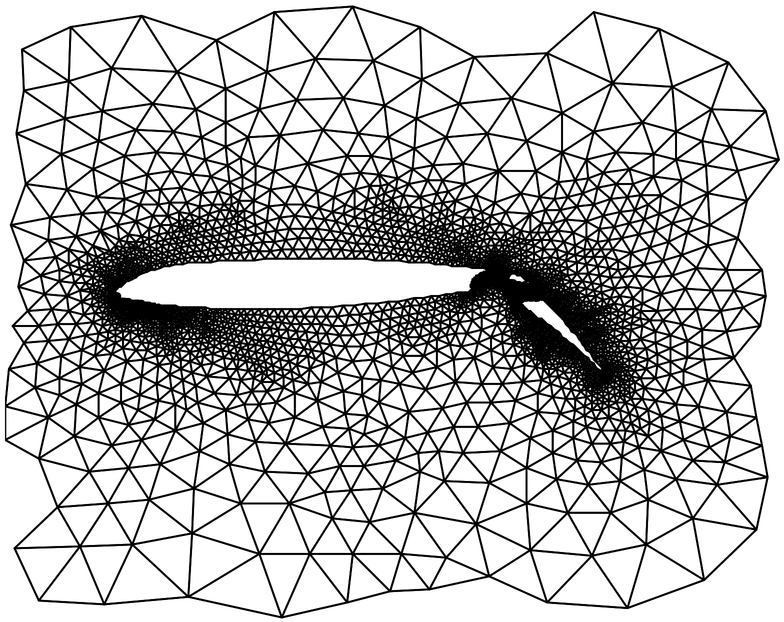
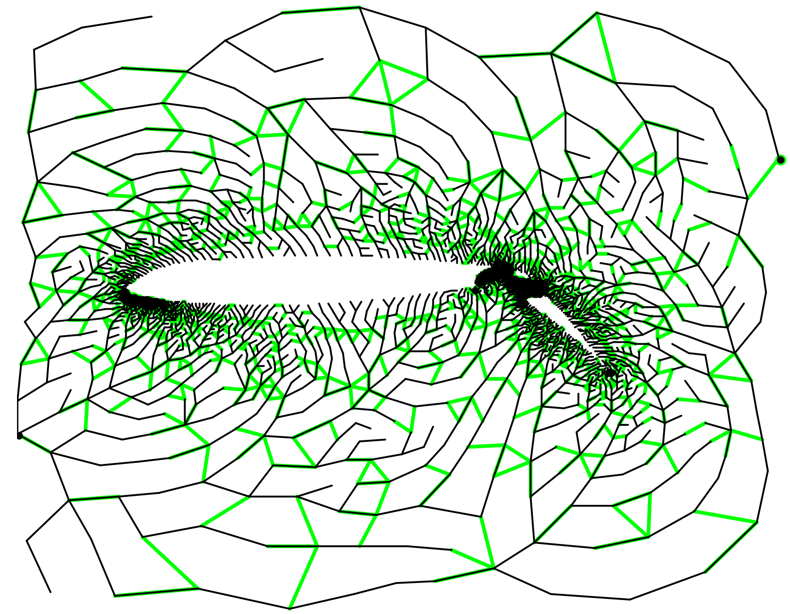
Algorithms, Graph Theory, and Linear Equations in Laplacians





