6.885	Lecture 3	Sept. 12, 2007

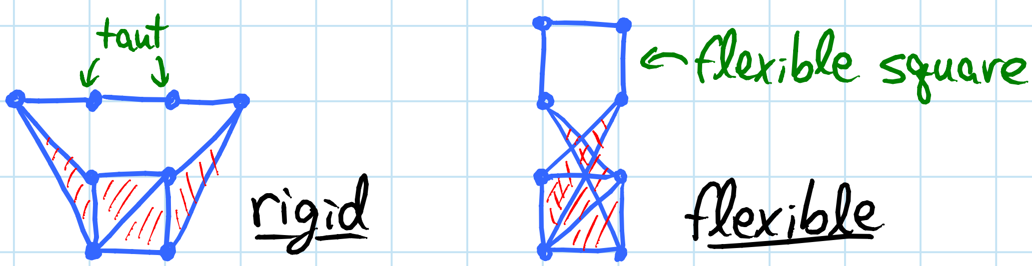
Review: (but without pinning)

Linkage = graph + lengths of edges $\ell : E \to \mathbb{R}^{>0}$

Configuration of linkage = coordinates for vertices $C : V \to \mathbb{R}^d$ satisfying edge lengths

Rigidity: a linkage configuration is
- *flexible* if there's a nontrivial motion starting there
  - → positive duration
  - → not a rigid motion (translation + rotation)
- *rigid* otherwise (Ø D.O.F. / isolated point)

Rigidity depends on configuration, not just linkage:



Conjecture: testing rigidity is coNP-hard
- by Kapovich & Millson [2002] universality, equivalent to testing whether a given point is isolated in a given algebraic set
- this problem is coNP-hard [Koiran 2000, Thm. 6]
- need to check reduction is efficiently computable (should be)

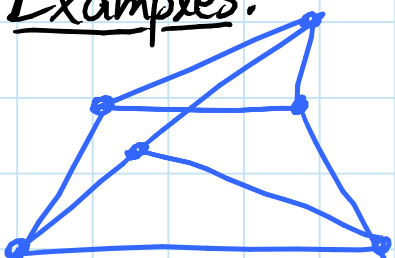
<u>Generic rigidity</u>: rigidity is usually a property of the graph, not the configuration or linkage

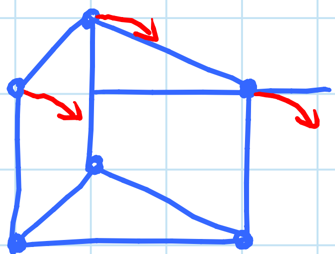
<u>Realization</u> of a graph
= coordinates for vertices $C: V \to \mathbb{R}^d$ (= pt. in $\mathbb{R}^{dn}$)
= configuration of a linkage of the graph

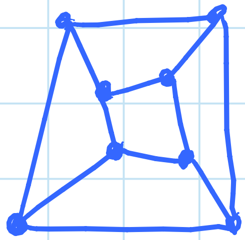
<u>Generic</u> realization avoids any "degeneracies":
e.g. { — 3 points collinear
— 4 points concyclic, or inducing parallel segs. }
another definition in L4 { — any <u>nontrivial</u> rational algebraic equation }
↳ not universally true — depends on input
→ lower-dimensional subset of $\mathbb{R}^{dn}$
⇒ almost every realization is generic
i.e. random $\underbrace{\text{realization}}_{\text{or } \varepsilon\text{-perterbation}}$ is generic with probability 1

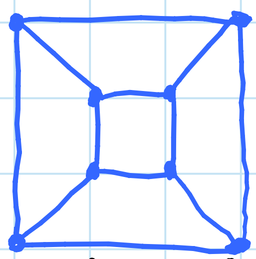
<u>Rigidity is generic</u>: for any graph, either
— all generic realizations are rigid
⇒ "<u>generically rigid</u>"
or — all generic realizations are flexible
⇒ "<u>generically flexible</u>"

<u>Examples</u>:


gen. rigid


rarely flex.


gen. flex.


rarely rigid

Minimally generically rigid graph = gen. rigid
+ removing any edge ⇒ gen. flexible
- characterize these first
- then gen. rigid graphs = supergraphs of these

Degree-of-freedom analysis: min. gen. rigid graph in 2D has exactly $2n-3$ edges (n vertices) & in dD has exactly $dn - \frac{d(d+1)}{2}$ edges

Proof: Empty graph has dn degrees of freedom.
Rigid graph has $\frac{d(d+1)}{2}$ degrees of freedom for rigid motion (2 transl. + 1 rotation in 2D)
Each edge imposes one $\underbrace{\text{algebraic constraint}}_{=(dn-1)\text{-dim. algebraic surface}}$, either:
- algebraically dependent: (redundant edge)
always holds given previous constraints
⇒ configuration space unchanged

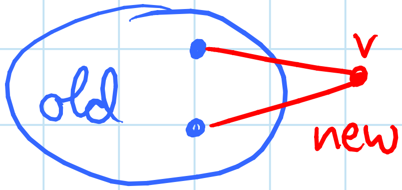
or - algebraically independent:
almost never holds (only if degenerate)
⇒ decreases dimension by 1

