Recall

Polyhedra

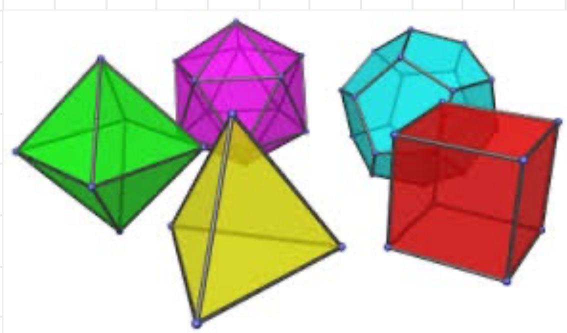
A ***polyhedron*** consists of a finite connected set of (plane) polygons called ***faces*** such that

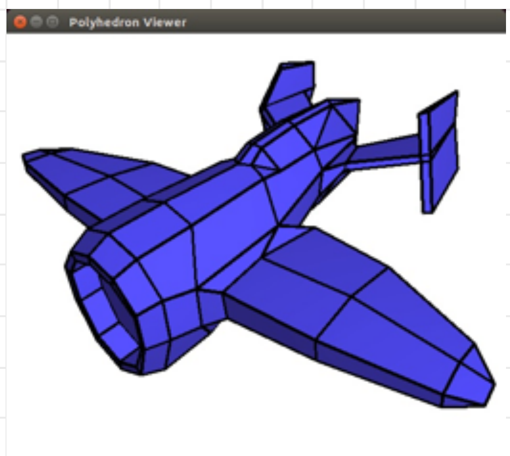
1. if two faces intersect it is only at a common vertex or edge

2. every edge of every face is an edge of exactly one other face

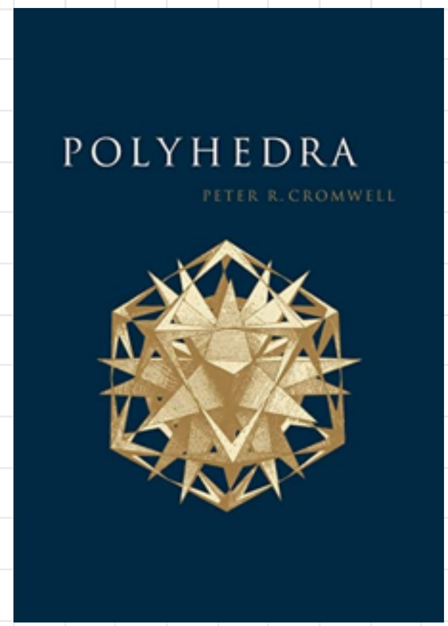
3. the faces surrounding each vertex form a single circuit

W https://en.wikipedia.org/wiki/Polyhedron





https://doc.cgal.org/latest/Polyhedron/index.html

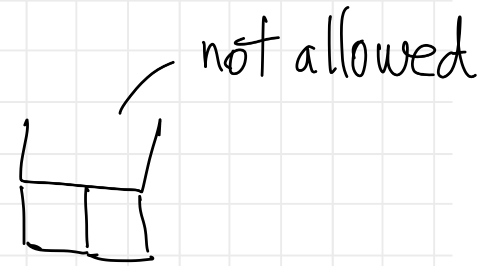


https://tinyurl.com/yy5tt439

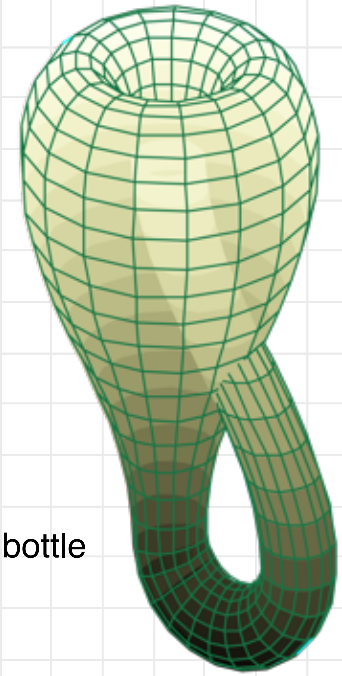
Recall

A ***polyhedron*** consists of a finite connected set of (plane) polygons called ***faces*** such that

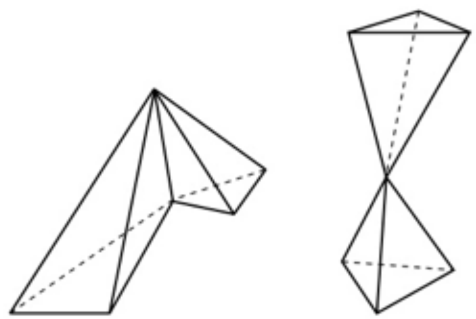
1. if two faces intersect it is only at a common vertex or edge
2. every edge of every face is an edge of exactly one other face
3. the faces surrounding each vertex form a single circuit



