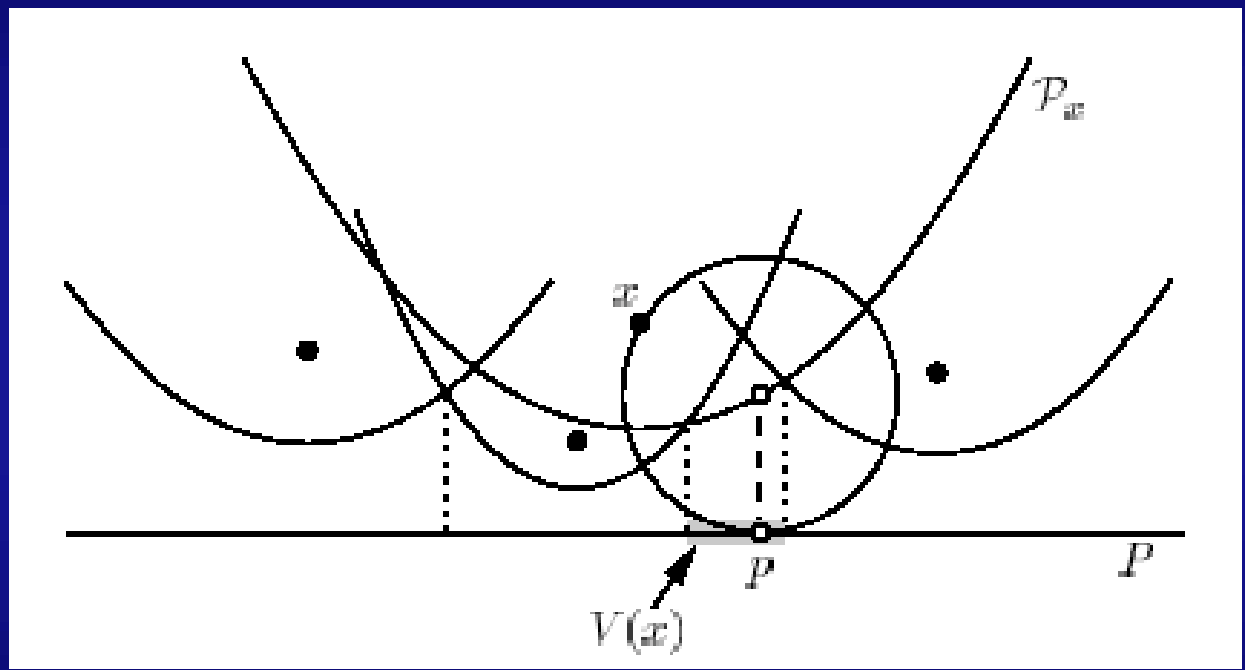
$O(n^{3/2})$ is easy; to get O(n), consider singular points above each polygon (below polygon similar).

Singular to non-singular

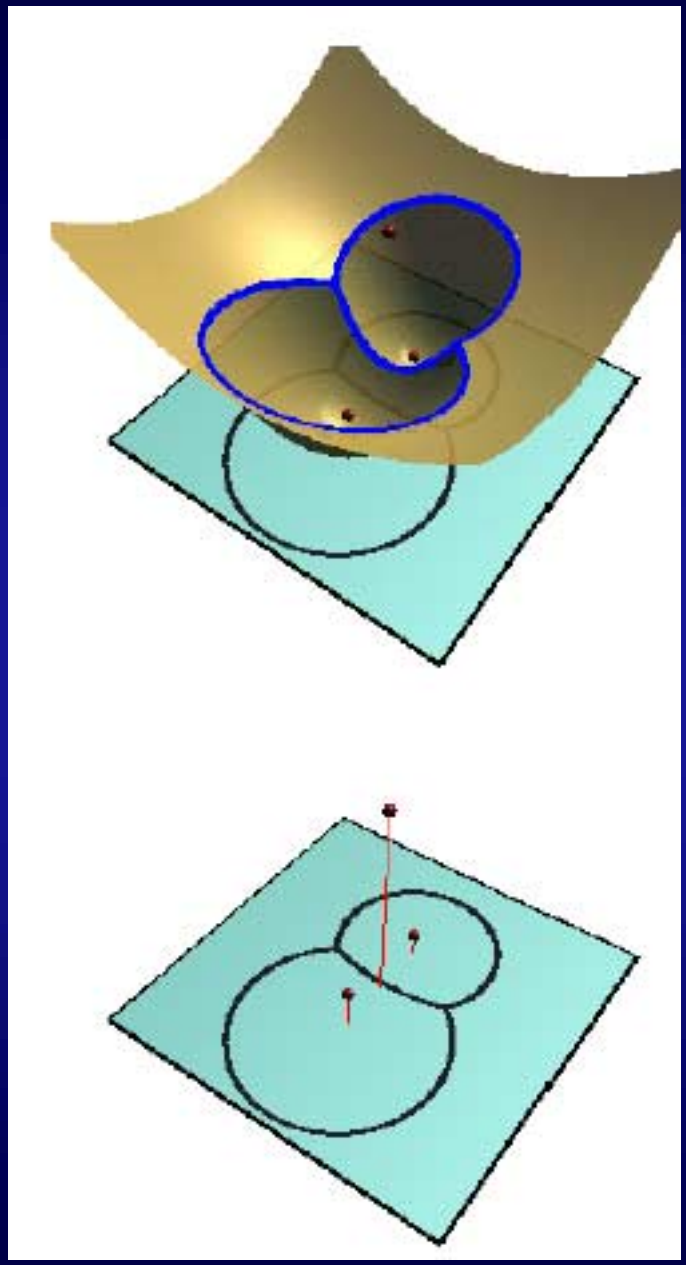


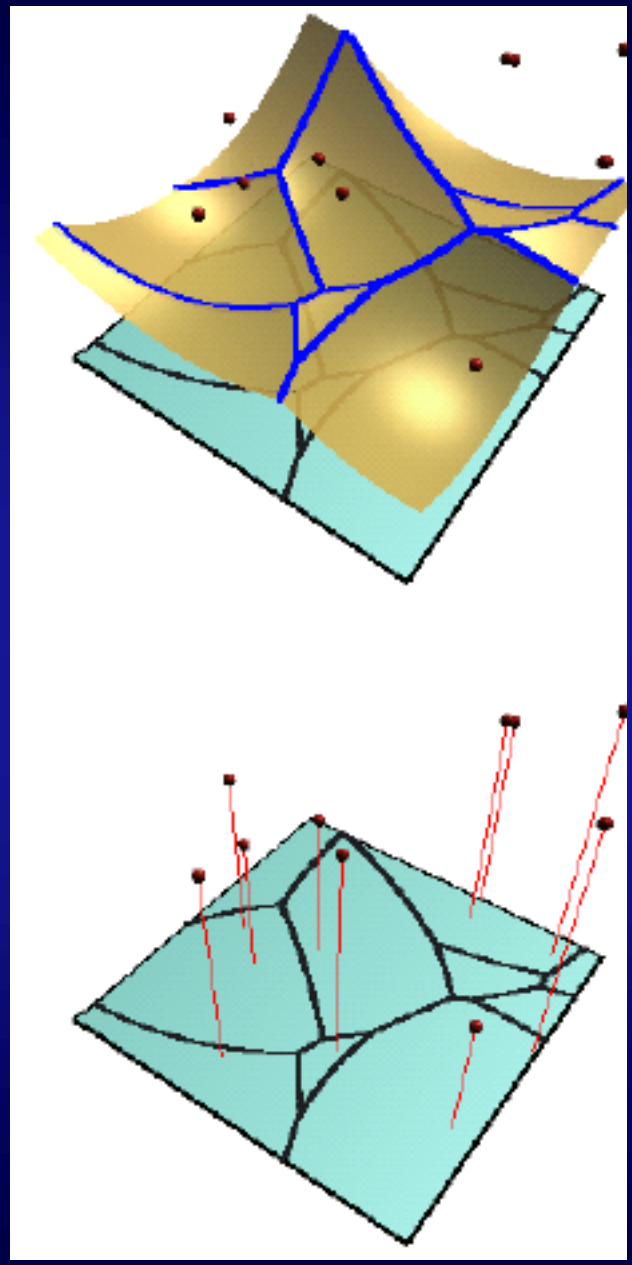
Assign point p on plane to first singular point hit by growing tanget ball.

