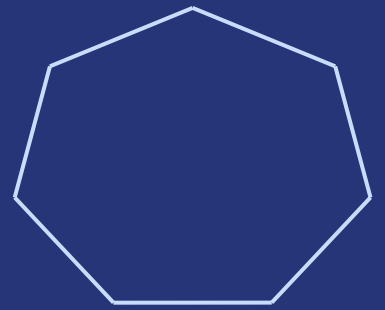
ADVANCED ALGORITHMS · FALL SEMESTER

Lecture 3



Geometric Algorithms: Convex Sets & Divide-and-Conquer

Faculty: K.R. Chowdhary

Professor of CS

Half-plane intersection, convexity, and the Jordan curve theorem

Computational Geometry

Computational geometry evolved from the classical design-and-analysis-of-algorithms discipline. It studies the computational complexity of geometric problems, exploiting underlying geometric structure — like metric space — to derive efficient algorithms.

Arises in

Pattern recognition, computer graphics, computer vision, robotics, VLSI layout, operations research, statistics.

Proximity	**Intersection**	**Searching**	**Point location**
Convex sets	**Convex hull**	**Voronoi diagrams**	**CAD/CAM**

This lecture focuses on convex sets and the divide-and-conquer approach to geometric problems.

The Common Intersection of n Half-Planes

Divide and conquer

Partition the problem into subproblems, recursively solve each, then combine the sub-solutions into the final answer.

The problem

Given a set S of n half-planes h_i, each represented by:

$$a_i x + b_i y \leq c_i, \quad i = 1, 2, \ldots, n$$

Example: h_1 is $5x + 4y \geq 20$ — shown at right, all points in the shaded half-plane.