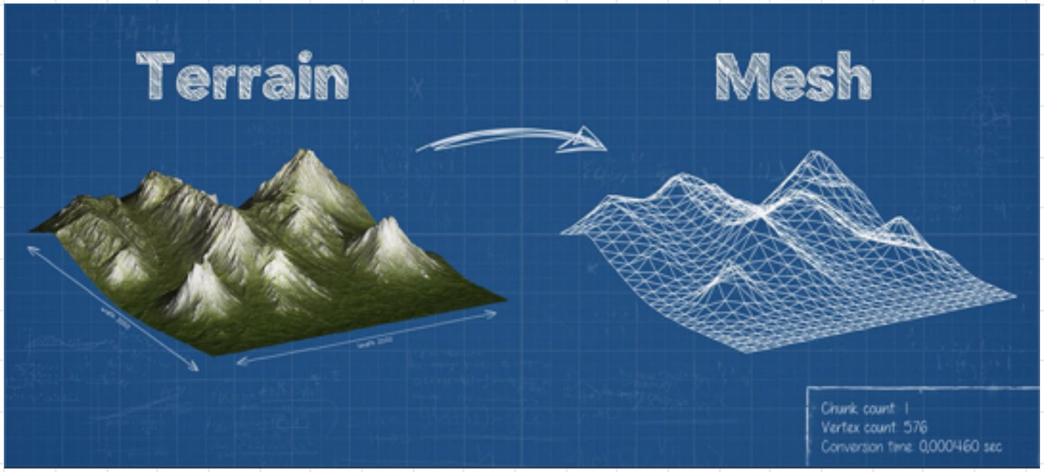
NOT Polyhedra



Klein bottle

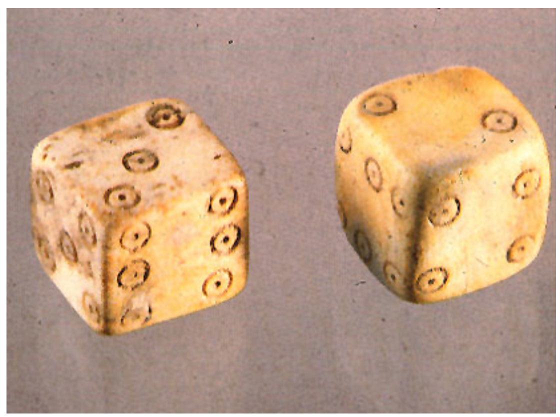


terrain



Mesh Materializer

Polyhedra

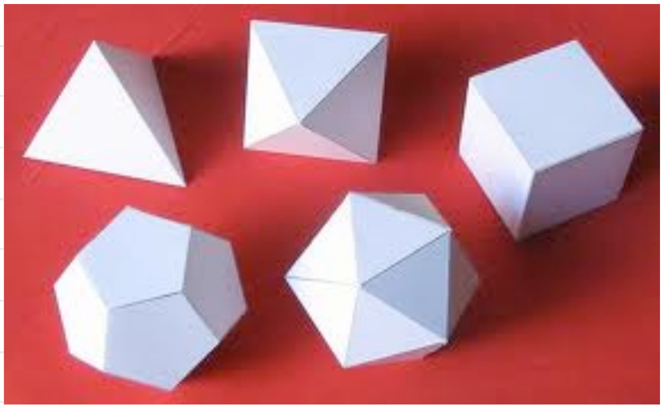


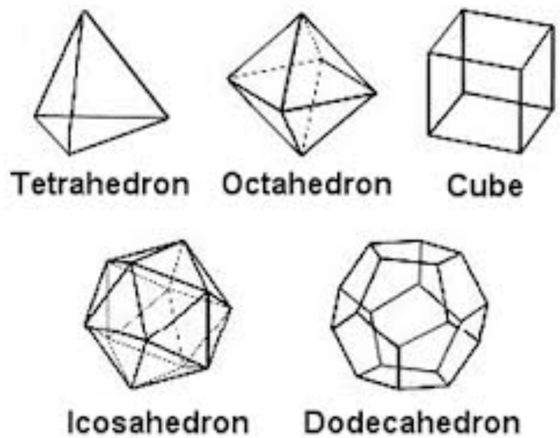
dice, Pompei, 1st century

icosahedral die, Roman, 2nd century

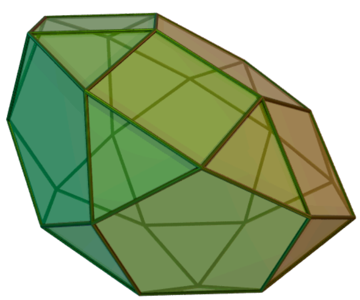
Polyhedra

Platonic solids

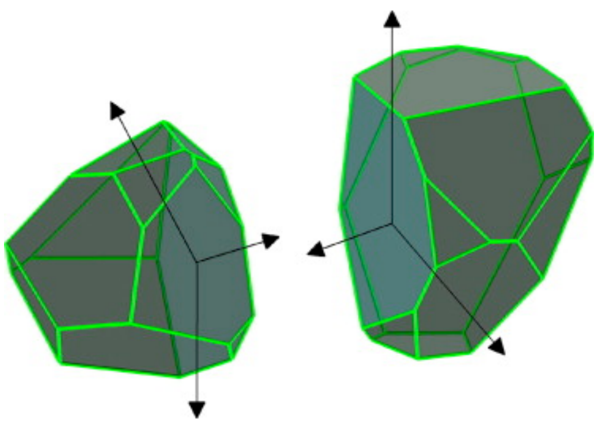




cuboctahedron



Pentagonal orthocupolarotunda

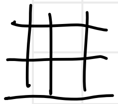


polycrystalline morphology

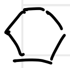
Lord Kelvin's Bubble Problem

Cells of equal volume with minimum surface area

in 2D

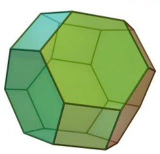


not

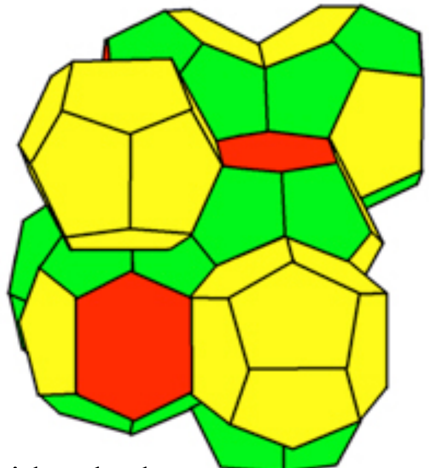


hex grid

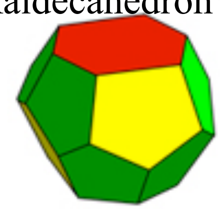
Kelvin structure, 1887
(truncated octahedra)



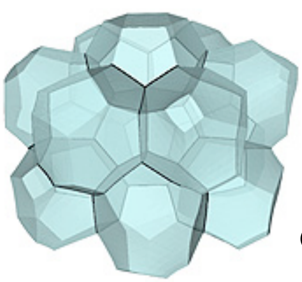
Weaire-Phelan structure, 1993
W https://en.wikipedia.org/wiki/Weaire–Phelan_structure



