

Last Time: PSLGs

Topology of embedded planar graphs { Jordan Curve Thm (for Polygons)
Euler's Formula

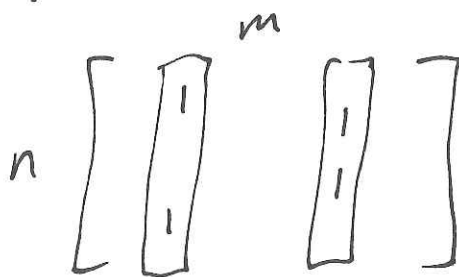
Today: Data Structures for PSLGs

Goal: store a PSLG that supports "walking around", i.e. incidence operations.

Warmup: Let's start in 1D \Rightarrow Incidence Operations { (i) edges incident to a vertex.
(ii) vertices incident to an edge



Idea 1: Edge-Vertex Incidence Matrix



Size: $O(nm)$

Operations: $O(n)$ & $O(m)$ } Big + Slow

Advantage: Works to store more general graphs.

However: Euler's Formula \Rightarrow planar graphs are sparse
(avg degree is constant)

Idea 2: Just store neighbors (nbrs).

- ≤ 2 edges per vertex
- ≤ 2 vertices per edge

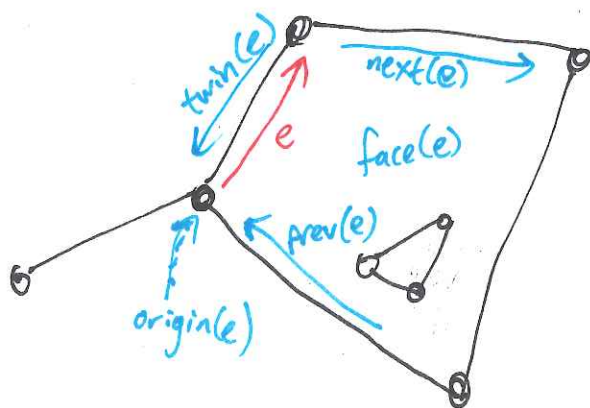


Starts to look like a doubly linked list.

Add a bit of geometry: Distinguish left and right.

Let's do this in the plane.

Doubly-Connected Edge List (A.K.A. Half-edge Data structure)



One Edge \rightsquigarrow Two Half-edges
(one in each direction)

Store V, E, F

$v \in V : e \in E \text{ s.t. } v = \text{origin}(e)$

$f \in F : e \in E \text{ s.t. } f = \text{face}(e)$
(one for each bdy component)

$e \in E : \text{origin}, \text{face}, \text{twin}, \text{next}, \text{previous}$

Convention: $\text{face}(e)$ is on the right.

Incidence Operations:

I. List edges incident to a vertex v (in ccw order)

Let $e = \text{halfedge}(v)$.

Let $e' = e$

Let $\text{output} = \emptyset$

Do

$\text{output} \ll e'$

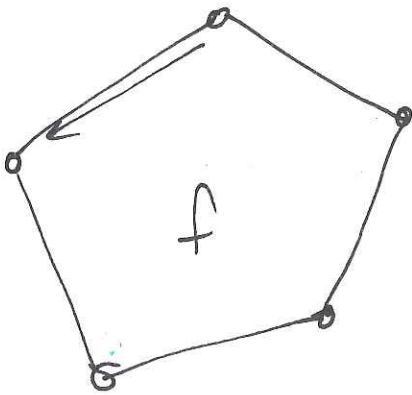
let $e' = \text{twin}(\text{prev}(e'))$

Loop while ($e' \neq e$)

return output.

II List edges in bdy of a face.

First consider a face with **No Holes.**



Let $e = \text{he}(f)$

Let $e' = e$

Let $\text{output} = \emptyset$

Do

$\text{output} \ll e'$

let $e' = \text{next}(e')$

Loop while ($e' \neq e$)

return output

If there are holes, repeat for each hole.

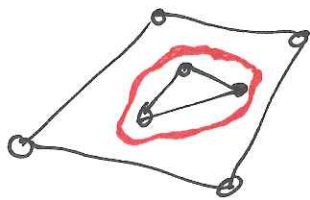
Restricting Ourselves to Graphs with some minimum Connectivity.

Def A graph G is k -connected (a.k.a. k -vertex connected) if G has $> k$ vertices and $\underbrace{G-X}$ is connected (induced subgraph) for every set X of $< k$ vertices.

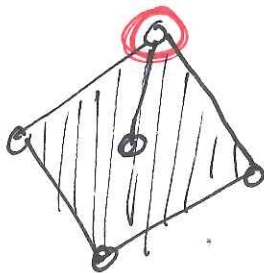
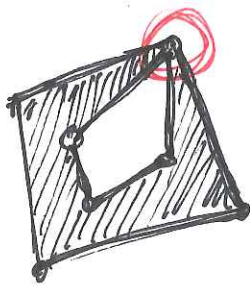
ex) 1-connected = connected (i.e. there is a path between every pair)

2-connected also called biconnected
 k -connected \Rightarrow $(k-1)$ -connected

For connected graphs, we only need to store one half-edge for each face because no faces have "holes."



For ~~2~~ 2-connected graphs, All bdy's are simple polygons



Posets and Duality

Incidence puts a partial order on edges, vertices, + faces.

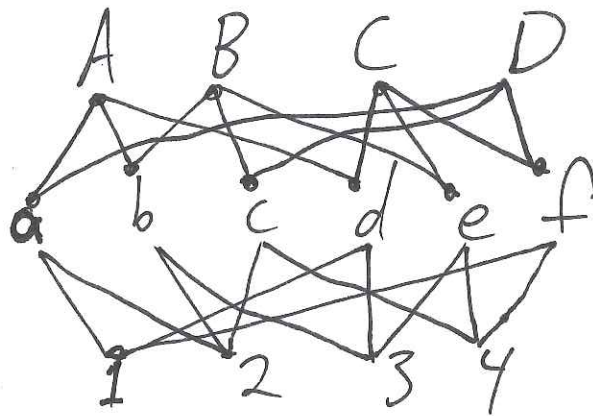
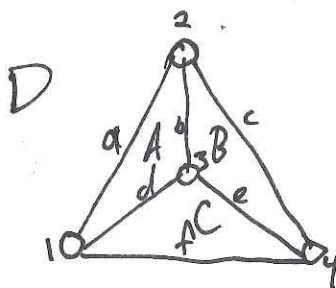
$$v \prec e \quad \text{iff} \quad v \in e$$

vertex edge

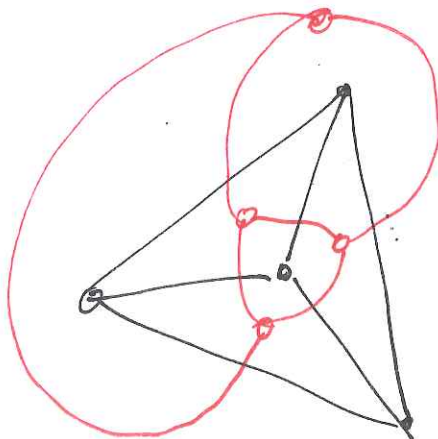
$$e \prec f \quad \text{iff} \quad e \in \text{bdy}(f)$$

edge face

The Poset (partially ordered set) or Hasse Diagram illustrates all the incidence relations.



The Dual of a PSLG has $\begin{cases} \text{a vertex for each face} \\ \text{a face for each vertex} \\ \text{an edge for each edge} \end{cases}$



If the graph is 2-connected, we get the dual by turning the POSET upside-down.

Why is it useful to look only at this case?

Famous Duals