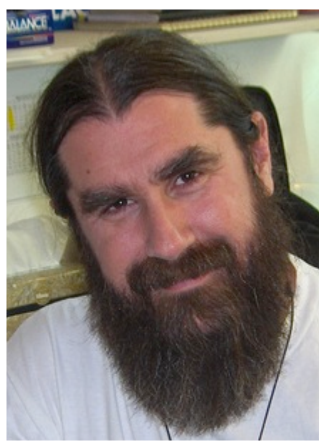
Daniel A. Spielman
Yale University

Shang-Hua Teng
滕 尚 華

Carter Fussell
TPS

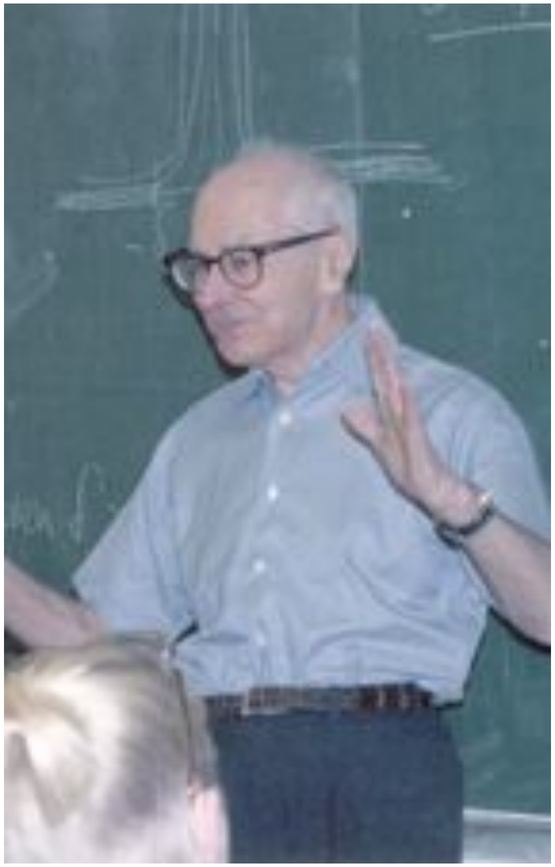
David Harbater
UPenn

Pavel
Curtis



Todd
Knoblock

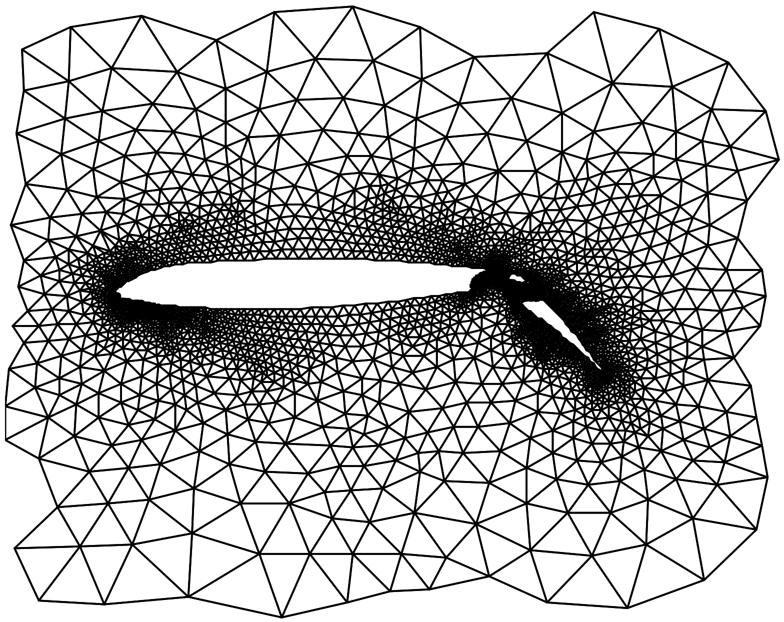
CTY

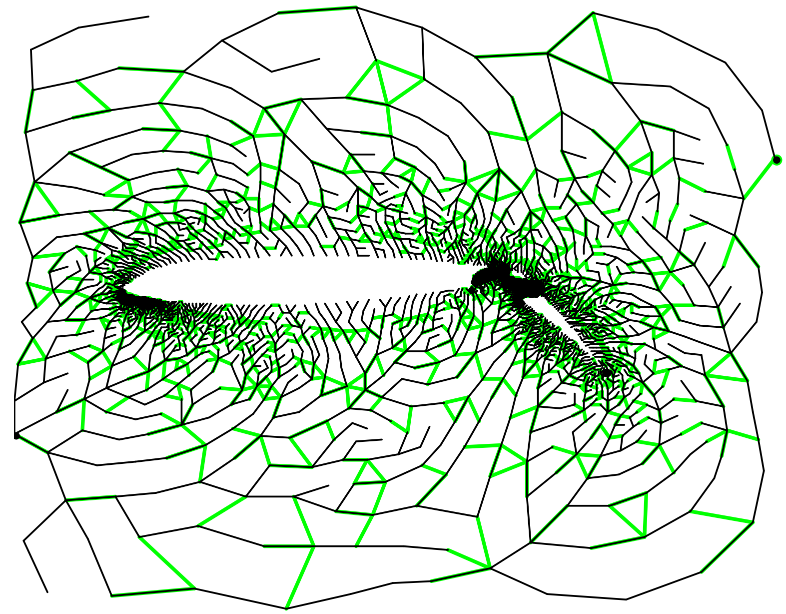


Serge Lang
1927-2005

Gian-Carlo Rota
1932-1999

Algorithms, Graph Theory, and Linear Equations in Laplacians





Daniel A. Spielman
Yale University

Solving Linear Equations $Ax = b$, Quickly

Goal: In time linear in the
number of non-zeros entries of A

Special case: A is the Laplacian Matrix of a Graph

Solving Linear Equations $Ax = b$, Quickly

Goal: In time linear in the
number of non-zeros entries of A

Special case: A is the Laplacian Matrix of a Graph

Cheeger's Inequality

Random Walks

Random Matrices

Expanders

Approximations of Graphs

Solving Linear Equations $Ax = b$, Quickly

Goal: In time linear in the
number of non-zeros entries of A

Special case: A is the Laplacian Matrix of a Graph

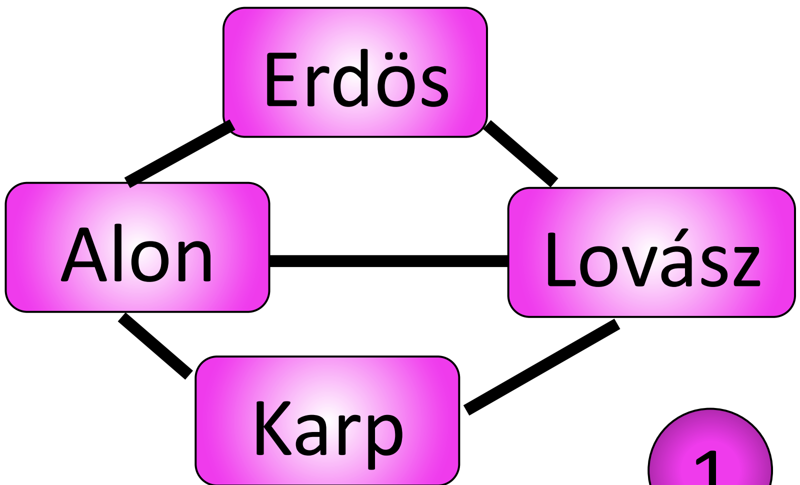
1. Why
2. Classic Approaches
3. Recent Developments
4. Connections

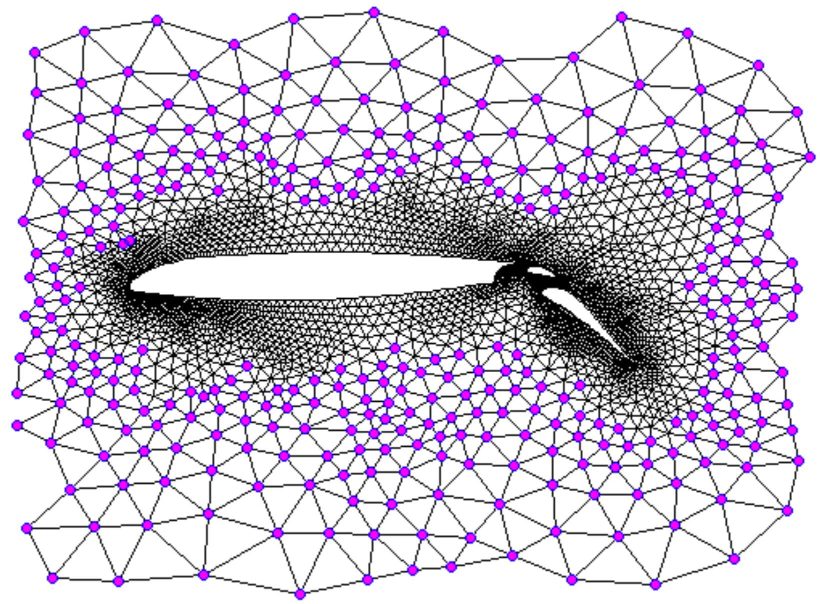
Graphs

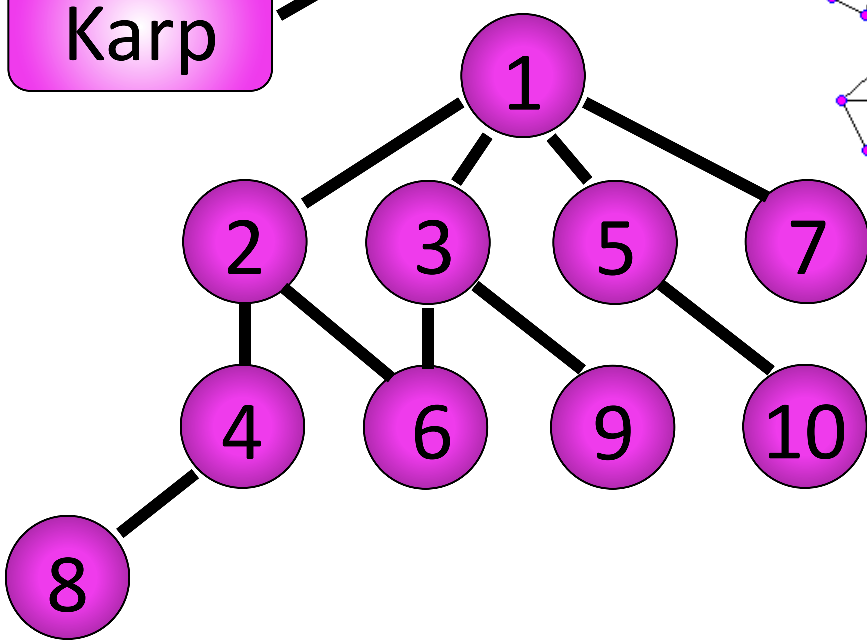
Set of vertices V. Set of edges E of pairs $\{u,v\} \subseteq V$

Graphs

Set of vertices V. Set of edges E of pairs $\{u,v\} \subseteq V$







Laplacian Quadratic Form of $G = (V,E)$

For $\boldsymbol{x} : V \to \mathbb{R}$

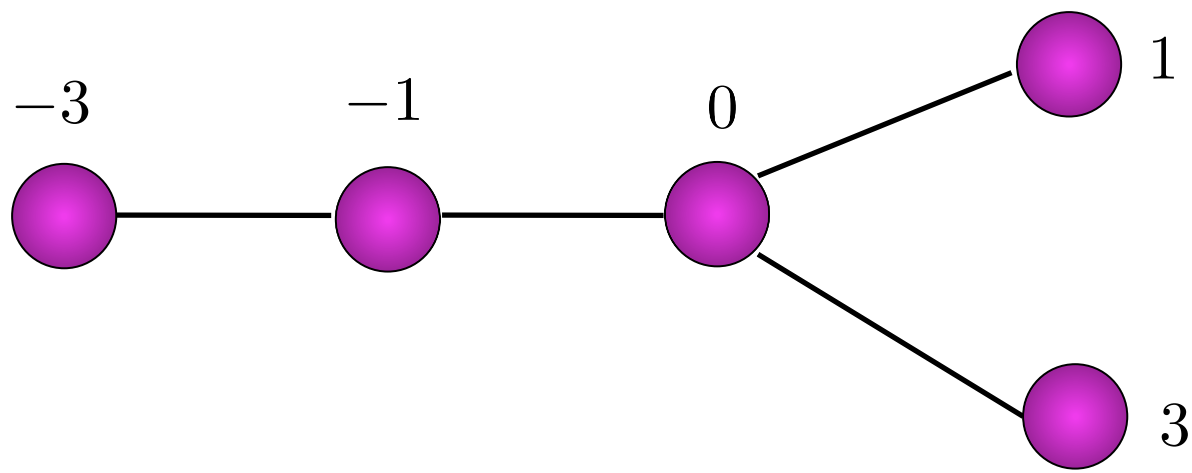
$$\boxed{\boldsymbol{x}^T L_G \boldsymbol{x} = \sum_{(u,v) \in E} \left(\boldsymbol{x}(u) - \boldsymbol{x}(v) \right)^2}$$