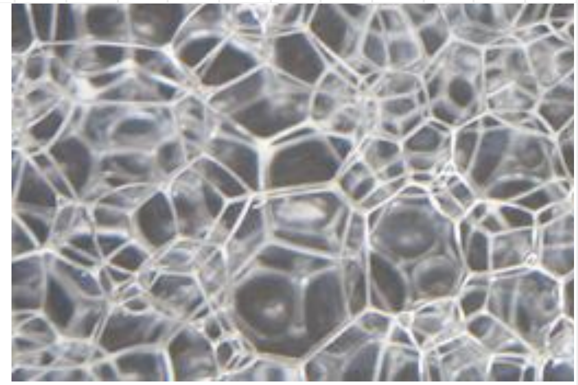
tetrakaidecahedron



irregular dodecahedron



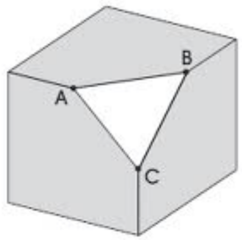
Gabrielli's structure, 2009

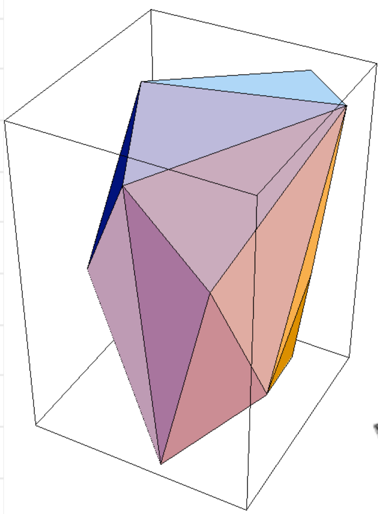


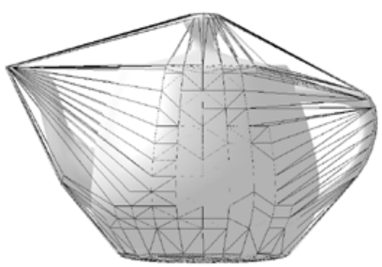
2008 Beijing Olympics

Non-convex Polyhedra

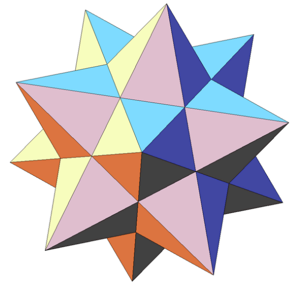
convex

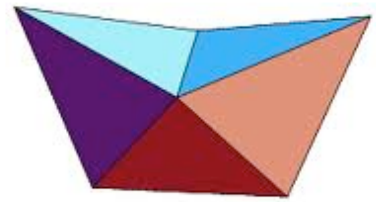


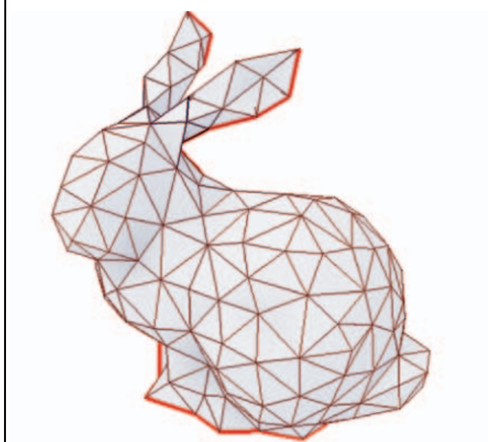




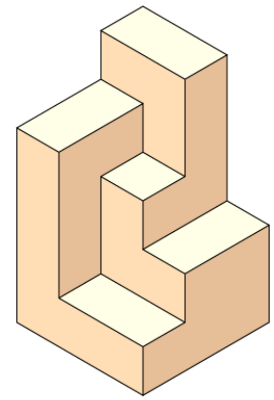
non-convex



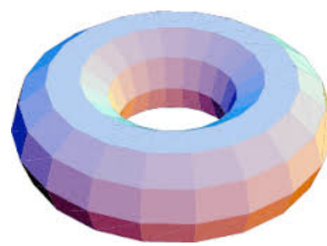


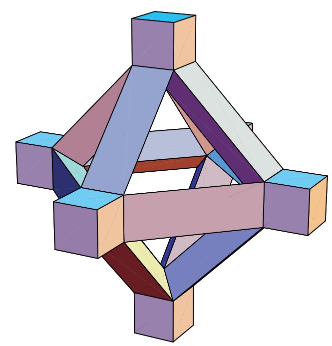


Tomohiro Tachi



David Eppstein



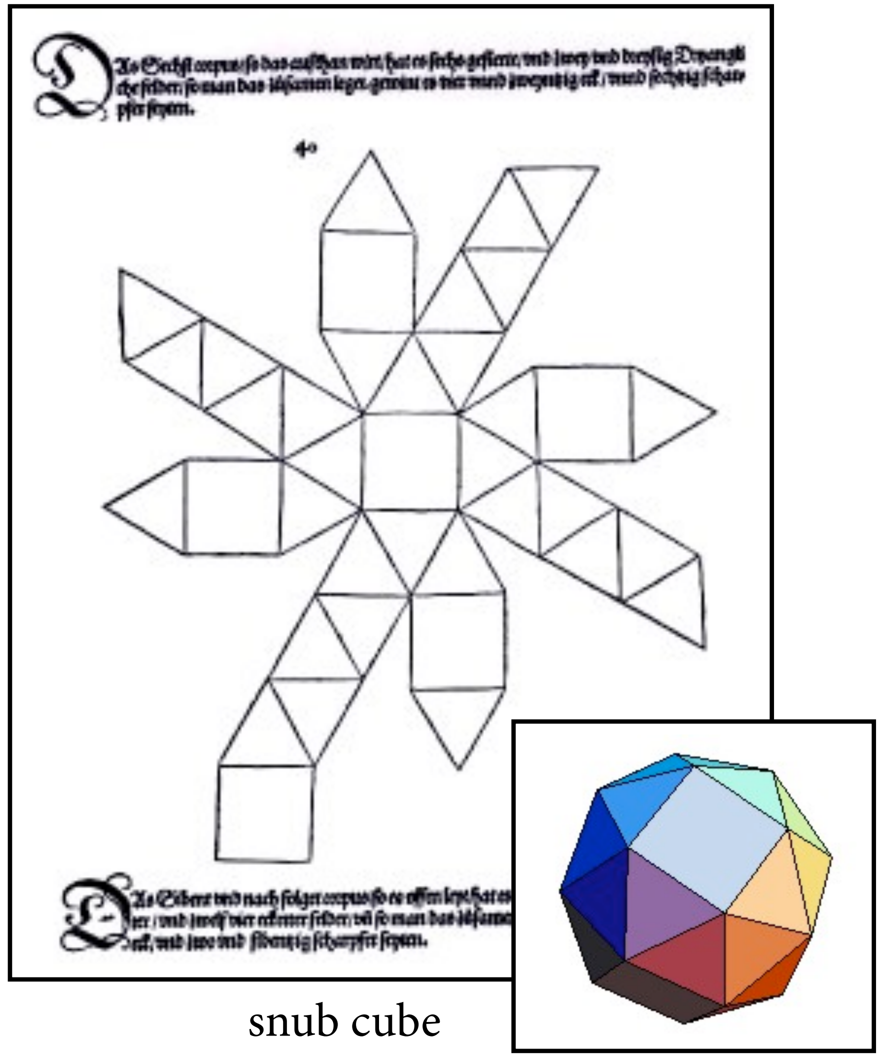


Donoso & O'Rourke

Maybe later in the course we will talk about unfolding polyhedra.