Singular to non-singular

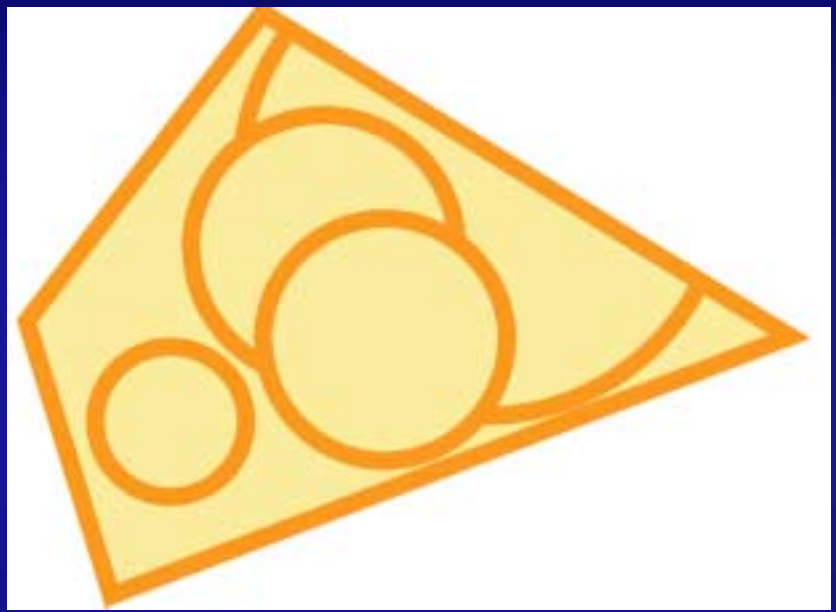


Assign point p on plane to first singular point hit by growing tanget ball.



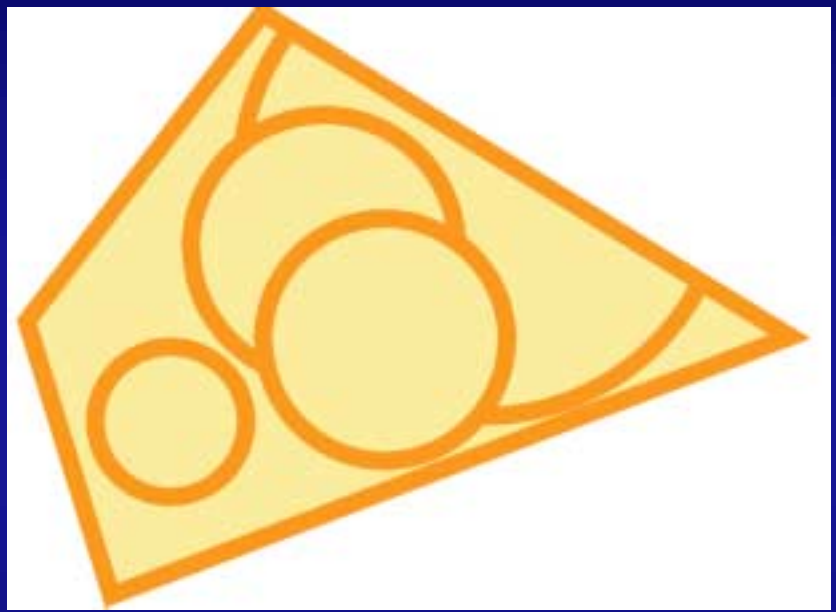


Singular to non-singular



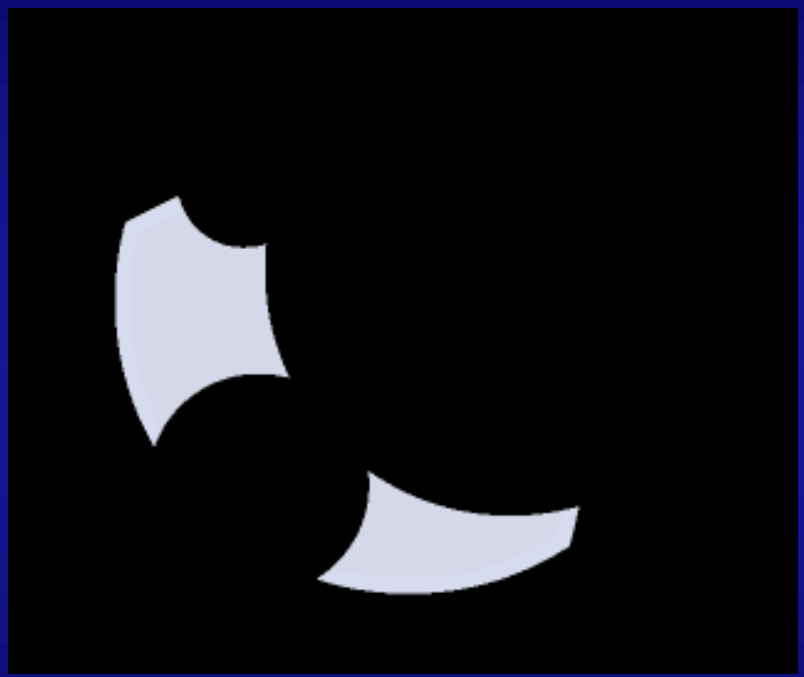
Decomposition of polygon with circular arcs. Region (possibly disconnected) for each singular samples above polygon.

Singular to non-singular



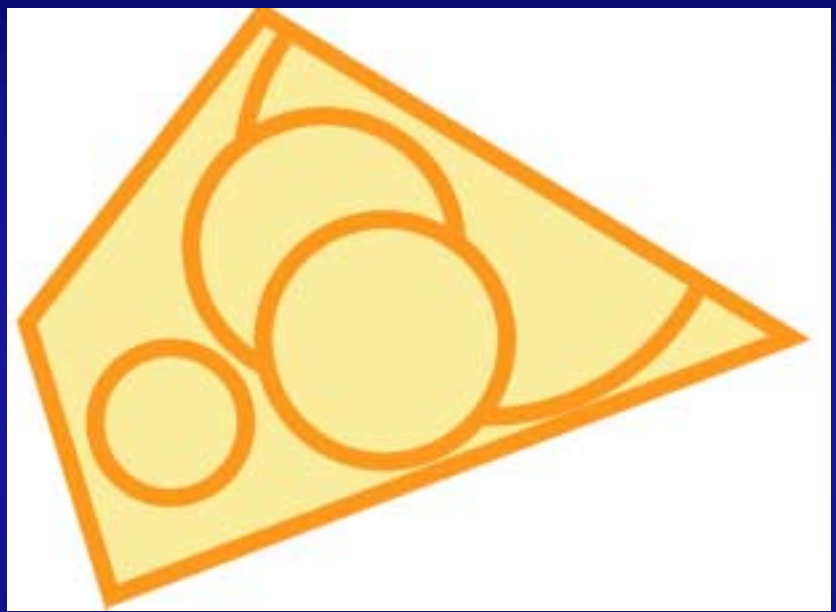
Delaunay balls are nearly tangent to polygon. Samples on polygon with Delaunay edge to s lie within 2ε of region of s.