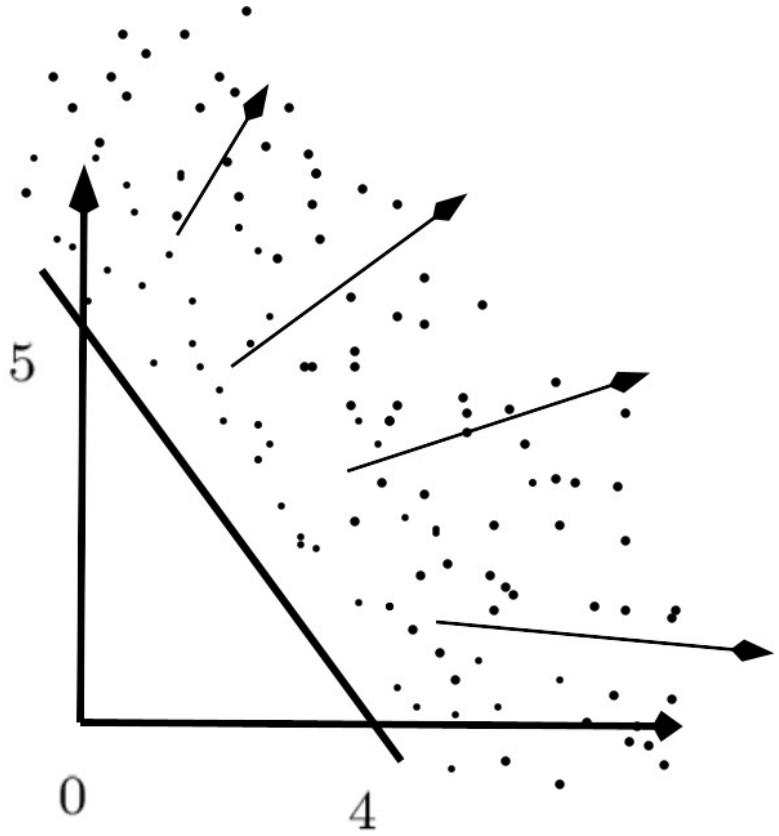


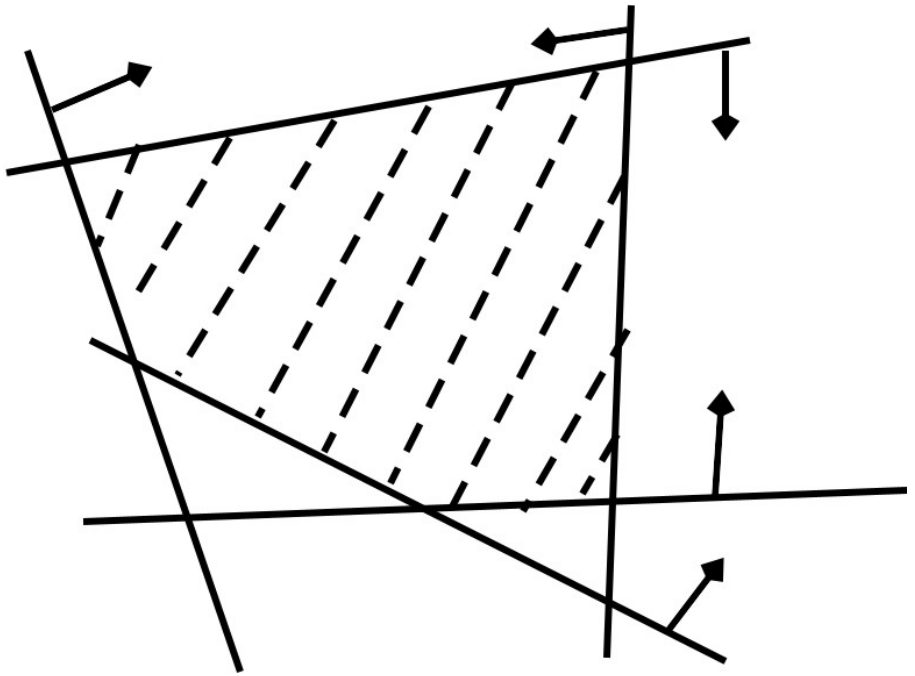
Figure: Half-plane $5x + 4y \geq 20$

The Algorithm

$C(S) = \bigcap_{i=1}^{n} h_i$ is a convex set (possibly unbounded); if bounded, it's a convex polygon.

ALGORITHM — COMMON-INTERSECTION-D&C(S)

```
if |S| ≤ 3:
   compute CI(S) explicitly
   return (CI(S))
divide S into ~equal S1, S2
CI(S1) = D&C(S1)
CI(S2) = D&C(S2)
CI(S) = Merge(CI(S1), CI(S2))
return (CI(S))
```



Half-planes are on the side of the arrows; shaded = intersection

Running Time of the D&C Algorithm

The key step is merging two common intersections. Since $CI(S_1)$ and $CI(S_2)$ are convex, this reduces to intersecting two convex polygons — solvable in time proportional to their size.

$$T(3) = O(1)$$

$$T(n) = 2T(n/2) + O(n) + M(n/2, n/2)$$

where $M(n/2, n/2) = O(n)$ is the merge time (step 5)

Total running time: $O(n \log n)$

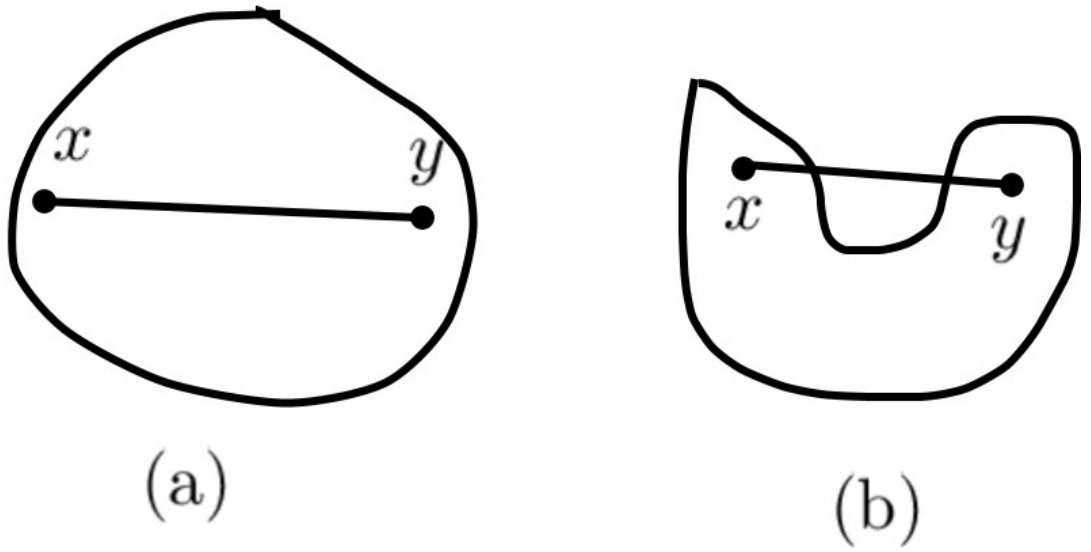
What Makes a Set Convex?