Unfolding Polyhedra—Durer 1400's

Durer, 1498



snub cube

A ***polyhedron*** consists of a finite connected set of (plane) polygons called ***faces*** such that

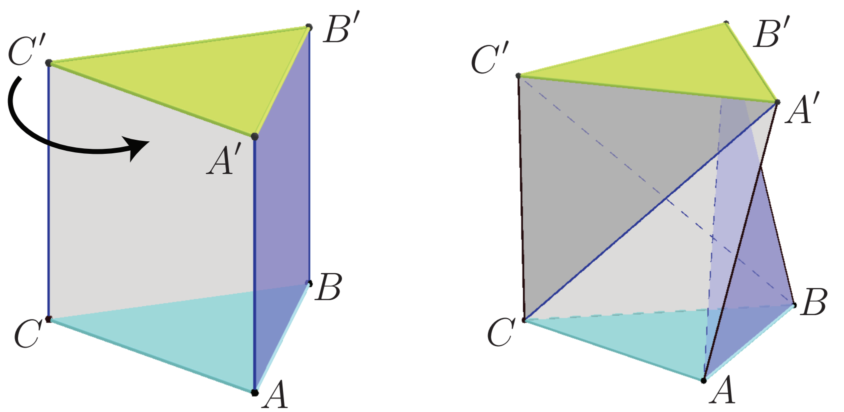
1. if two faces intersect it is only at a common vertex or edge
2. every edge of every face is an edge of exactly one other face
3. the faces surrounding each vertex form a single circuit

A ***tetrahedron*** is a polyhedron with 4 triangular faces. (aka a *simplex*)

To ***tetrahedralize*** a polyhedron means to partition its interior into disjoint tetrahedra whose vertices are vertices of the polyhedron.

Not all polyhedra can be tetrahedralized

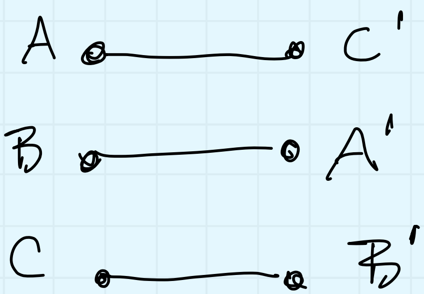
Schönhardt 1928



triangular prism with top face twisted, produces reflex edge in each rectangular face

why no tetrahedralization?

there is no single tetrahedron
each vertex rules out one other



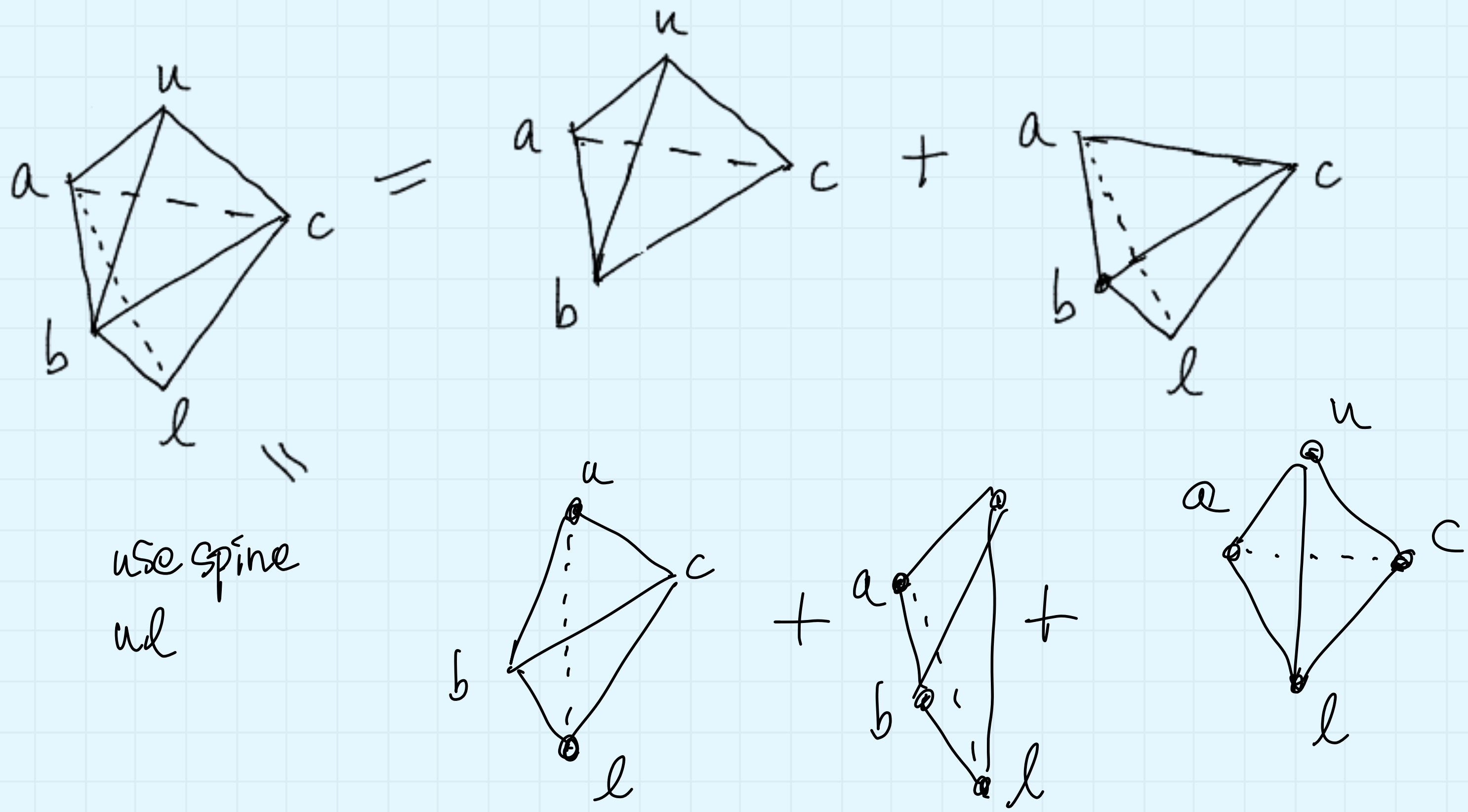
we need 4 "independent" vertices.
But no such set.