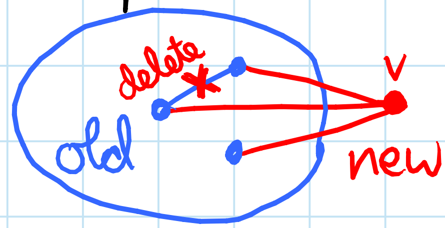
(in degenerate cases, can reduce dimension by 0 or more than 1: surfaces meeting degenerately in fewer dimensions)

∎

Henneberg characterization: [Henneberg 1911]

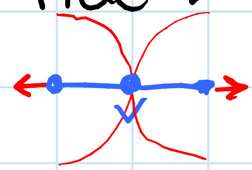
a graph is minimally generically rigid in 2D $\Leftrightarrow$
it can be built from  using two operations:

Type I: 

Type II: 

Proof: ($\Leftarrow$) holds in any dimension ~ degrees d & $d+1$

① G gen. rigid $\Leftrightarrow$ G' resulting from Type I is

($\Leftarrow$) if G is gen. flexible
then v can "come for the ride":
generically v won't be taut: 

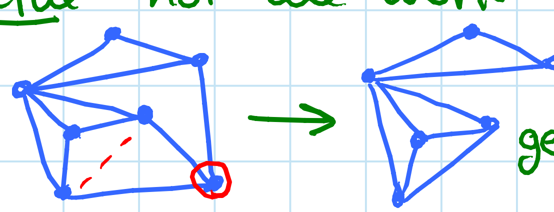
($\Rightarrow$) if G' is gen. flexible
then so is G: just ignore v's motion
Motion is nontrivial: can't just move v

② G min. gen. rigid $\Leftrightarrow$ G' from Type I is
– similar: prove $G-e$ flexible for all e
$\Leftrightarrow$ $G'-e$ flexible for all e

③ G min. gen. rigid $\Rightarrow$ G' from Type II is
– harder: need infinitesimal rigidity [L4]

④ G' min. gen. rigid with degree-3 vertex v
$\Rightarrow$ G' is result of Type II of <u>some</u> min. gen. rigid G
– ditto

Careful: not all work! [Abbott]

gen. rigid  gen. flexible

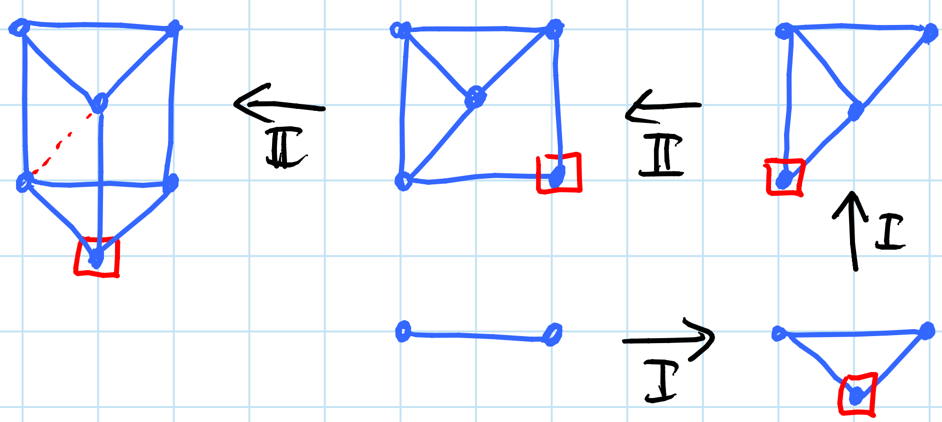
Proof of Henneberg characterization: (cont'd)

($\Rightarrow$) holds only in 2D
- DOF analysis $\Rightarrow$ $2n-3$ edges
- Handshaking Lemma: in any graph, $\sum$ vertex degrees $= 2$(# edges)

$\Rightarrow$ average degree $= 4 - \frac{6}{n} < 4$

$\Rightarrow$ some vertex v of degree < 4
- can't be degree 1 or else flexible
- if degree 2: delete v (anti-Type I op.)
  2 $\Rightarrow$ still min. gen. rigid
  induct
- if degree 3: 4 $\Rightarrow$ anti-type II op.
  induct

Example:



Laman's characterization: [Laman 1970]

a graph is minimally generically rigid in 2D $\Leftrightarrow$
it has $2n-3$ edges ($n>1$ vertices)
& every subset of k vertices induces $\le 2k-3$ edges

Proof: [based on Jack Graver's *Counting on Frameworks*]

($\Rightarrow$) Consider Henneberg construction.
Each operation adds one vertex ($n \nearrow 1$),
increases # edges by 2, and
any subset of vertices either
- excludes v $\Rightarrow$ # edges $\searrow$ 0 or 1

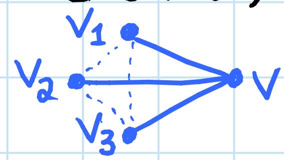
or - includes v $\Rightarrow$ $k \nearrow 1$, # edges $\nearrow \le 2$