Laplacian Quadratic Form of $G = (V,E)$

For $\boldsymbol{x} : V \to \mathbb{R}$

$$\boxed{\boldsymbol{x}^T L_G \boldsymbol{x} = \sum_{(u,v)\in E} (\boldsymbol{x}(u) - \boldsymbol{x}(v))^2}$$

$\boldsymbol{x}$:

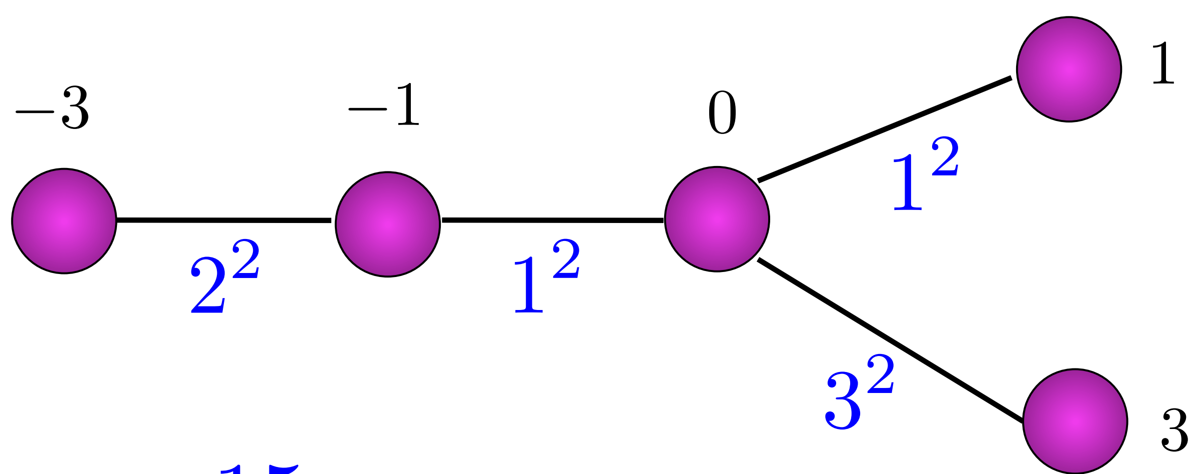


Laplacian Quadratic Form of $G = (V,E)$

For $\boldsymbol{x} : V \to \mathbb{R}$

$$\boxed{\boldsymbol{x}^T L_G \boldsymbol{x} = \sum_{(u,v)\in E} \left(\boldsymbol{x}(u) - \boldsymbol{x}(v)\right)^2}$$

$\boldsymbol{x}:$



$$\boldsymbol{x}^T L_G \boldsymbol{x} = 15$$

Laplacian Quadratic Form for Weighted Graph

$$G = (V, E, w)$$

$w : E \to \mathbb{R}^+$ assigns a positive weight to every edge

$$\boxed{\boldsymbol{x}^T L_G \boldsymbol{x} = \sum_{(u,v)\in E} w_{(u,v)} \left(\boldsymbol{x}(u) - \boldsymbol{x}(v)\right)^2}$$

Matrix L_G is positive semi-definite

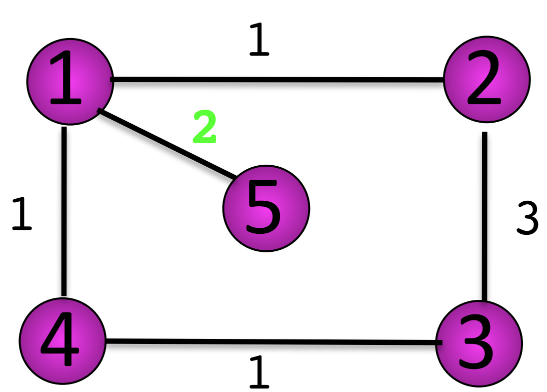
nullspace spanned by const vector, if connected

Laplacian Matrix of a Weighted Graph

$$L_G(u,v) = \begin{cases} -w(u,v) & \text{if } (u,v) \in E \\ d(u) & \text{if } u = v \\ 0 & \text{otherwise} \end{cases}$$

$$d(u) = \sum_{(v,u)\in E} w(u,v)$$

the weighted degree of u



4	−1	0	−1	−2
−1	4	−3	0	0
0	−3	4	−1	0
−1	0	−1	2	0
−2	0	0	0	2

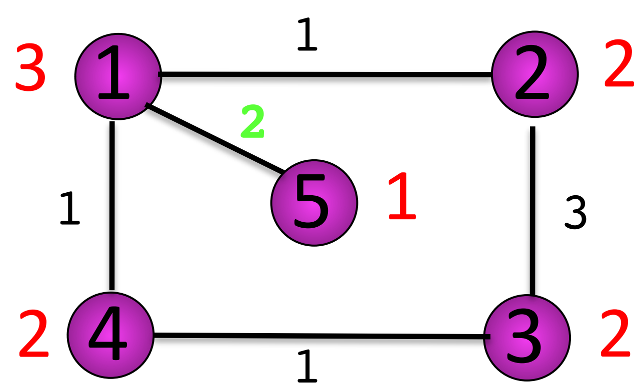
Laplacian Matrix of a Weighted Graph

$$L_G(u,v) = \begin{cases} -w(u,v) & \text{if } (u,v) \in E \\ d(u) & \text{if } u = v \\ 0 & \text{otherwise} \end{cases}$$