NP-hard to test if a polyhedron can be tetrahedralized:

On the difficulty of triangulating three-dimensional nonconvex polyhedra
J **Ruppert**, R **Seidel** - Discrete & Computational Geometry, 1992 - Springer
A number of different polyhedral decomposition problems have previously been studied, most notably the problem of triangulating a simple polygon. We are concerned with the polyhedron triangulation problem: decomposing a three-dimensional polyhedron into a set …
https://doi.org/10.1007/BF02187840

The number of tetrahedra in a tetrahedralization is not unique

Example:



Exercise: Show that a cube can be cut into 5 tetrahedra and into 6 tetrahedra.

There are examples where number of tetra. can be $2n-7$ or $\binom{n-2}{2}$

Using Steiner points to partition a polyhedron into tetrahedra

Note: the output is no longer combinatorial — we need coordinates for Steiner points

Determining the min number of Steiner points for a given polyhedron is NP-hard.

Determining the minimum number of tetrahedra for a given polyhedron is NP-hard.
Even for convex polyhedra! (where min. number of Steiner points is 0)

https://link.springer.com/content/pdf/10.1007/s004540010058.pdf
Minimal simplicial dissections and triangulations of convex 3-polytopes
A Below, U Brehm, JA De Loera… - Discrete & Computational …, 2000 - Springer

d https://doi.org/10.1016/S0196-6774(03)00092-0
The complexity of finding small triangulations of convex 3-polytopes
A Below, JA De Loera, J Richter-Gebert - Journal of Algorithms, 2004 - Elsevier

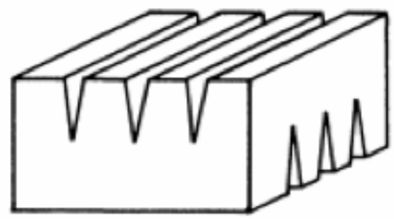
Can adding Steiner points reduce the number of tetrahedra? Yes.

Exercise: Find an example.

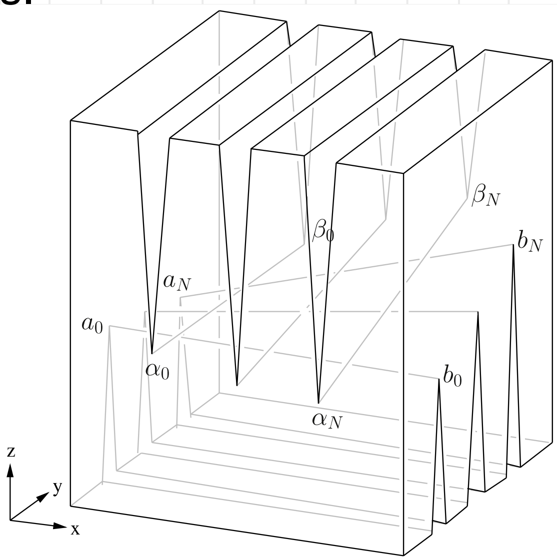
Explore efficient algorithms to approximate the min number of Steiner points or tetrahedra to within some guaranteed ratio.

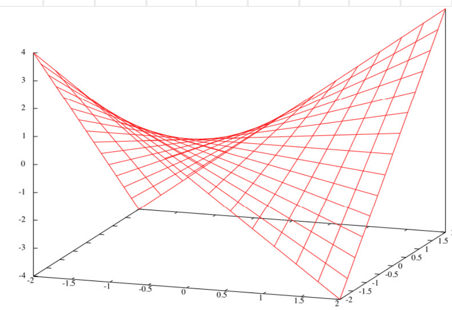
Using Steiner points to partition a polyhedron into tetrahedra

a lower bound:
There are polyhedra that require Omega(n^2) Steiner points even to partition into convex pieces. Chazelle, 1980's.



Chazelle





Hang Si & Nadja Goerigk

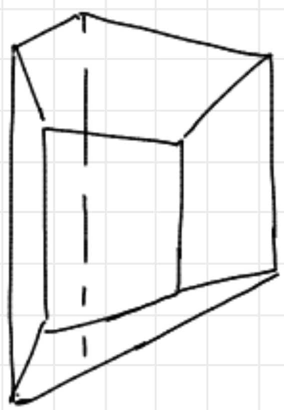
Cut wedges from a cube so they *almost* meet in the middle, and their lines form a hyperbolic paraboloid. The lines cut the hyperbolic paraboloid into Theta(n^2) pieces, pairwise invisible, so Omega(n^2) convex pieces are needed in any partition.

d https://doi.org/10.1137/0213031
Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm
B **Chazelle** - SIAM Journal on Computing, 1984 - SIAM
The problem of partitioning a polyhedron into a minimum number of convex pieces is known to be NP-hard. We establish here a quadratic lower bound on the complexity of this problem, and we describe an algorithm that produces a number of convex parts within a constant …

a positive result:
Any polyhedron can be partitioned into $O(n^2)$ tetrahedra using $O(n^2)$ Steiner points.

Bern and Eppstein, "Mesh generation and optimal triangulation", 1995

Idea — a bit like trapezoidization:
- from each edge of the polyhedron, extend a vertical wall up and down.
- pieces are "generalized prisms"



- vertical sides (each is a trapezoid)
- one top face, one bottom face
  (not necessarily parallel)

- this gives $O(n^2)$ pieces
- then tetrahedralize these pieces:
    - cut into triangular prisms by triangulating the top and bottom the same way
    - then add one Steiner point in each, making sure that tetrahedra match face-to-face