Definition — Convex Set

In Euclidean space, a region is convex if, for every pair of points within it, every point on the straight line segment joining them is also within the region.

A NOTE ON COMPUTATION

This geometric definition isn't directly computable — there are infinitely many point pairs to check. Efficient algorithms instead exploit other convexity properties.



(a) Convex set — segment xy stays inside. (b) Non-convex — segment xy leaves the region.

The Formal (Vector-Space) Definition

Let S be a vector space over $\mathbb{R}$. A set C in S is convex if, for all $p, q \in C$ and all $\alpha \in [0, 1]$:

$$(1 - \alpha)p + \alpha q \in C$$

i.e., every point on the segment connecting p and q is in C. A convex set is path-connected, hence connected.

Strictly convex

Every point on the segment other than the endpoints lies in the interior of C.

Convex — not its boundary

A cube in $\mathbb{R}^3$ is convex, but its boundary isn't (segment pq escapes the surface unless p, q share a face).

Lines, rays, segments

$(1-\alpha)p+\alpha q$ for all α gives a line; $\alpha \geq 0$ gives a ray from p through q; $0 \leq \alpha \leq 1$ gives the segment pq.

Familiar Examples

In $\mathbb{R}$ (the reals)

Convex subsets are simply the intervals of $\mathbb{R}$.

In the Euclidean plane

Solid regular polygons, solid triangles, and intersections of solid triangles.

In 3-D Euclidean space

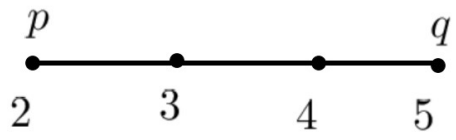
Archimedean solids and Platonic solids are convex.

Non-convex examples

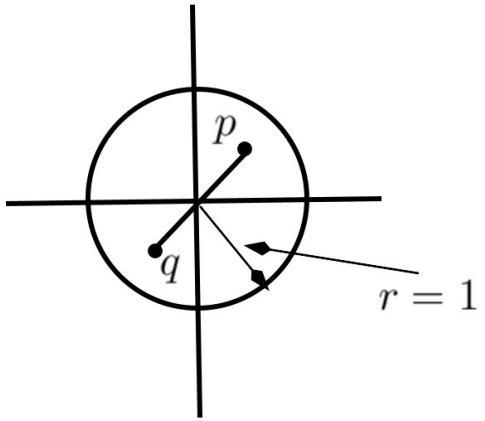
The Kepler–Poinsot polyhedra are classic non-convex sets.

Convexity theory matters because convex sets arise across many areas of mathematics — even infinite-dimensional theory leans on 2-D and 3-D reasoning.

Verifying Convexity Directly



(a)



(b)

(a) Line segment

Points 2, 3, 4, 5 form set C. With α=0.5, p=2, q=5: $(1-0.5)\cdot 2 + 0.5\cdot 5 = 3.5 \in C$. ✓

(b) Unit circle

p=(0.5, 0.75), q=(−0.4, −0.8), α=0.6. $(1-0.6)p + 0.6q = (-0.04, -0.18) \in C$. ✓