$$d(u) = \sum_{(v,u)\in E} w(u,v)$$

the weighted degree of u

combinatorial degree is # of attached edges



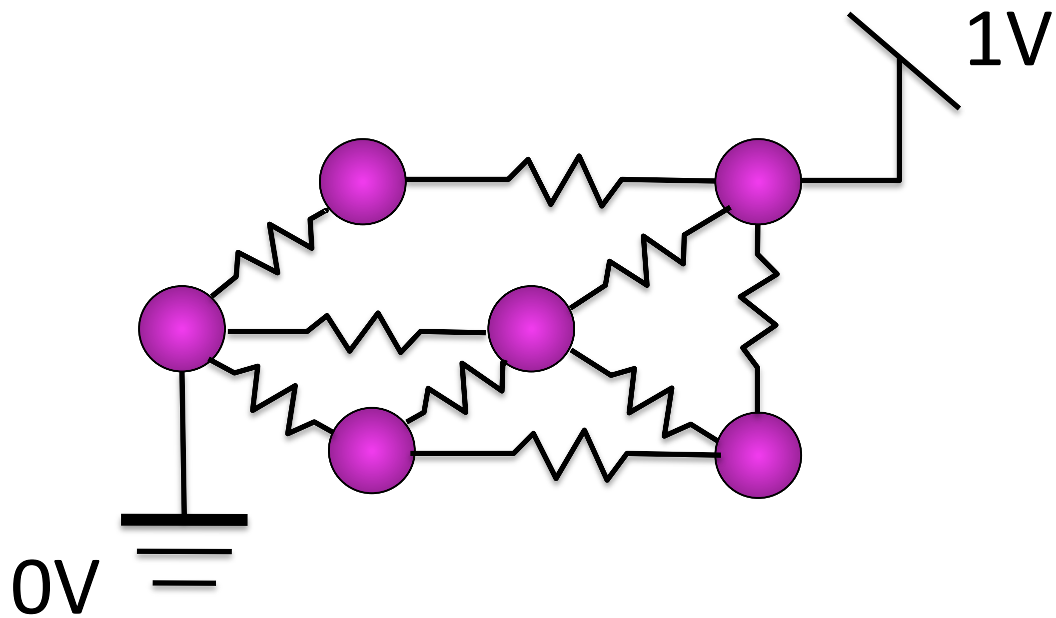
4	−1	0	−1	−2
−1	4	−3	0	0
0	−3	4	−1	0
−1	0	−1	2	0
−2	0	0	0	2

Networks of Resistors

Ohm's laws gives $i = v/r$

In general, $\boldsymbol{i} = L_G \boldsymbol{v}$ with $w_{(u,v)} = 1/r_{(u,v)}$

Minimize dissipated energy $\boldsymbol{v}^T L_G \boldsymbol{v}$



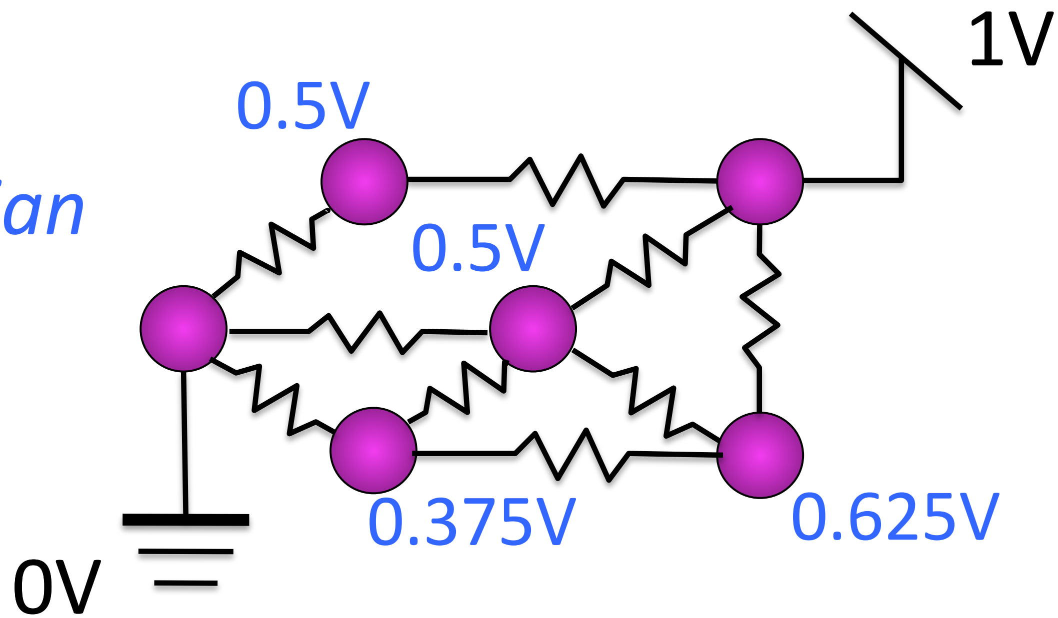
Networks of Resistors

Ohm's laws gives $i = v/r$

In general, $\boldsymbol{i} = L_G \boldsymbol{v}$ with $w_{(u,v)} = 1/r_{(u,v)}$

Minimize dissipated energy $\boldsymbol{v}^T L_G \boldsymbol{v}$

By solving Laplacian

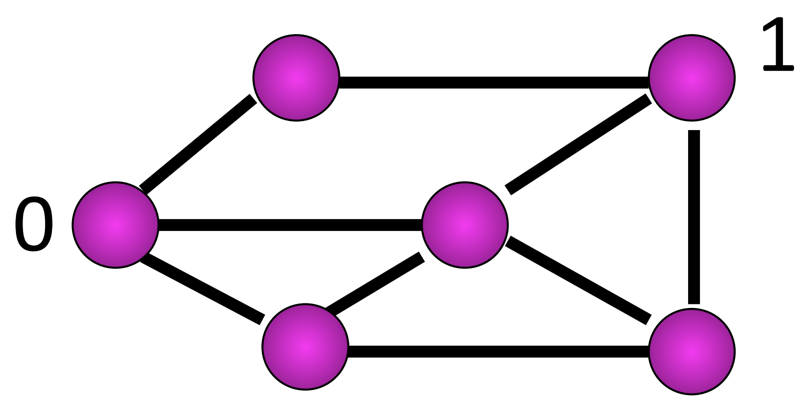


Learning on Graphs

Infer values of a function at all vertices
from known values at a few vertices.

Minimize $\boldsymbol{x}^T L_G \boldsymbol{x} = \sum_{(u,v)\in E} w_{(u,v)} \left(\boldsymbol{x}(u) - \boldsymbol{x}(v)\right)^2$

Subject to known values

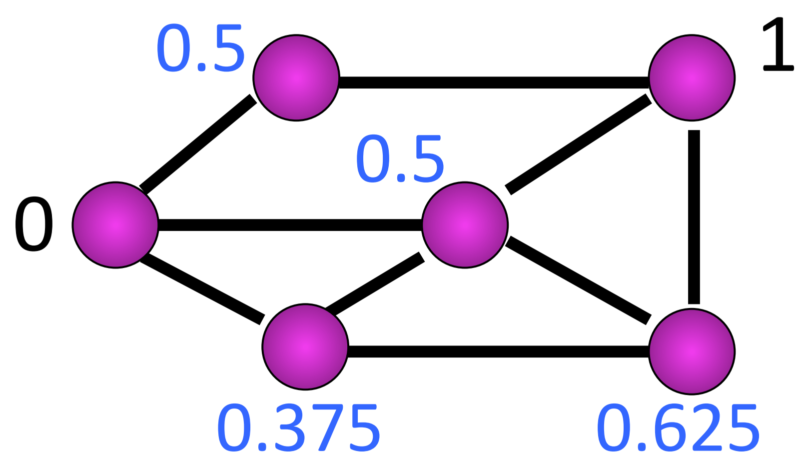


Learning on Graphs

Infer values of a function at all vertices
from known values at a few vertices.