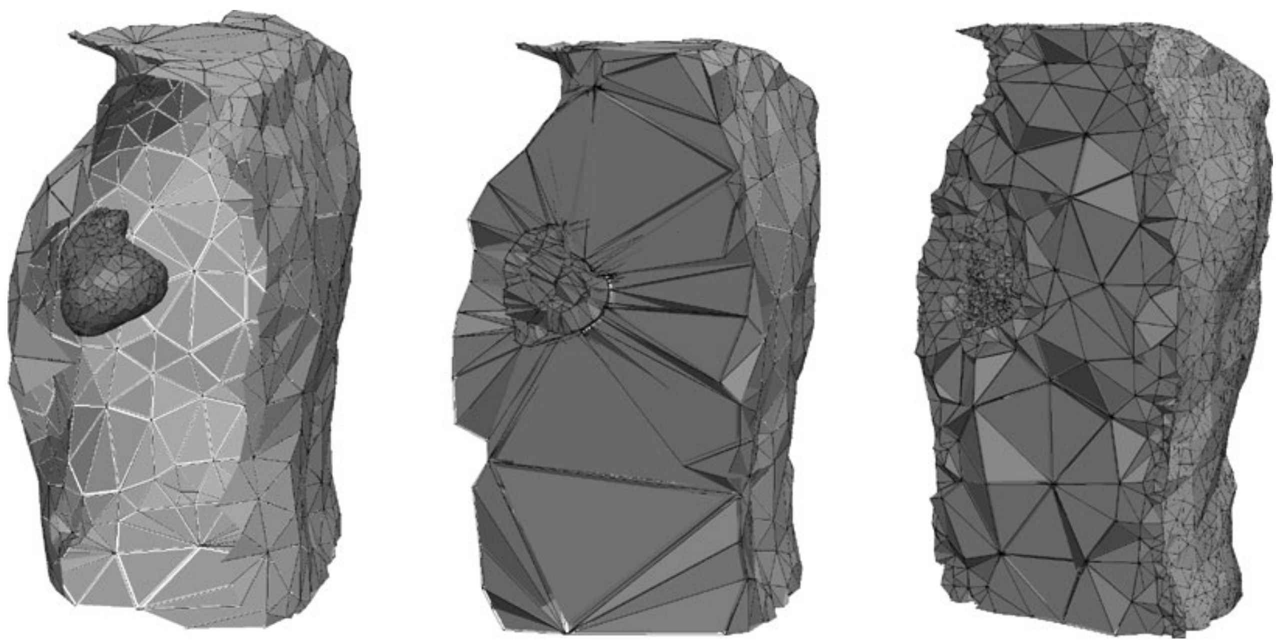
an approach from meshing (uses Delauany tetrehedralization, which we'll cover later on)

https://doi.org/10.1007/3-540-29090-7_9

Meshing piecewise linear complexes by constrained Delaunay tetrahedralizations

H Si, K Gärtner - Proceedings of the 14th international meshing …, 2005 - Springer

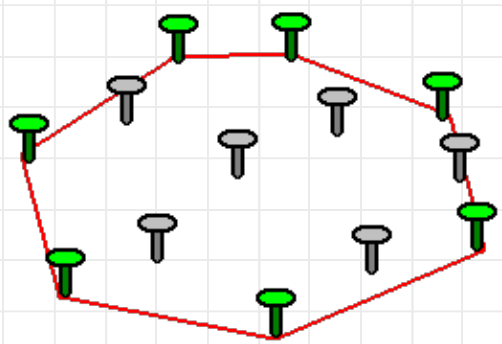
We present a method to decompose an arbitrary 3D piecewise linear complex (PLC) into a constrained Delaunay tetrahedralization (CDT).



NEW TOPIC: Convex Hulls

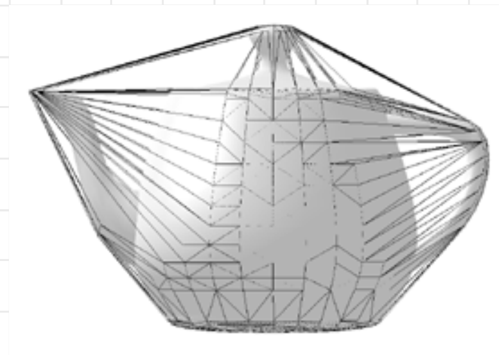
Given points in d-dimensional space, find a good “container” = convex polytope.
Many applications, e.g. collision detection, pattern recognition, motion planning . . .

In 2D, imagine putting a rubber band around the points



https://brilliant.org/wiki/convex-hull/

In 3D, wrap with shrink-wrap



Newton Collision Convex Hull

More formally:

$\subseteq S$

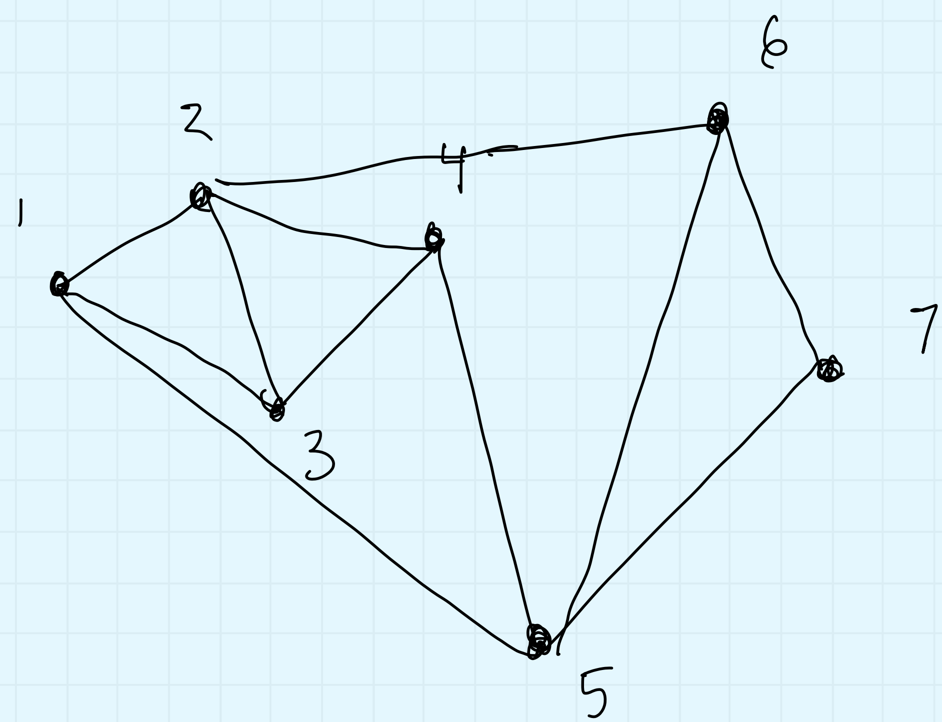
In 2D, the ***convex hull*** of a set of points S is a convex polygon P with vertices in S such that every point of S lies inside.
(definition in 3D and higher later on)

Convex Hull Algorithms in 2D

Almost any algorithmic paradigm will work, so this problem is a great one for Algorithms courses. See [Zurich notes, Chapter 4].