One region



Amount by which size of region is expanded is $O(\varepsilon \lambda)$, where λ is boundary length.

Singular to non-singular



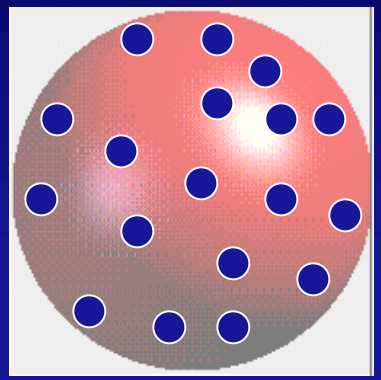
Total extra area over all regions = $O(\mathrm{sqrt}(n)\,\varepsilon) = O(1)$, so number of additional Delaunay edges =

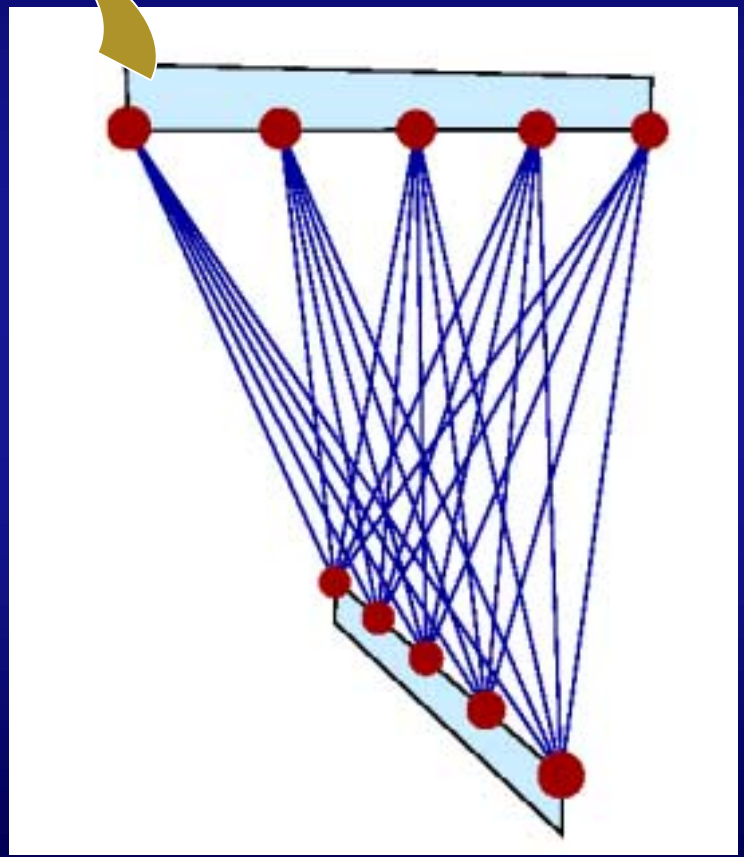
$O(1/\varepsilon^2) = O(n)$.

Lower bounds

Jeff Erickson (by Howard Sun)

Lower bounds





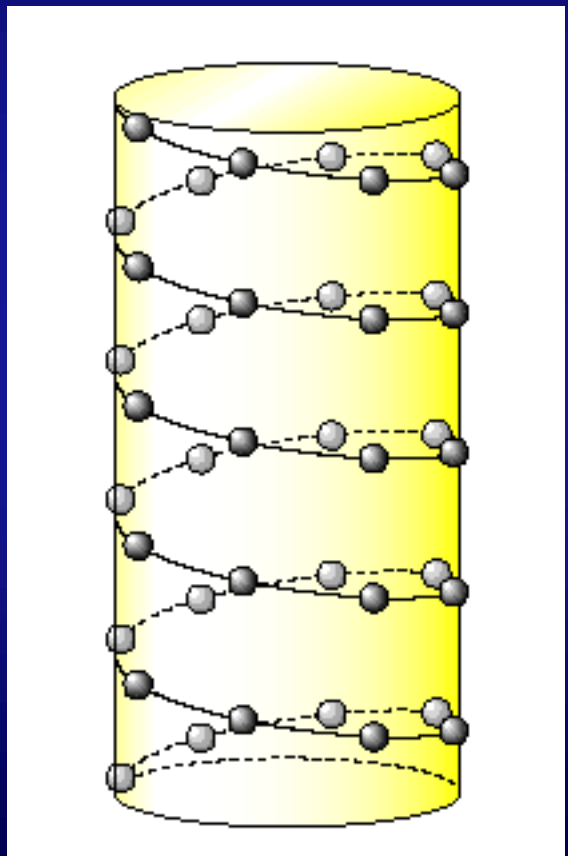
Given n, ε , can construct a suface and an ε–sample with $O(n^2\varepsilon^2)$ triangulation.

$O(\varepsilon n)$ balls

Lower bounds

Helix with sqrt(n) turns, sqrt(n) samples per turn.

Fact (Erickson, Bochis & Santos): ball tangent to cylinder at 2 samples in same turn contains no other samples -> $O(n^{3/2})$ Delaunay edges.



Generic surfaces?

Attali, Boissonnat & Lieuter, SoCG 03

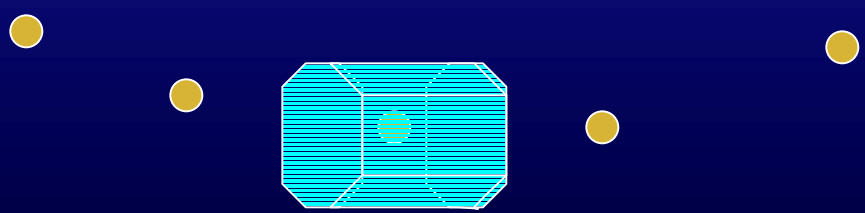
"Generic" smooth surface S: singular points (with osculating maximal tangent balls) form a 1D set with fixed length L.

At least one sample, at most k, in ball of radius ε, ε->0.

O(n log n) - best possible?

Higher dimensions?

Conjecture: Nice distribution of samples from surface of co-dimension c has Delaunay triangulation of complexity $O(n^{\lfloor (c/2) \rfloor + 1})$.