Minimize $\boldsymbol{x}^T L_G \boldsymbol{x} = \sum_{(u,v)\in E} w_{(u,v)} \left(\boldsymbol{x}(u) - \boldsymbol{x}(v)\right)^2$

Subject to known values

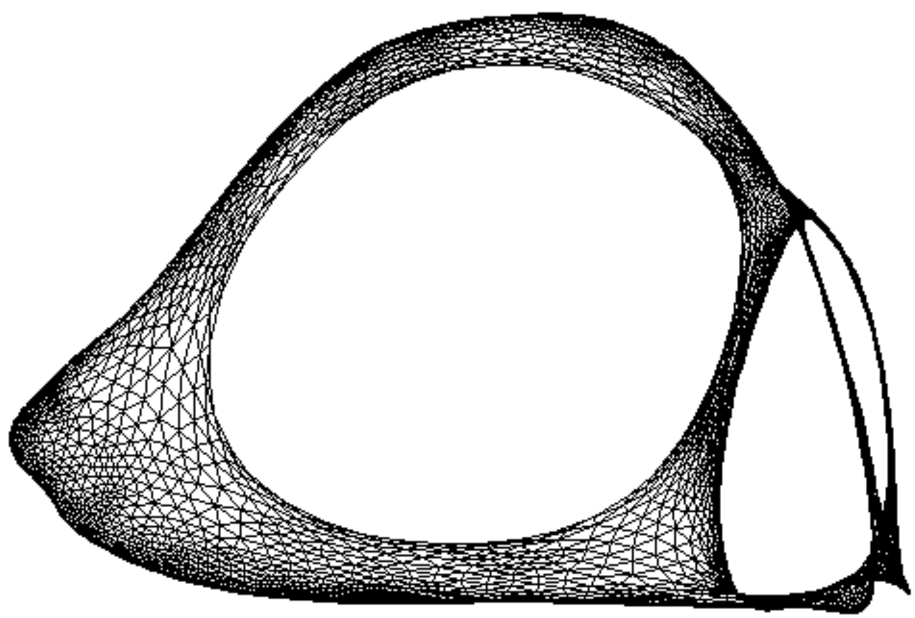


By solving Laplacian

Spectral Graph Theory

Combinatorial properties of G are
revealed by eigenvalues and eigenvectors of L_G

Compute the most important ones by
solving equations in the Laplacian.



Solving Linear Programs in Optimization

Interior Point Methods for Linear Programming:
network flow problems $\Rightarrow$ Laplacian systems

Numerical solution of Elliptic PDEs

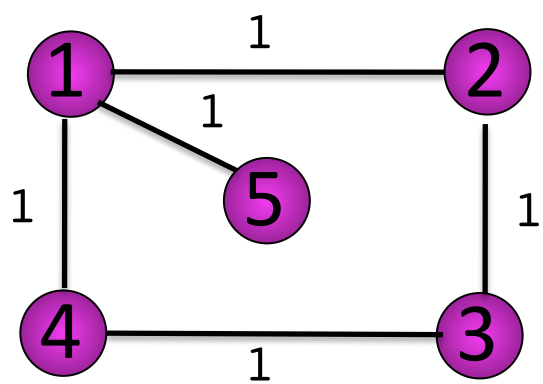
Finite Element Method

How to Solve Linear Equations Quickly

Fast when graph is simple,
by elimination.

Fast when graph is complicated*,
by Conjugate Gradient (Hestenes '51, Stiefel '52)

Cholesky Factorization of Laplacians

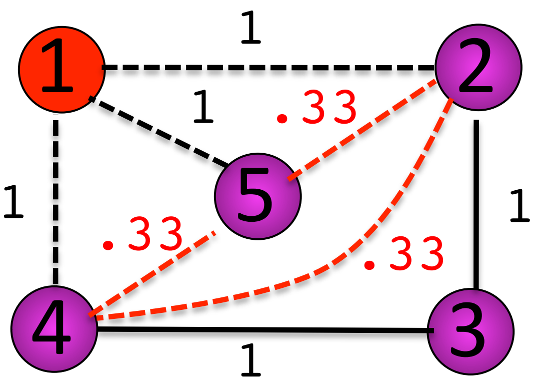


$$\begin{matrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 2 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{matrix}$$

When eliminate a vertex,
connect its neighbors.

Also known as Y-Δ

Cholesky Factorization of Laplacians



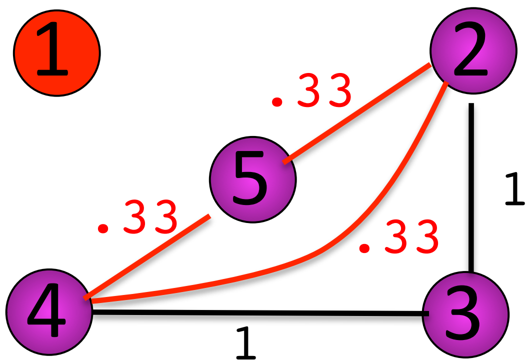
3	−1	0	−1	−1
−1	2	−1	0	0
0	−1	2	−1	0
−1	0	−1	2	0
−1	0	0	0	1

3	0	0	0	0
0	1.67	−1.00	−0.33	−0.33
0	−1.00	2.00	−1.00	0
0	−0.33	−1.00	1.67	−0.33
0	−0.33	0	−0.33	0.67

When eliminate a vertex, connect its neighbors.

Also known as Y-Δ

Cholesky Factorization of Laplacians

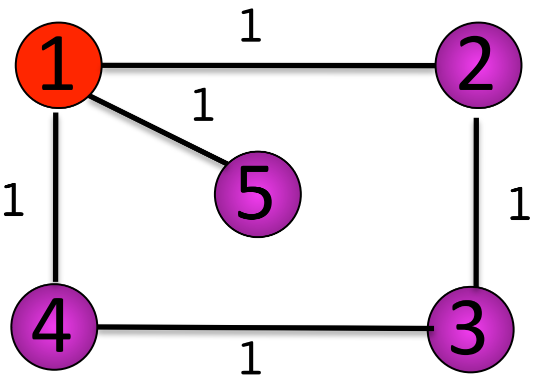


$$\begin{matrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 2 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{matrix}$$

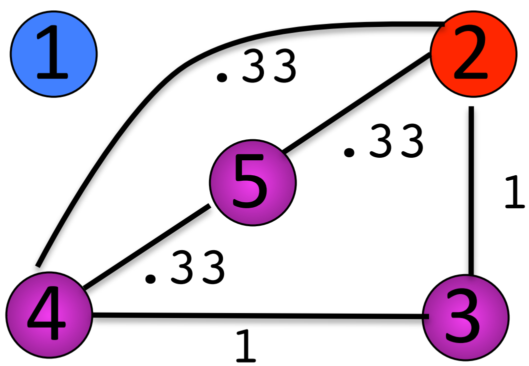
$$\begin{matrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 1.67 & -1.00 & -0.33 & -0.33 \\ 0 & -1.00 & 2.00 & -1.00 & 0 \\ 0 & -0.33 & -1.00 & 1.67 & -0.33 \\ 0 & -0.33 & 0 & -0.33 & 0.67 \end{matrix}$$

When eliminate a vertex, connect its neighbors.

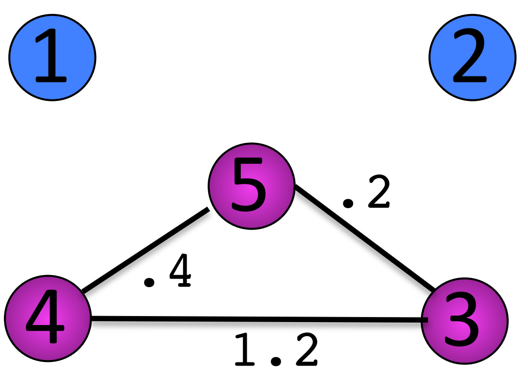
Also known as Y-Δ



$$\begin{matrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 2 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{matrix}$$

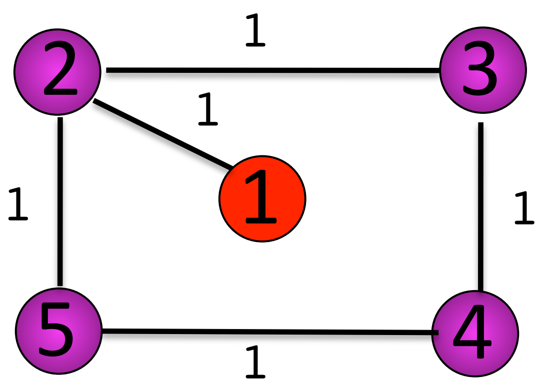


$$\begin{matrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 1.67 & -1.00 & -0.33 & -0.33 \\ 0 & -1.00 & 2.00 & -1.00 & 0 \\ 0 & -0.33 & -1.00 & 1.67 & -0.33 \\ 0 & -0.33 & 0 & -0.33 & 0.67 \end{matrix}$$