Incremental Algorithm — add points one by one in sorted order by x coordinate

Example

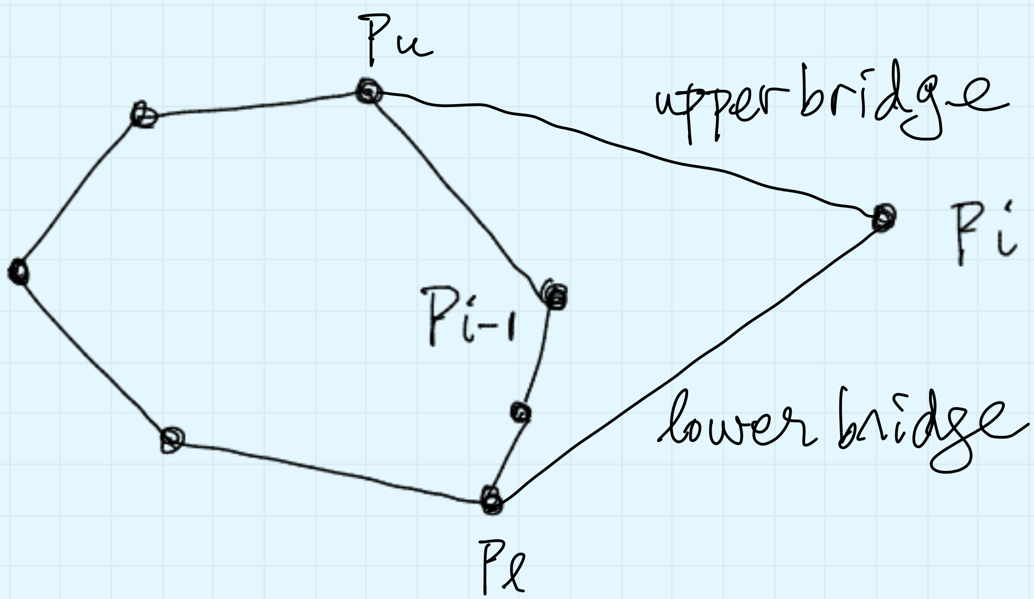


Incremental Algorithm — add points one by one in sorted order by x coordinate

general situation

We have:
- H_{i-1} = CH($p_1, \ldots, p_{i-1}$) as a doubly linked list
- p_{i-1} is a vertex of H_{i-1}

We want:
- add p_i to get H_i
- p_i is joined to:
  - p_u by *upper bridge*
  - p_l by *lower bridge*

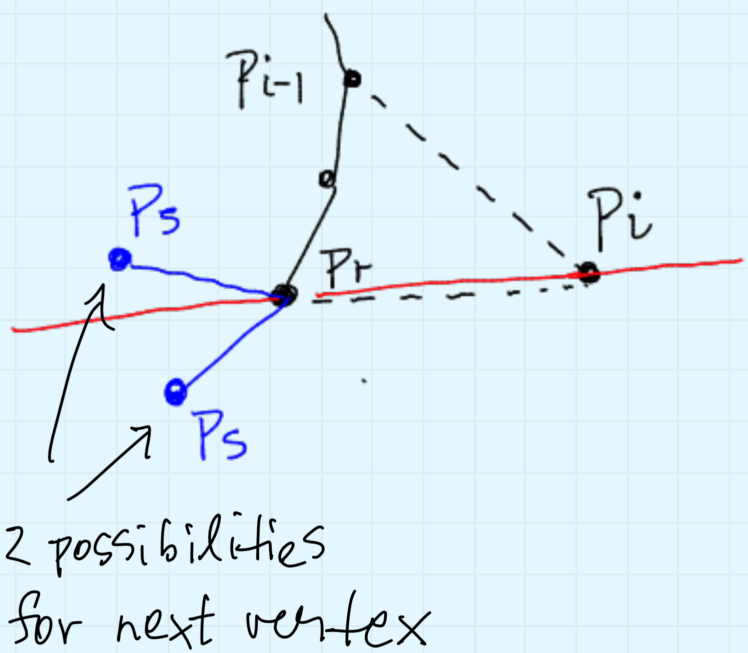


Incremental Algorithm — add points one by one in sorted order by x coordinate

- starting from p_{i-1} scan forward (clockwise) to find p_l
- starting from p_{i-1} scan backward (counterclockwise) to find p_u

invariant: the line segment from p_i to the current vertex is outside the CH, H_{i-1}
(true initially for line segment $p_i p_{i-1}$)

How to stop the scan



currently at p_r
next vertex is p_s

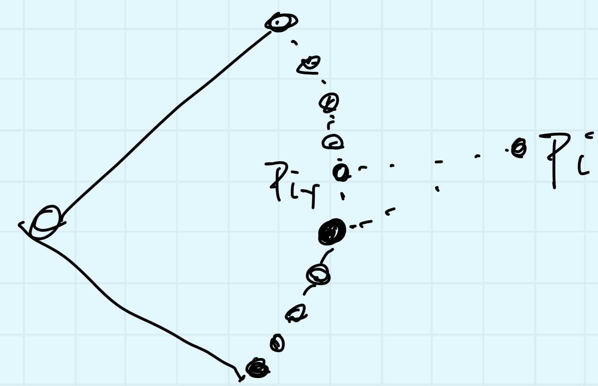
- if p_s is above line $p_i p_r$ then we have lower bridge $p_l \leftarrow p_r$
- else scan moves to p_s

Run time

Adding one point

can take $\Theta(n)$

? bound is $\Theta(n^2)$??



Amortized analysis

each input point is added once
and deleted at most once at $O(1)$ cost

So ~~total~~ is $O(n)$.

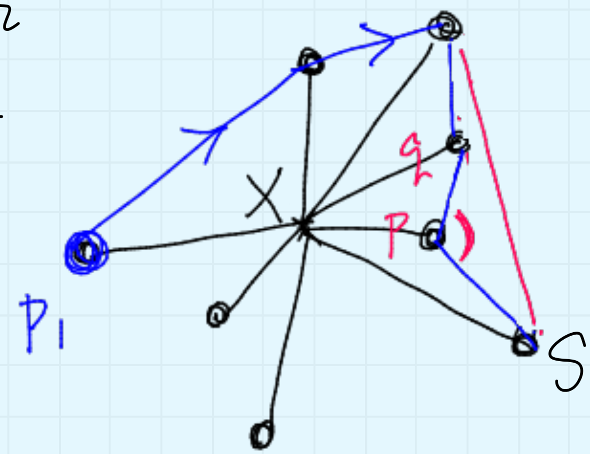
+ sort $O(n \log n)$

final total $O(n \log n)$.