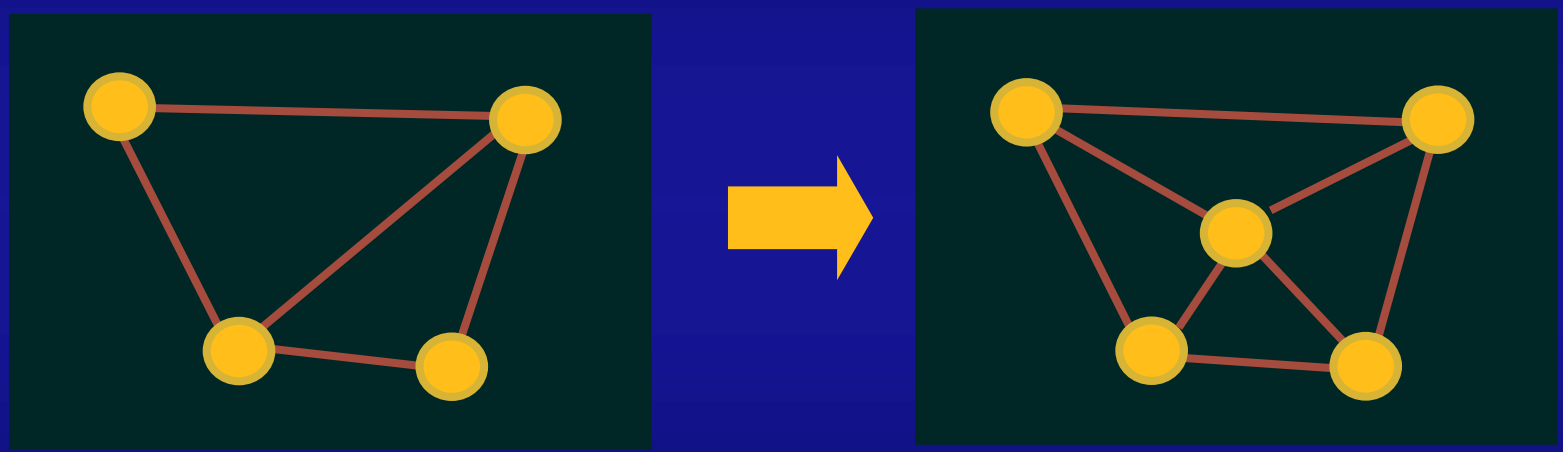


Special case of convex hull

Dimension 3 (or low)
n = 20,000 - 20,000,000
Theoretically optimal O(n lg n + t)
t = O(n)

Contrast with arbitrary-dimension possibly non-simplicial convex hull/halfspace intersection codes like **lrs, porta, cdd.**

Randomized incremental algorithm



Add points one by one in random order, update triangulation with flips. Simple, optimal (worst-case expected time).

Implementations

delcx - Edelsbrunner, Muecke, Facello 92,96

hull - Clarkson 96

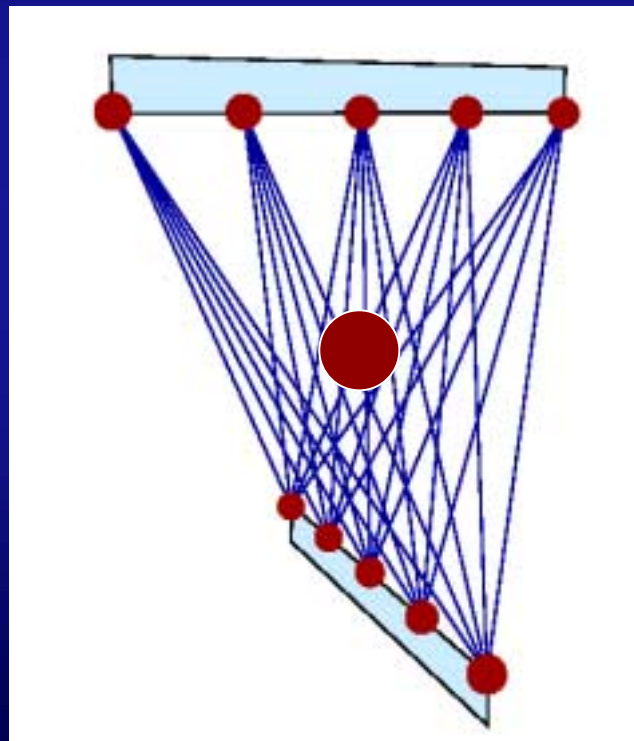
CGAL Delaunay hierarchy - Devillers, Teillaud, Pion 01

pyramid - Shewchuk, unreleased

Not generally output sensitive

Avis, Bremner & Seidel, 91

"Dwarfed" polytopes, t = O(n) but intermediate polytopes in construction have $t_i = O(n^2)$.



… but close in our case.

Golin & Na 02 - Uniform random on polyhedral surface, $O(n \log^4 n)$

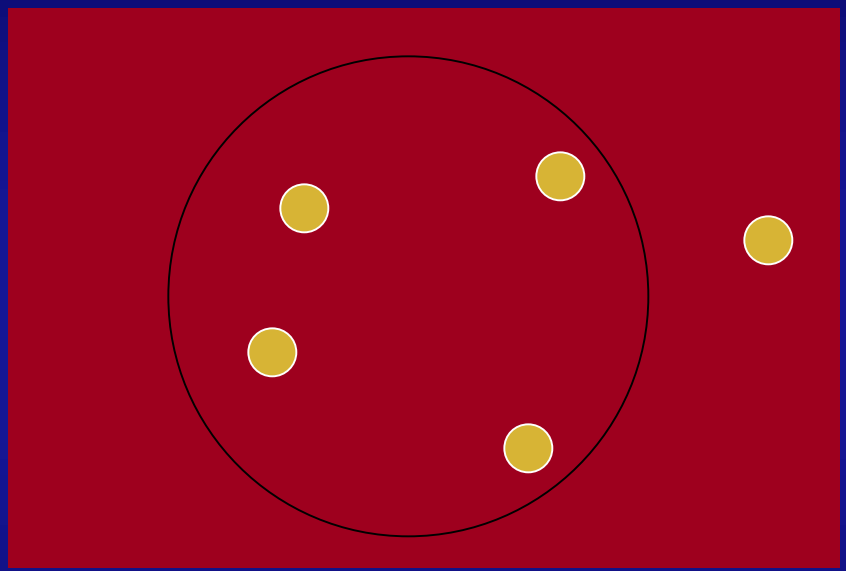
For large enough n, apply to large enough random subsets.

Degeneracy

All degeneracies are “accidental” - triangulated output is fine.

Simulation Of Simplicity, Edelsbrunner & Muecke, 90. Treat samples as if small non-degenerate perturbation was applied.

Robustness



Primitive insphere operation: is 5th sample in sphere through given 4?

Numerically unstable! Incorrect answers can cause topologically impossible output or crash!

Robustness

Exact computation more popular than algorithm re-design.

For efficiency, compute only to accuracy required to determine answer.