Convex Functions

Definition

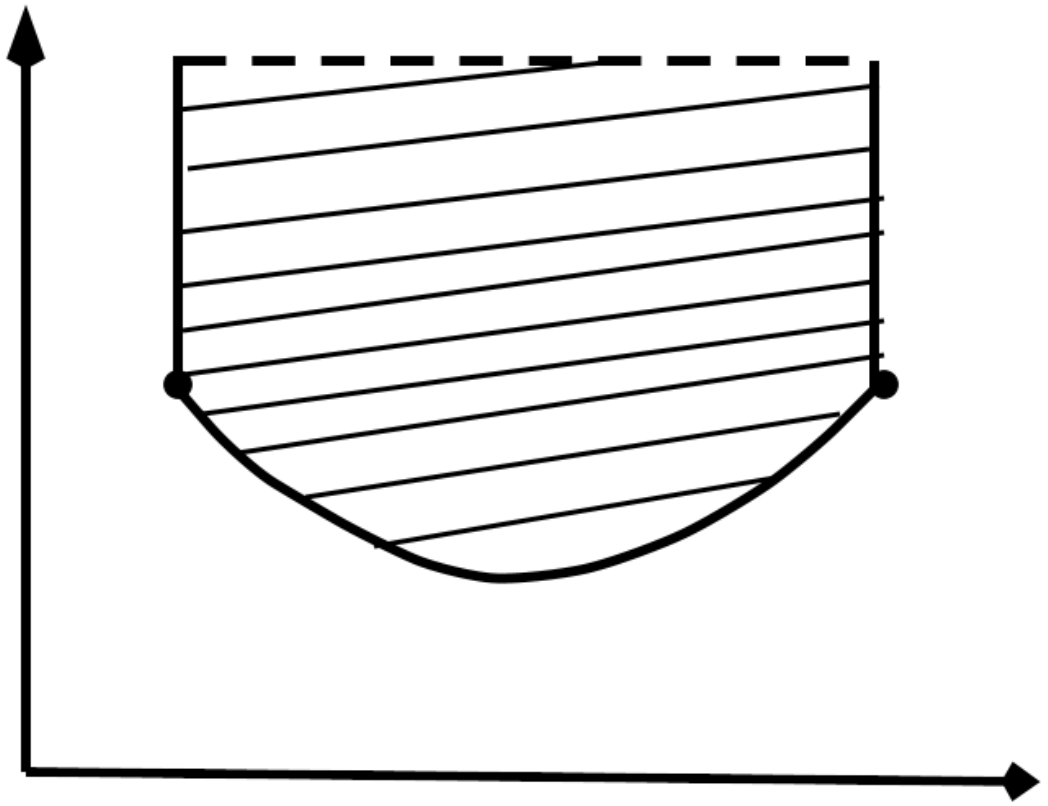
A function is convex if and only if its epigraph — the region above its graph — is a convex set.

PROPERTIES OF CONVEX SETS

If S is convex, for any r vectors $u_1,\ldots,u_r \in S$ and nonnegative $\lambda_1,\ldots,\lambda_r$ with $\Sigma\lambda_k = 1$:

$$\sum_{k=1}^{r} \lambda_k u_k \in S$$

This is called a convex combination of $u_1,\ldots,u_r$.

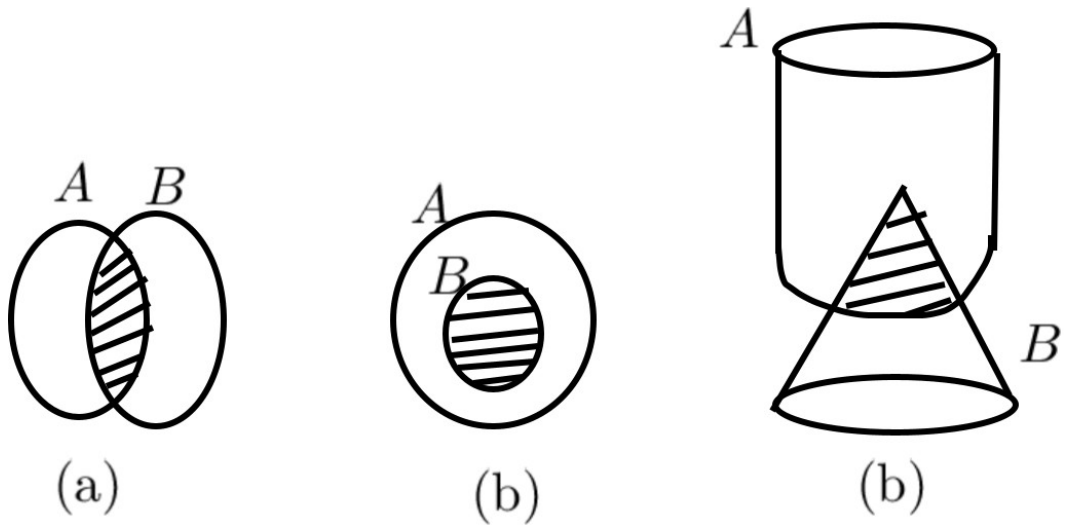


The epigraph (hashed region above the curve) is convex

Intersections and Unions

The collection of convex subsets of a vector space has these properties:

- The empty set and the whole vector space are convex.
- The intersection of any collection of convex sets is convex.
- The union of a non-decreasing sequence of convex sets is convex — but the union of two arbitrary convex sets need not be.