Graham's Algorithm

Another sorting-base approach.
1. Sort the points radially around some point X inside the convex hull.

2. scan from p_1 in clockwise order
repeatedly remove 2nd last point if it forms a reflex angle.



remove p,
then remove q.
then go on from s.

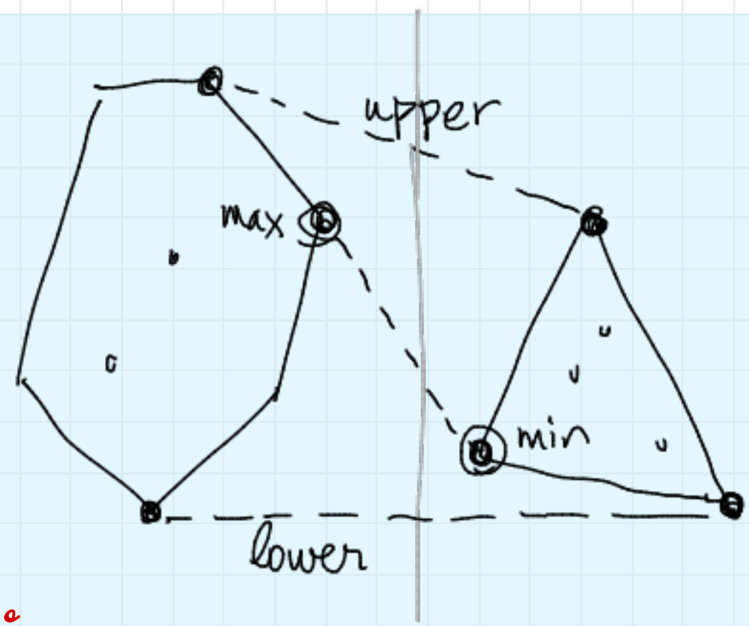
To find X: take average of any 3 non-collinear points.

To sort the points radially around X: use sidedness tests.

Runtime: $O(n \log n) + O(n)$
sort scan.

Divide and Conquer Algorithm

Divide the points in two by a vertical line (easy if we sort by x coordinate).
Recurse on each side.
Then combine the two sides.



To combine

Find upper & lower bridges
Start with segment from max x on left to min x on right
Walk up to get upper bridge
" down - - lower "
Time is O(# points that are removed)
(similar to incremental)

EX. Show that taking max y on left/right does not always give upper bridge.

Divide and Conquer Algorithm

Runtime

Combine step $O(n)$

Sort $O(n \log n)$

$$T(n) = 2T\left(\frac{n}{2}\right) + \underbrace{c \cdot n}_{\text{combine step}}$$

$$T(n) = O(n \log n)$$

(recall from merge sort OR prove by induction).

Lower Bound

There is an Omega(n log n) lower bound on computing the ordered convex hull in 2D on a RAM (Random Access Machine) with +,-,x.

i.e. the polygon

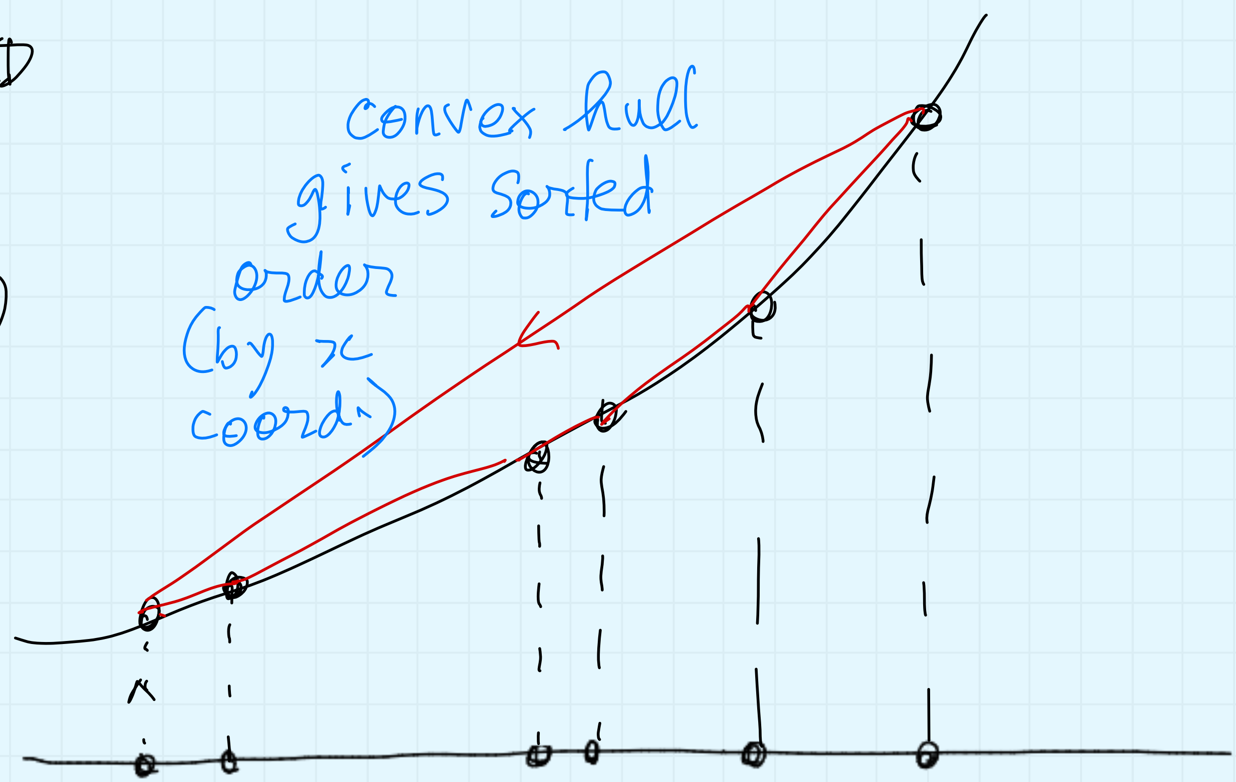
(recall sorting is $\Omega(n \log n)$ on that model).

Proof. Reduce sorting to finding the convex hull.

map points to 2D

map

$x \mapsto (x, x^2)$



input points we want to sort

Note: even finding the (unsorted) CH vertices takes n log n (needs different proof)

Summary

- partitioning polyhedra
- algorithms for convex hull in the plane

References

- [Handbook] Chapter 30 for partitioning

For convex hulls:

- [CGAA] Section 1.1
- [Zurich notes] Chapter 4
- [O'Rourke] Chapter 3