CGAL - many choices of filtering/exact arithmetic combos

pyramid - adaptive floating point

Point location strategies

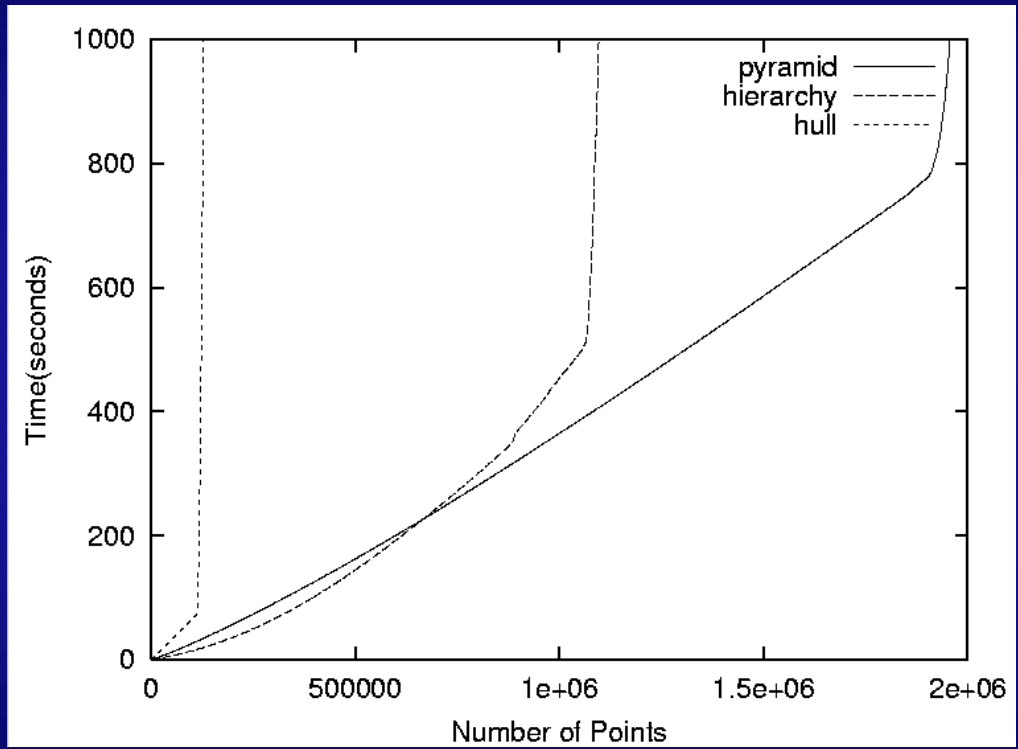
Theoretical bottleneck.

O(log n) per location possible with search data structure, but is it worth the effort in practice?

CGAL, hull - data structures

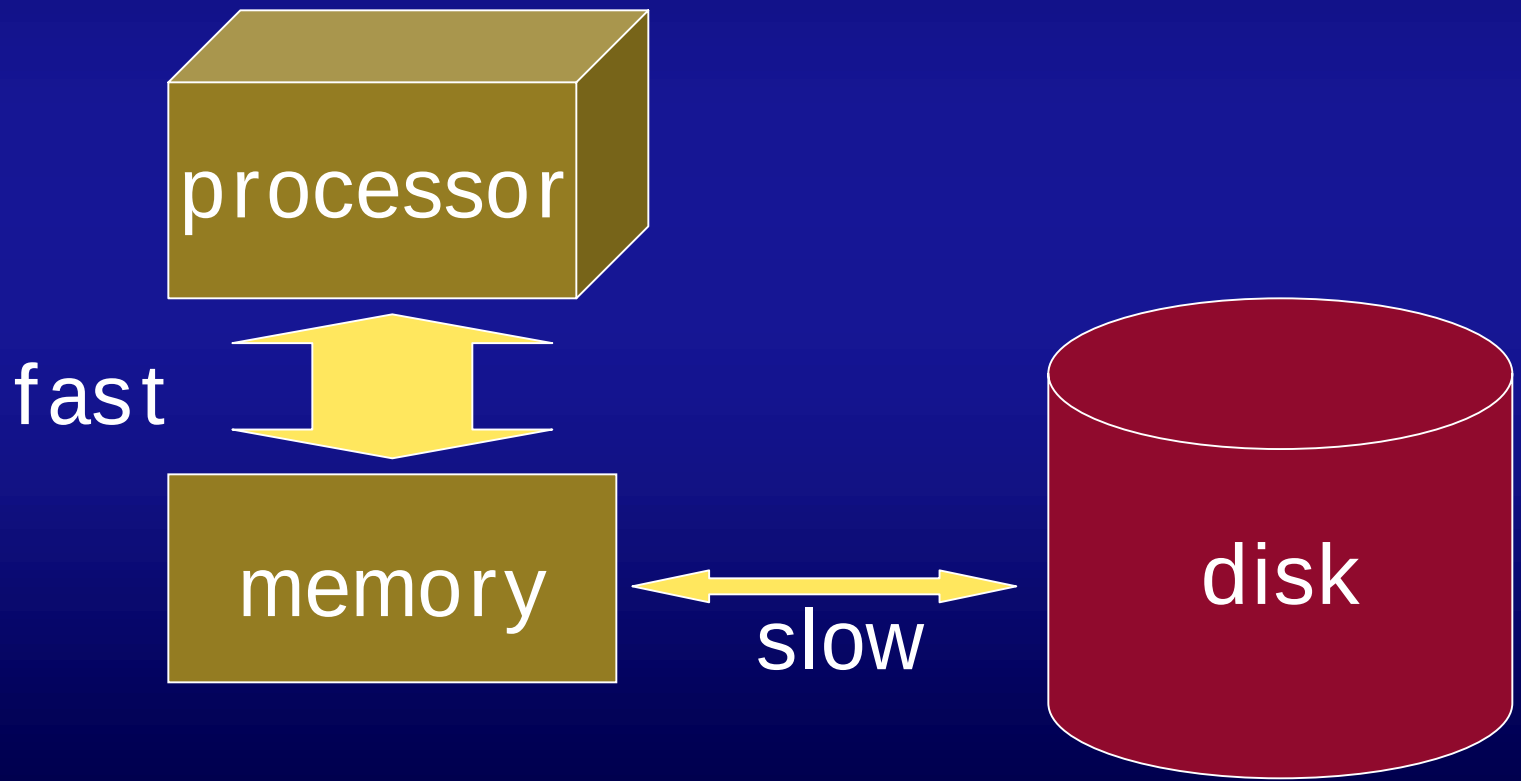
delcx, pyramid - no data structures

Memory usage

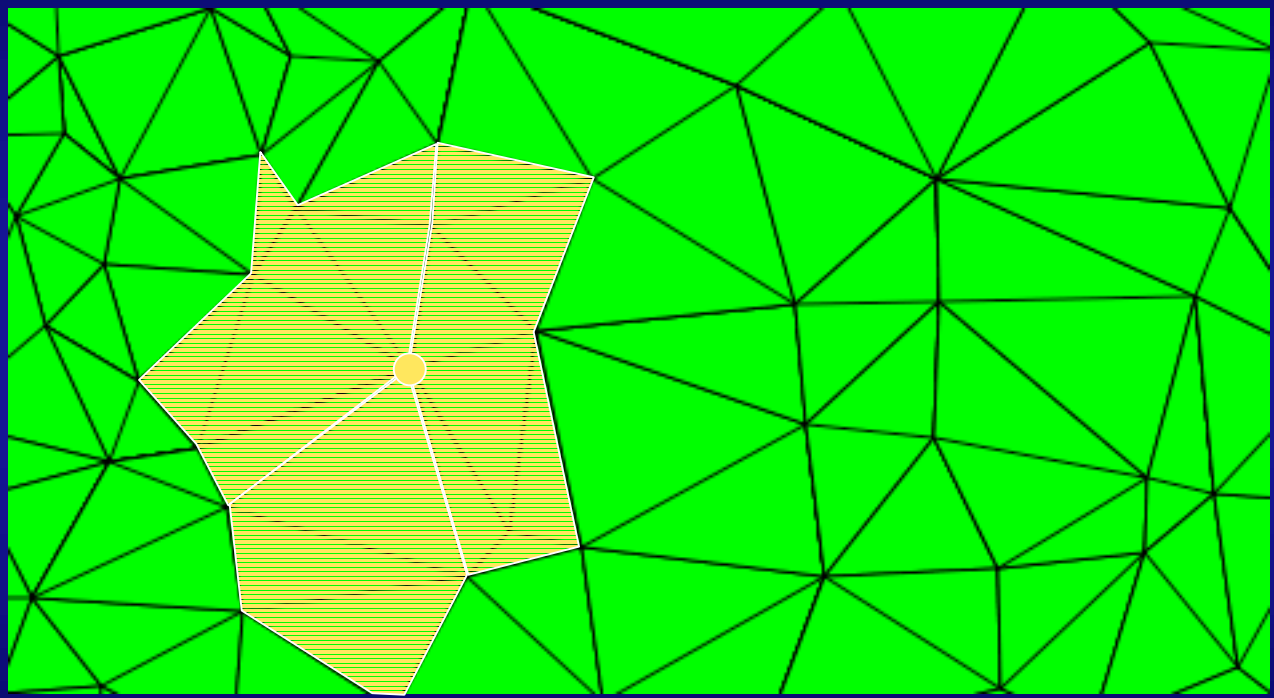


Performs great…until !