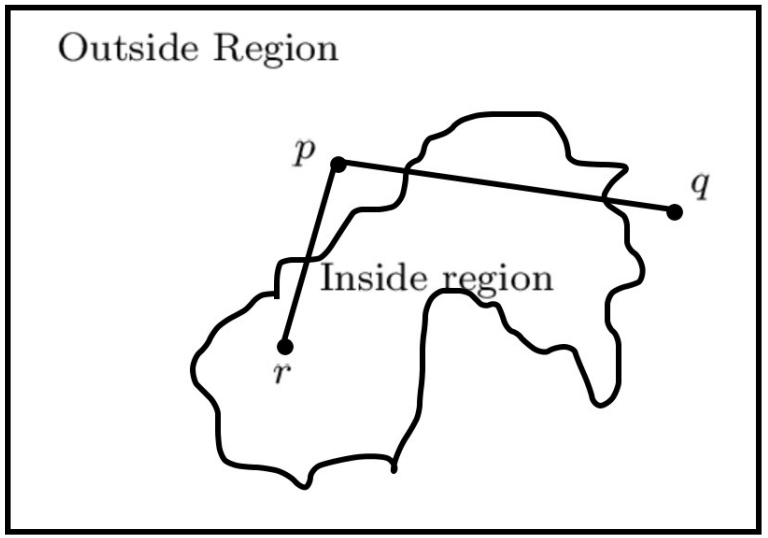


Intersections of convex sets A and B: (a) 2-D, (b) 2-D nested, (c) 3-D — all convex

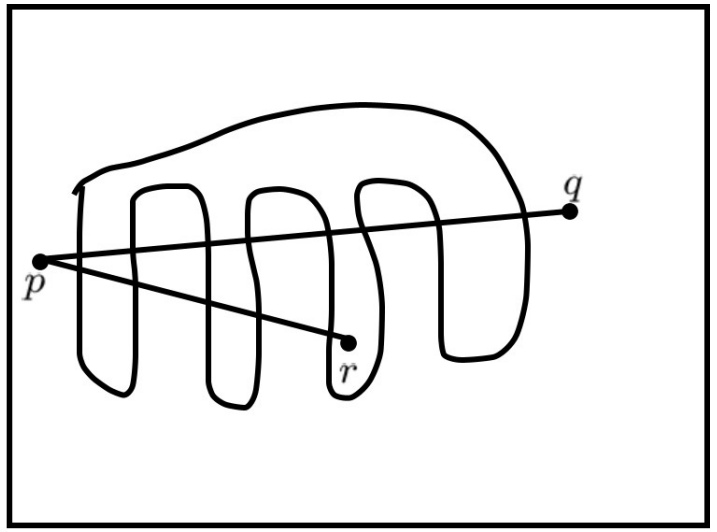
The Jordan Curve Theorem

Definition — Jordan curve

A non-self-intersecting continuous loop in the plane. The theorem asserts it divides the plane into an "interior" and an "exterior" region; any path between the two regions must cross the curve.



(a)



(b)

A path between two points outside the curve crosses it an even number of times; a path between an inside and an outside point crosses it an odd number of times.

APPLICATION

Is a Polygon Convex?

Given a simple closed Jordan polygon curve, determine whether its interior region is convex.

Observation

If the segment $\ddot{u}_i v_j$ (for every pair of vertices, $i \neq j$, $1 \leq i, j \leq n$) lies totally inside the region, the region is convex.

NAIVE COMPLEXITY

$$O(n^3)$$

$O(n^2)$ segments; testing each against every polygon segment takes $O(n)$ — $O(n^2) \times O(n) = O(n^3)$.

Convexity condition on angles

Each interior vertex angle must be strictly less than π. An angle greater than π is called reflex. Every vertex being non-reflex is necessary for the region to be convex.

APPLICATION

Is a Point Inside the Polygon?

Given a point (x, y) and a polygon P (as a vertex sequence), is (x, y) inside P, on the boundary, or outside?

Ray-casting idea

Draw a ray from (x, y) in any fixed direction and count how many polygon edges it crosses. An odd count means the point is inside; an even count means outside.

ALGORITHM — IS-POINT-INSIDE-POLYGON(polygon, point)

```
for each line segment of the polygon:
    if ray down from (x,y) crosses segment:
        crossings++
if crossings is odd: return (inside)
else: return (outside)
```

References

[1] **Mikhail J. Atallah & Marina Blanta,**
"Algorithms and Theory of Computation Handbook, Special Topics and Techniques, 2nd ed." — CRC Press, 2010

[2] **Allen B. Tucker, Jr.,**
"The Computer Science and Engineering Handbook" — CRC Press, 1997

Disclaimer: these notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the faculty.