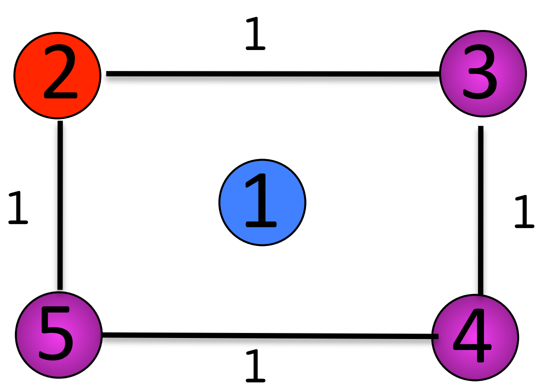


$$\begin{matrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 1.67 & 0 & 0 & 0 \\ 0 & 0 & 1.4 & -1.2 & -0.2 \\ 0 & 0 & -1.2 & 1.6 & -0.4 \\ 0 & 0 & -0.2 & -0.4 & 0.6 \end{matrix}$$

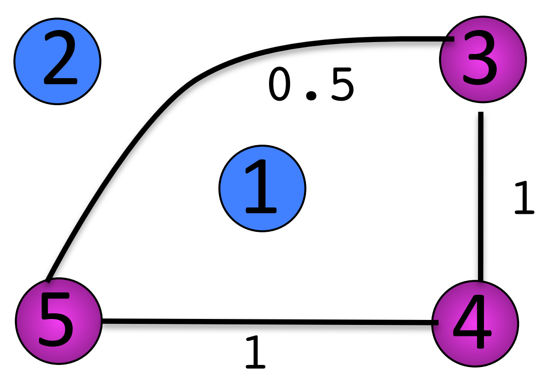
The order matters



1	-1	0	0	0
-1	3	-1	0	-1
0	-1	2	-1	0
0	0	-1	2	-1
0	-1	0	-1	2



1	0	0	0	0
0	2	-1	0	-1
0	-1	2	-1	0
0	0	-1	2	-1
0	-1	0	-1	2



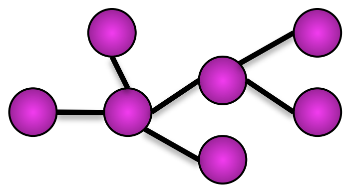
1	0	0	0	0
0	2	0	0	0
0	0	1.5	-1	-0.5
0	0	-1.0	2	-1.0
0	0	-0.5	-1	1.5

Complexity of Cholesky Factorization

#ops ~ Σ_v (degree of v when eliminate)^2

Tree

(connected,
no cycles)

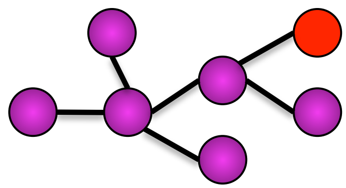


#ops ~ O(|V|)

Complexity of Cholesky Factorization

#ops ~ Σ_v (degree of v when eliminate)^2

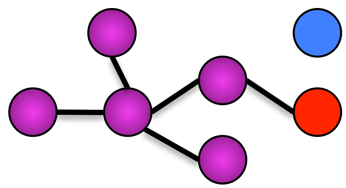
Tree
(connected,
no cycles)



#ops ~ O(|V|)

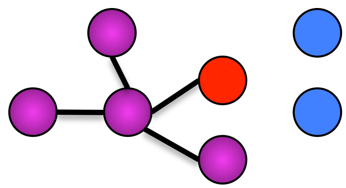
Complexity of Cholesky Factorization

#ops ~ Σ_v (degree of v when eliminate)^2

Tree  #ops ~ O(|V|)

Complexity of Cholesky Factorization

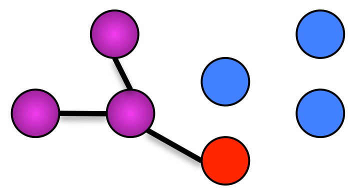
#ops ~ Σ_v (degree of v when eliminate)^2

Tree  #ops ~ O(|V|)

Complexity of Cholesky Factorization

#ops ~ Σ_v (degree of v when eliminate)^2

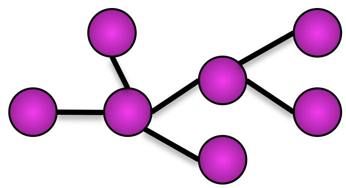
Tree



#ops ~ O(|V|)

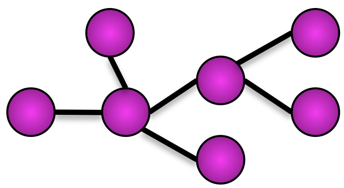
Complexity of Cholesky Factorization

#ops ~ Σ_v (degree of v when eliminate)^2

Tree  #ops ~ O(|V|)