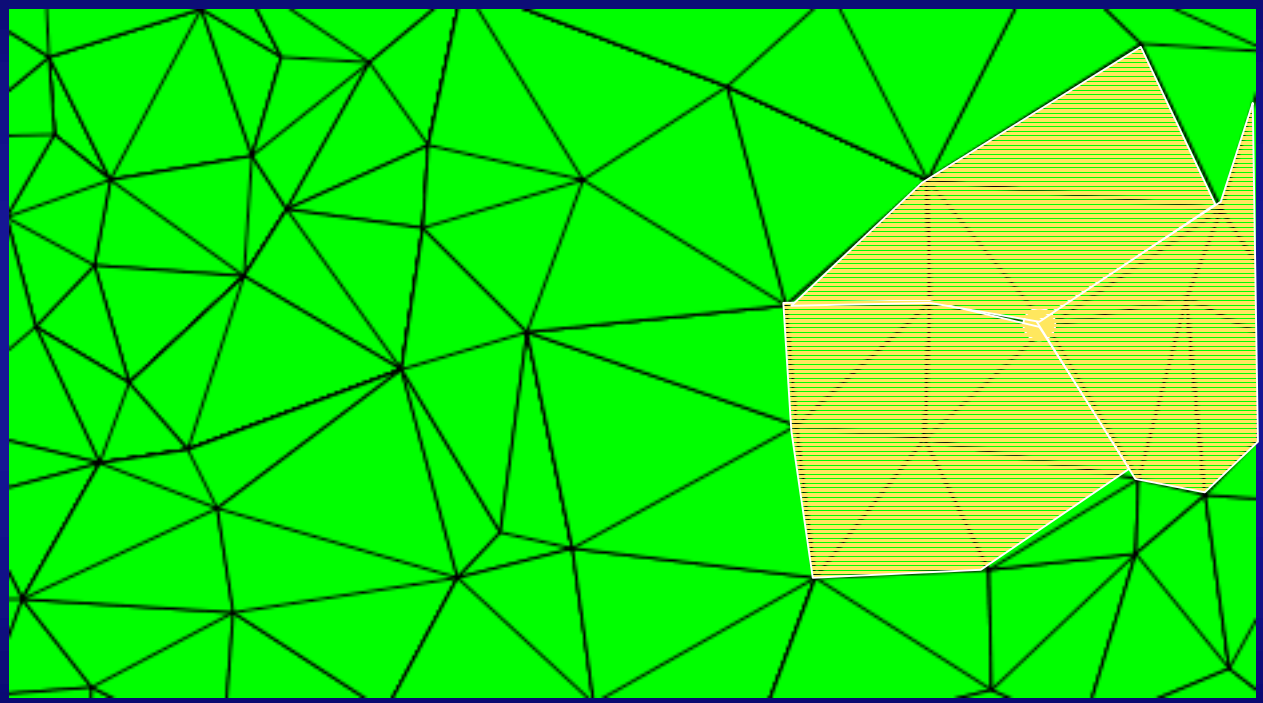
Thrashing



Thrashing



Thrashing



Idea

Partially randomized insertion order

- increase locality of reference, especially as data structure gets large
- retain enough randomness to guarantee optimality

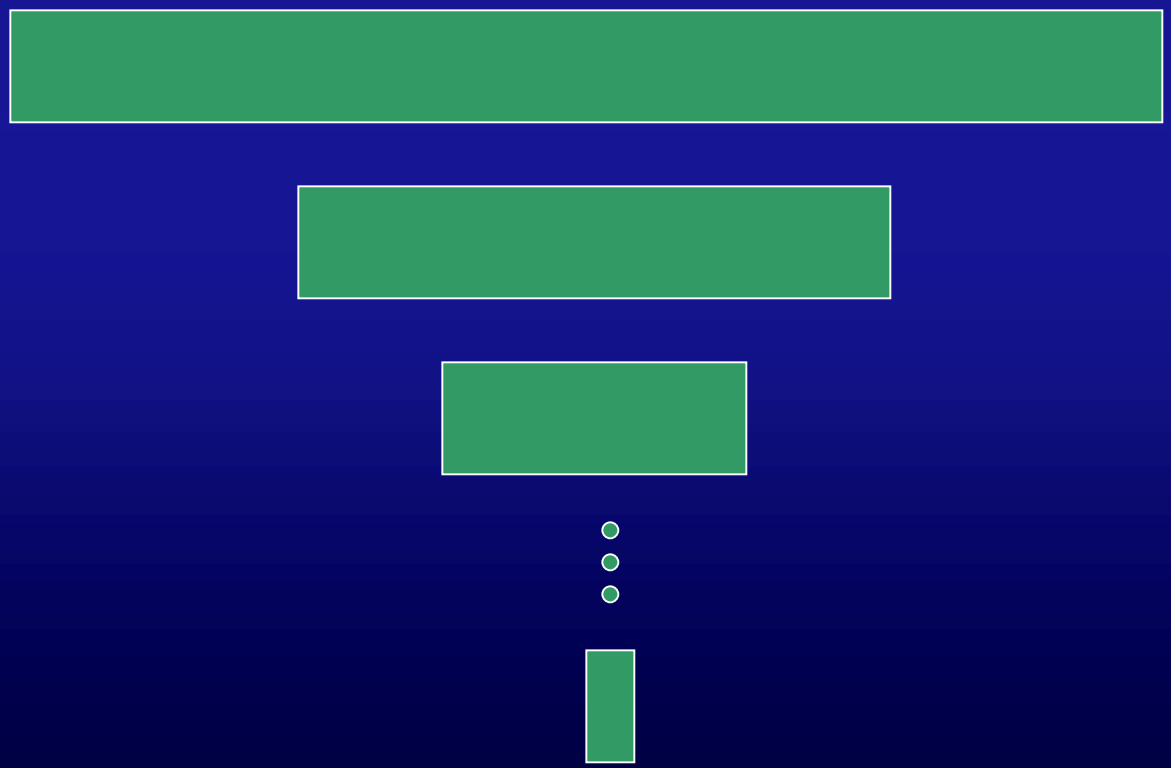
Biased Randomized Insertion Order (BRIO)

(A, Choi, Rote, 03)

- Choose each point with prob = 1/2.
- Insert chosen points recursively con BRIO.
- Insert the remaining points in arbitrary order.