Base case: 

($\Leftarrow$) $2n-3$ edges, so again
there is a vertex v of degree <4
- can't be degree 1: then remaining $n-1$ vertices induce $2n-4$ edges
- if degree 2:
  - delete v (anti-Type I op.)
  - every subset of remaining vertices induces same # edges as before

  $\Rightarrow$ still Laman
  - induct & use 2

Proof of Laman characterization: (cont'd)



($\Leftarrow$) – if degree 3:

- some edge is missing among v's neighbors: 4 vertices induce $\leq 2\cdot 4-3=5$ out of 6 edges
- need to find anti-Type II op. preserving Laman
- any k vertices including v_1, v_2, v_3 & not v must have $< 2k-3$ edges ($\Rightarrow$ room for 1 more): adding v increases $k \nearrow 1$, #edges $\nearrow 3$
- suppose there are k_i vertices S_i ($i=1,2,3$) containing v_{i+1}, v_{i+2} but not v & inducing $2k_i-3$ edges
- Inclusion-Exclusion principle:

$$\begin{aligned} E(S_1 \cup S_2) &\geq E(S_1)+E(S_2)-E(S_1 \cap S_2) \\ &= 2(k_1+k_2)-6-E(\underbrace{S_1 \cap S_2}_{\text{suppose } >1 \text{ vertex}}) \\ &\geq 2(k_1+k_2-|S_1 \cap S_2|)-3 \\ &= 2|S_1 \cup S_2|-3 \end{aligned}$$

but $S_1 \cup S_2$ contains v_1, v_2, v_3 & not v $\Rightarrow$ impossible

$\Rightarrow S_1 \cap S_2 = \{v_3\}, \ S_2 \cap S_3 = \{v_1\}, \ S_1 \cap S_3 = \{v_2\}$

$\Rightarrow |S_1 \cup S_2 \cup S_3| = k_1+k_2+k_3-3$

& $$\begin{aligned} E(S_1 \cup S_2 \cup S_3) &\geq E(S_1)+E(S_2)+E(S_3) \\ &= 2(k_1+k_2+k_3)-9 \\ &= 2|S_1+S_2+S_3|-3 \end{aligned}$$

but $S_1 \cup S_2 \cup S_3$ contains v_1, v_2, v_3 & not v

$\Rightarrow$ can add some edge among v_1, v_2, v_3 & delete v & preserve Laman; induct. $\square$

O(n m) algorithm: (n vertices, m edges)

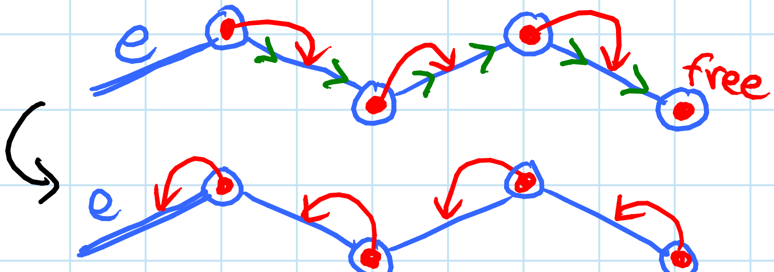
[Jacobs & Hendrickson 1997]

Edge set E is <u>independent</u> if, for each edge e & for each subset of k vertices, # induced edges + 3 copies of e $\leq 2k$.

<u>Equivalent formulation of Laman:</u> a graph is generically rigid $\Leftrightarrow$ has $2n-3$ independent edges (forming minimally generically rigid subgraph)

<u>Pebble algorithm:</u>

- repeatedly try to add edge to independent set (outer loop of m iterations)
- give each vertex 2 pebbles, assignable to any 2 incident edges
- to see whether edge e can be added to indep. set:
  - add 4 copies of e ~ unpebbled
  - try to get a pebble on every edge by searching for directed path from e to free pebble:



<u>directed graph</u> connectivity — O(n) time

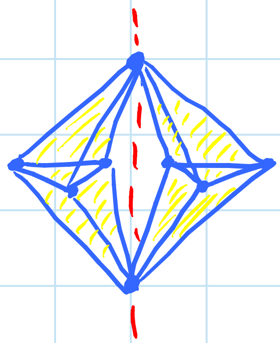
  - succeed precisely if e is independent
  - otherwise e is <u>redundant</u>: discard

OPEN: fastest 2D generic rigidity test?

- best so far: $O(n^2)$ [Hendrickson 1992]
- similar to above, using bipartite matching
- can we use faster matching algorithms? ($O(m\sqrt{n})$)
- can we use dynamic connectivity data structures?

OPEN: combinatorial test for 3D generic rigidity

- Laman condition ($3n-6$) still neccessary but insufficient:



double bananas

PROJECT: implement Laman and/or Henneberg

- color code over/underbracing, min. rigid subgraph
- show Henneberg construction

PROJECT: Henneberg computer game

- player starts from •—• & tries to build target min. gen. rigid graph by Henneberg ops.
- fairly easy backwards; hard forwards!