Complexity of Cholesky Factorization

#ops ~ Σ_v (degree of v when eliminate)^2

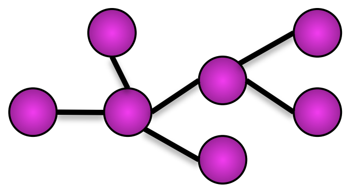
Tree  #ops ~ $O(|V|)$

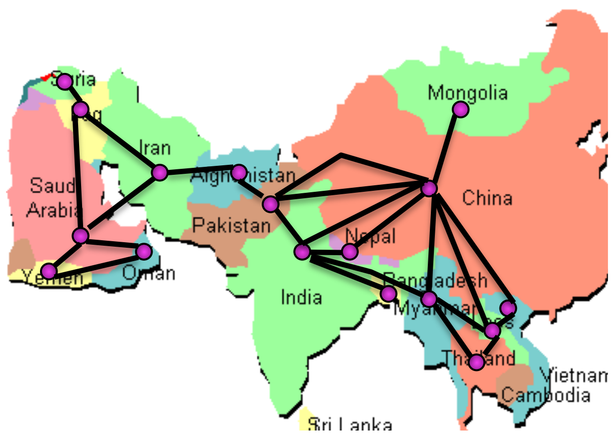
Planar 

#ops ~ $O(|V|^{3/2})$
Lipton-Rose-Tarjan '79

Complexity of Cholesky Factorization

#ops ~ Σ_v (degree of v when eliminate)^2

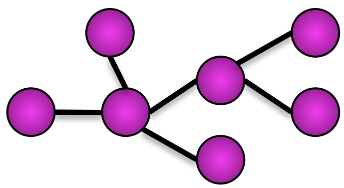
Tree  #ops ~ $O(|V|)$

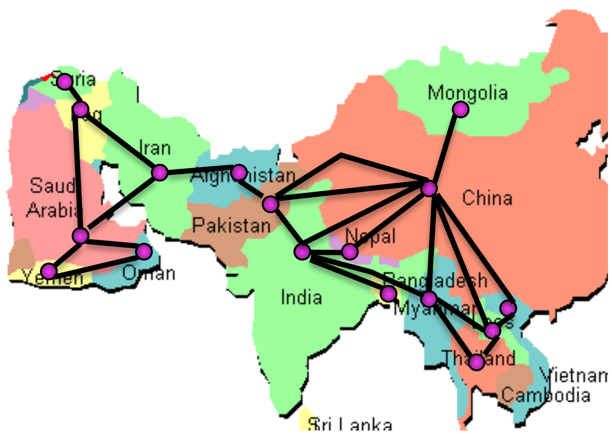
Planar 

#ops ~ $O(|V|^{3/2})$
Lipton-Rose-Tarjan '79

Complexity of Cholesky Factorization

#ops ~ Σ_v (degree of v when eliminate)^2

Tree  #ops ~ $O(|V|)$

Planar 

#ops ~ $O(|V|^{3/2})$
Lipton-Rose-Tarjan ‘79

Expander

like random,
but $O(|V|)$ edges

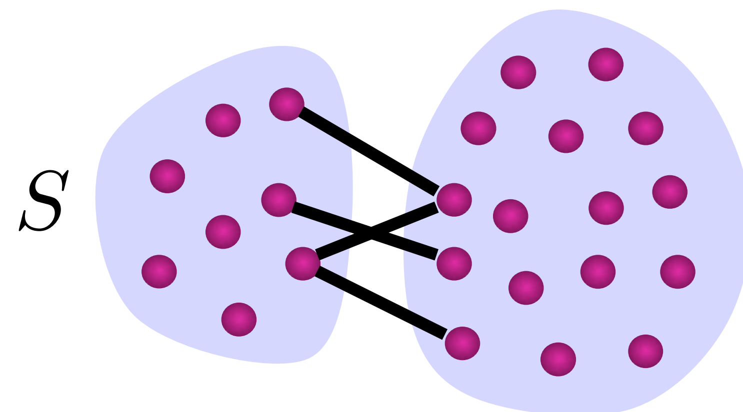
#ops $\gtrsim \Omega(|V|^3)$
Lipton-Rose-Tarjan ‘79

Conductance and Cholesky Factorization

Cholesky slow when conductance high

Cholesky fast when low for G and all subgraphs

For $S \subset V$

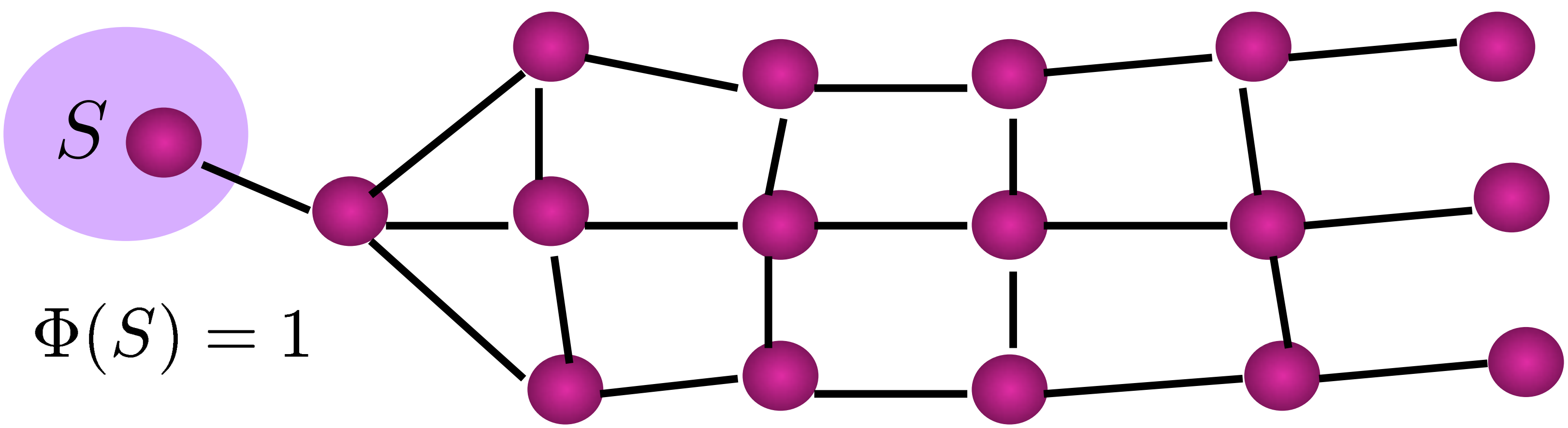


$$\Phi(S) = \frac{\#\text{ edges leaving } S}{\text{sum degrees on smaller side, } S \text{ or } V - S}$$

$$\boxed{\Phi_G = \min_{S \subset V} \Phi(S)}$$

Conductance

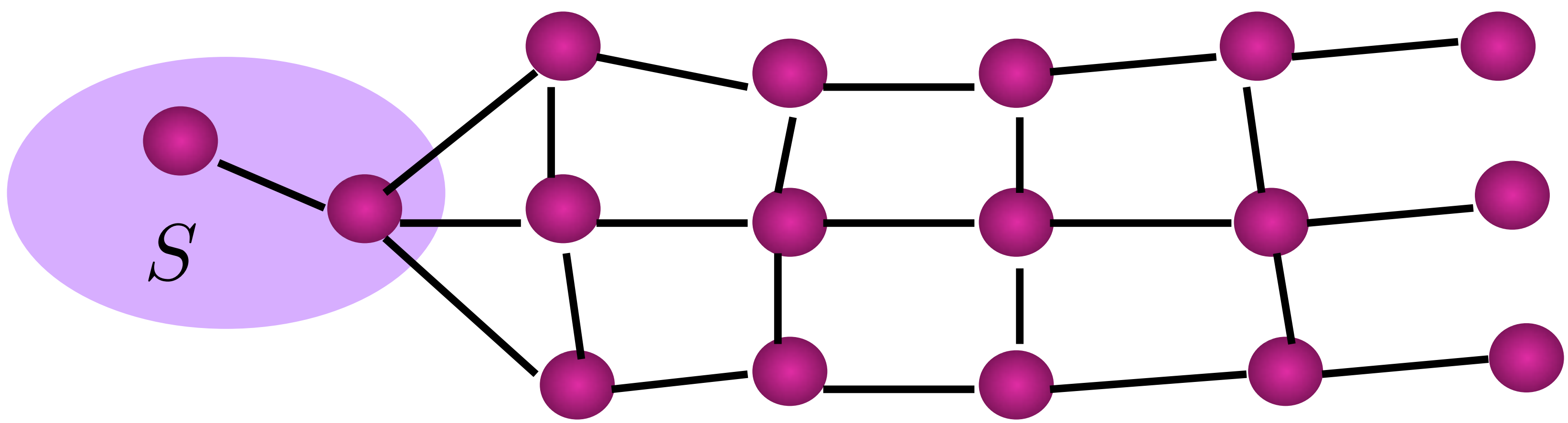
$$\Phi(S) \stackrel{\text{def}}{=} \frac{\#\text{ edges leaving } S}{\text{sum of degrees on smaller side}}$$



$\Phi(S) = 1$

Conductance

$$\Phi(S) \stackrel{\text{def}}{=} \frac{\#\text{ edges leaving } S}{\text{sum of degrees on smaller side}}$$