BRIO

log n rounds of insertion



round logn

round logn-1

round logn-2

round 0

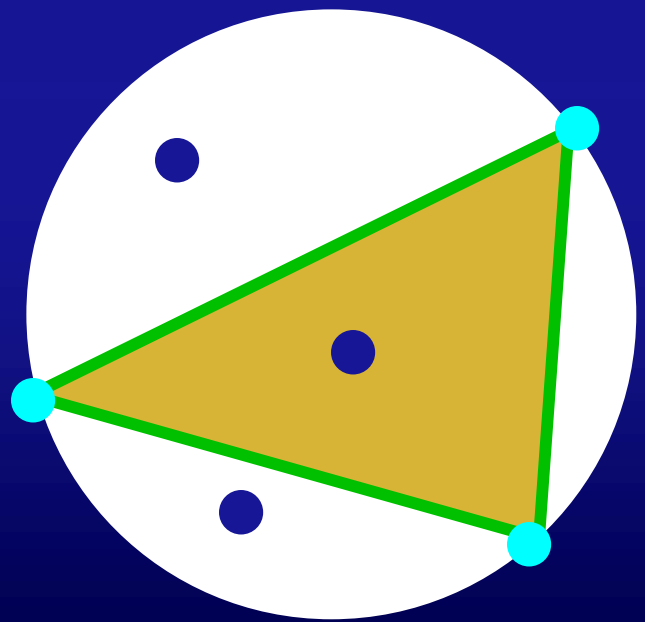
Analysis

Randomness has two benefits:

- Bound total number of tetrahedra
- Bound time required for locating new points in triangulation

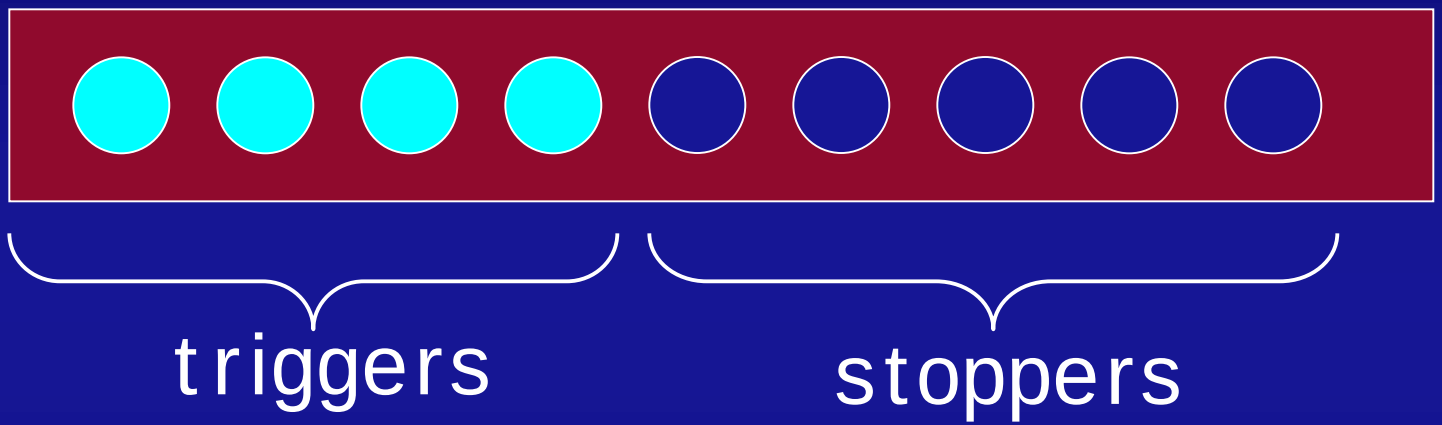
Analysis

Adapted from Clarkson and Shor, Mulmuley



- triggers
- stoppers

Triggers and stoppers



A tetrahedron appears during construction if all its triggers are inserted before any of its stoppers.

Analysis Sketch

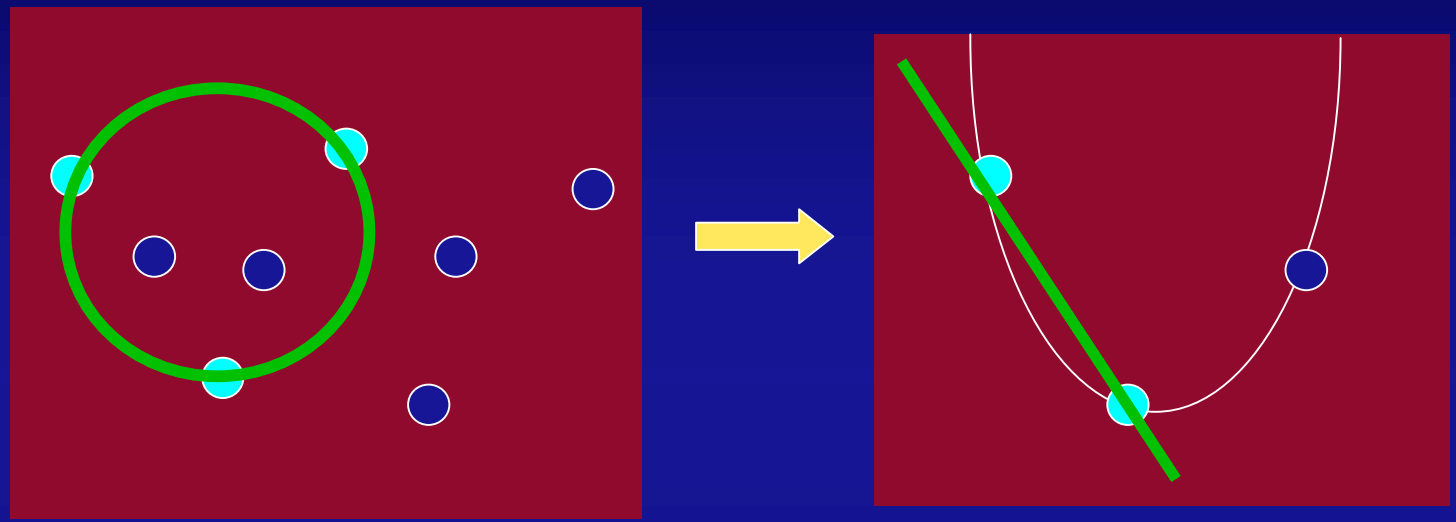
E[total number of tetrahedra] = $\sum_s k_s\, p_s$

E[running time] = $\sum_s k_s\, p_s\, s$

k_s = number of possible tetrahedra with s stoppers

p_s = prob. that tetrahedron with s stoppers appears during construction

Bounding k_s

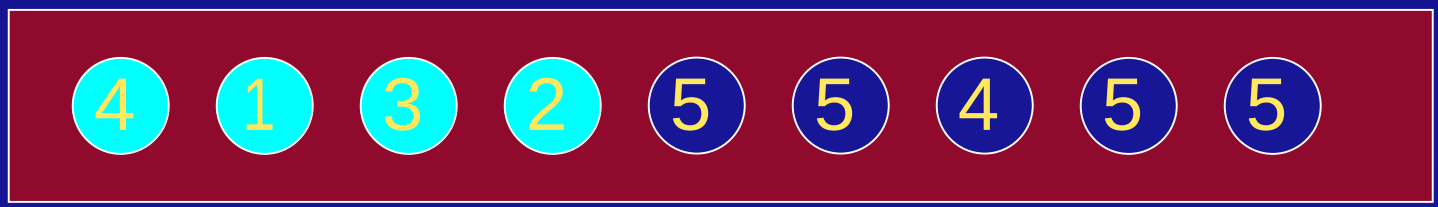


k_s = number of tetrahedra with s stoppers

number of subsets of size s seperatable by hyperplane in 4D - k-set problem

Bounding p_s

p_s <= P[the round where all triggers are chosen is <= the first round where any stopper is chosen]



= P [s+4 random numbers, the first 4 <= others]

= $O(1/s^4)$

Bounds

In “realistic case”:

Expected total number of tetrahedra = $O(n)$

Expected running time = $O(n \lg n)$

Bounds

In worst case:

Expected total number of tetrahedra $= O(n^2)$

Expected running time $= O(n^2)$

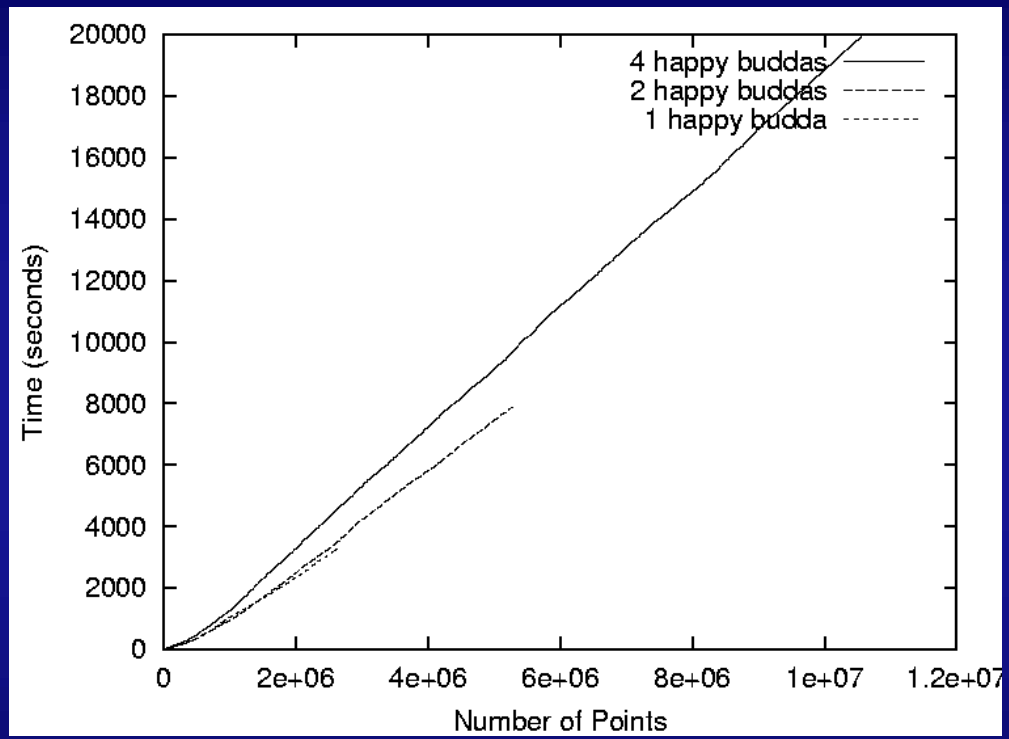
Experiments - pyramid

Point location: “walk” from last inserted point.

Multiple “Happy buddha”. 4096 kd-cells.

360 MHz, 128 M RAM, 4 GB Virtual memory

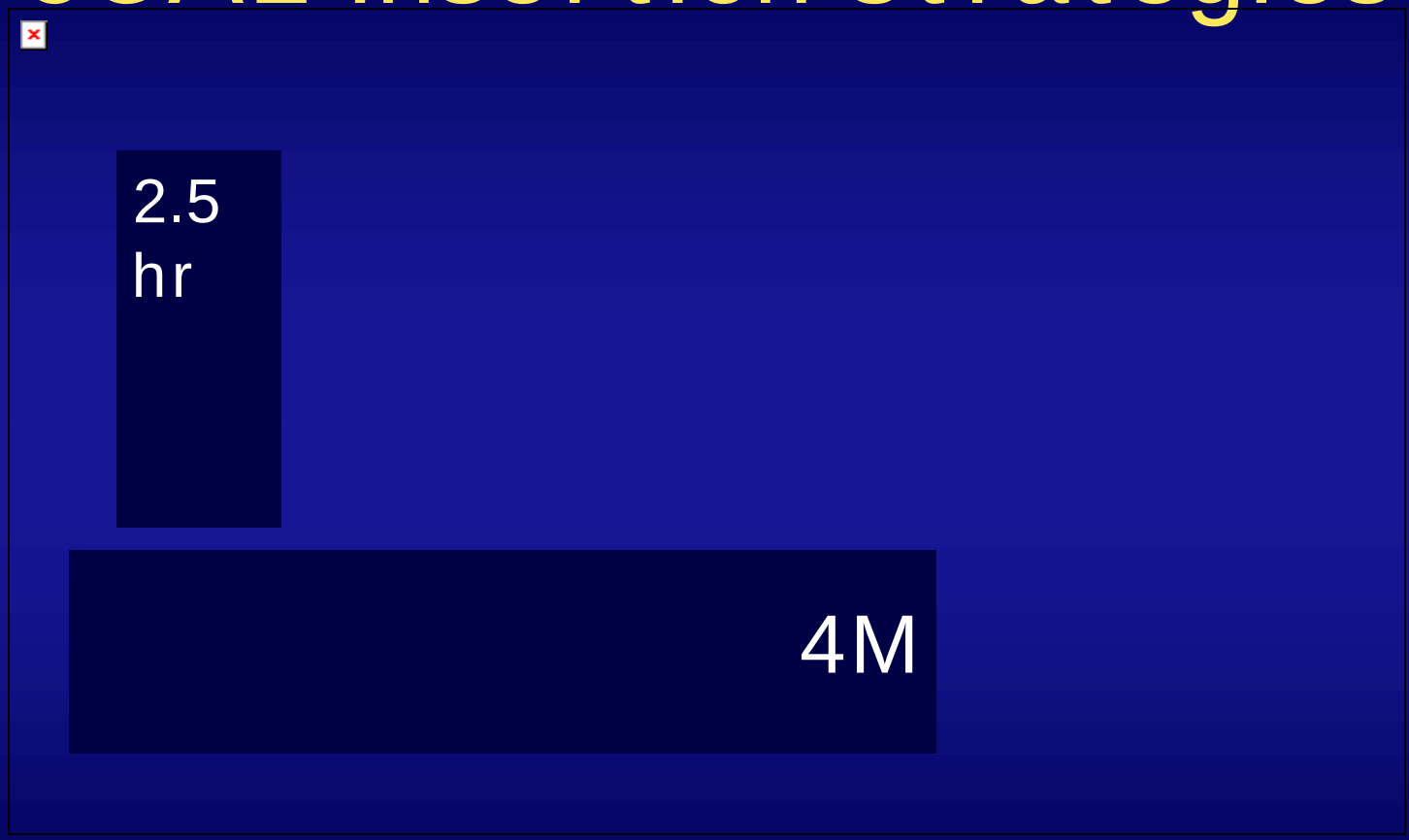
Pyramid



10 million points on tiny machine

1/2 hour on reasonable machine

CGAL insertion strategies



Thanks to Monique Teillaud and Ian Bowman.

Conclusion

Think of 3D Delaunay triangulation as essentially linear time, fairly efficient.

Replace this hack by real out-of-core implementation?