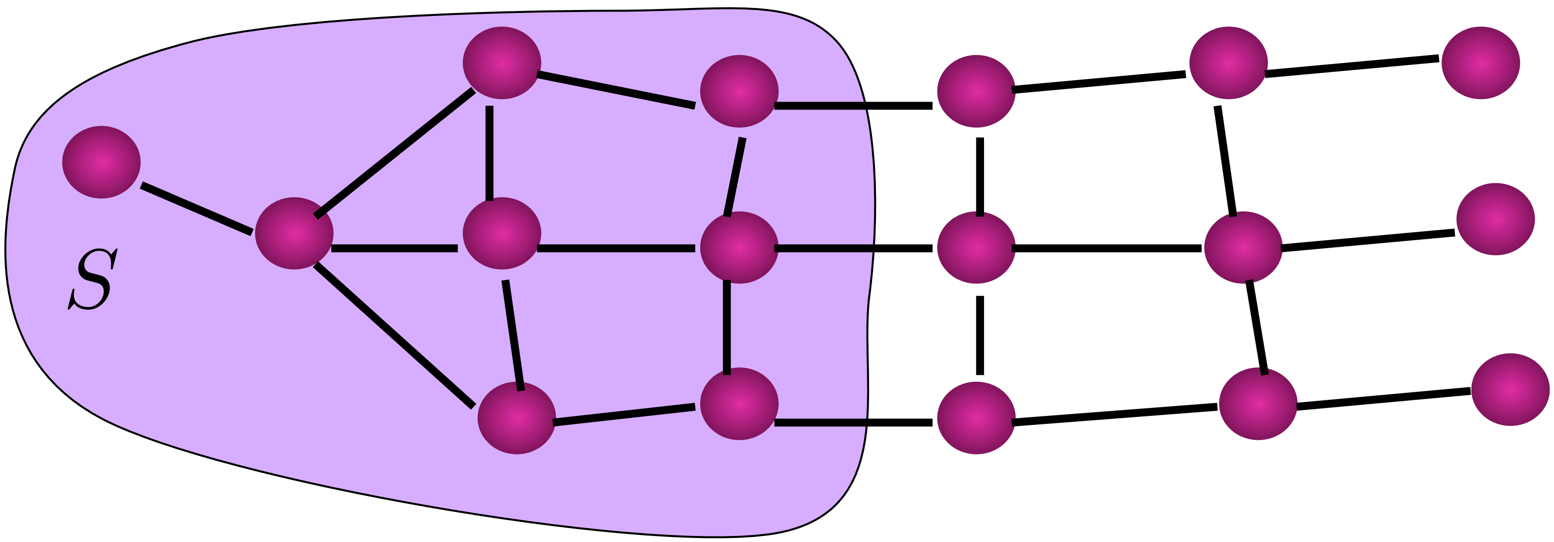


$$\Phi(S) = 3/5$$

Conductance

$$\Phi(S) \stackrel{\text{def}}{=} \frac{\#\text{ edges leaving } S}{\text{sum of degrees on smaller side}}$$

$$\Phi_G \stackrel{\text{def}}{=} \min_S \Phi(S)$$



$$\Phi(S) = 3/\min(25, 23) = \Phi_G$$

Cheeger's Inequality and the Conjugate Gradient

Cheeger's inequality (degree-d unwted case)

$$\boxed{\frac{1}{2}\frac{\lambda_2}{d} \leq \Phi_G \leq \sqrt{2\frac{\lambda_2}{d}}}$$

λ_2 = second-smallest eigenvalue of L_G

~ d/mixing time of random walk

Cheeger's Inequality and the Conjugate Gradient

Cheeger's inequality (degree-d unwted case)

$$\boxed{\frac{1}{2}\frac{\lambda_2}{d} \leq \Phi_G \leq \sqrt{2\frac{\lambda_2}{d}}}$$

λ_2 = second-smallest eigenvalue of L_G
~ d/mixing time of random walk

Conjugate Gradient finds ϵ-approx solution to $L_G x = b$

in $O(\sqrt{d/\lambda_2}\log \epsilon^{-1})$ mults by L_G

in $O(|E|\sqrt{d/\lambda_2}\log \epsilon^{-1})$ ops

Fast solution of linear equations

CG fast when conductance high.

Elimination fast when low for G and all subgraphs.

Fast solution of linear equations

CG fast when conductance high.

Planar graphs

Elimination fast when low for G and all subgraphs.

Problems:

Want speed of extremes in the middle

Fast solution of linear equations

CG fast when conductance high.

Planar graphs

Elimination fast when low for G and all subgraphs.

Problems:

Want speed of extremes in the middle

Not all graphs fit into these categories!

Preconditioned Conjugate Gradient

Solve $L_G x = b$ by

Approximating L_G by L_H (the preconditioner)

In each iteration
solve a system in L_H
multiply a vector by L_G

ϵ-approx solution after
$O(\sqrt{\kappa(L_G, L_H)} \log \epsilon^{-1})$ iterations

Preconditioned Conjugate Gradient

Solve $L_G x = b$ by

Approximating L_G by L_H (the preconditioner)

In each iteration
 solve a system in L_H
 multiply a vector by L_G

ϵ-approx solution after

$$O(\underbrace{\sqrt{\kappa(L_G, L_H)}}_{\textit{relative condition number}} \log \epsilon^{-1}) \text{ iterations}$$

Inequalities and Approximation

$L_H \preccurlyeq L_G$ if $L_G - L_H$ is positive semi-definite,
i.e. for all x,

$$x^T L_H x \preccurlyeq x^T L_G x$$

Example: if H is a subgraph of G

$$\boldsymbol{x}^T L_G \boldsymbol{x} = \sum_{(u,v)\in E} w_{(u,v)} \left(\boldsymbol{x}(u) - \boldsymbol{x}(v)\right)^2$$

Inequalities and Approximation

$L_H \preccurlyeq L_G$ if $L_G - L_H$ is positive semi-definite,
i.e. for all x,

$$x^T L_H x \preccurlyeq x^T L_G x$$

$\kappa(L_G, L_H) \leq t$

if $\quad L_H \preccurlyeq L_G \preccurlyeq t L_H$

iff $\ cL_H \preccurlyeq L_G \preccurlyeq ctL_H$ for some c

Inequalities and Approximation

$L_H \preccurlyeq L_G$ if $L_G - L_H$ is positive semi-definite,
i.e. for all x,

$$x^T L_H x \preccurlyeq x^T L_G x$$

$\kappa(L_G, L_H) \leq t$

if $\quad L_H \preccurlyeq L_G \preccurlyeq t L_H$

iff $\; cL_H \preccurlyeq L_G \preccurlyeq ct L_H$ for some c

$\boxed{\text{Call } H \text{ a } t\text{-approx of } G \text{ if } \kappa(L_G, L_H) \leq t}$

Other definitions of the condition number

(Goldstine, von Neumann '47)

$$\kappa(L_G, L_H) = \left(\max_{x \in \mathbf{Span}(L_H)} \frac{x^T L_G x}{x^T L_H x} \right) \left(\max_{x \in \mathbf{Span}(L_G)} \frac{x^T L_H x}{x^T L_G x} \right)$$

pseudo-inverse

$$\kappa(L_G, L_H) = \frac{\lambda_{max}(L_G L_H^+)}{\lambda_{min}(L_G L_H^+)}$$

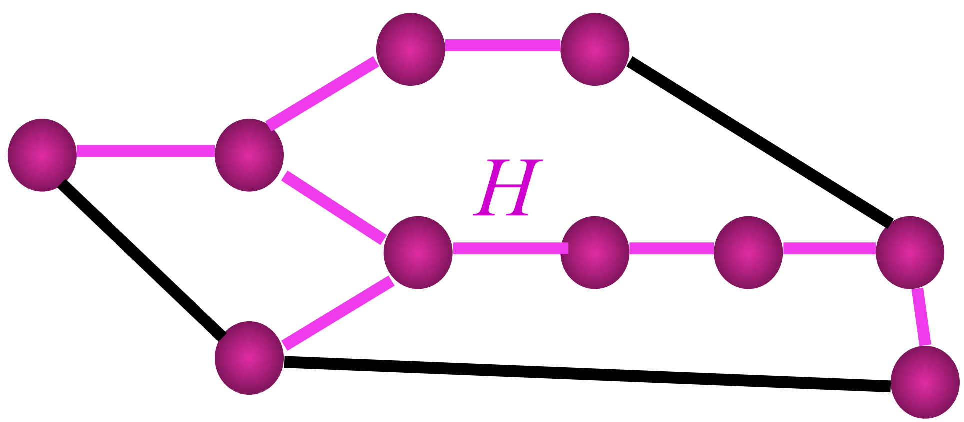
min non-zero eigenvalue

Vaidya's Subgraph Preconditioners

Precondition G by a subgraph H

$L_H \preccurlyeq L_G$ so just need t for which $L_G \preccurlyeq t L_H$

Easy to bound t if H is a spanning tree

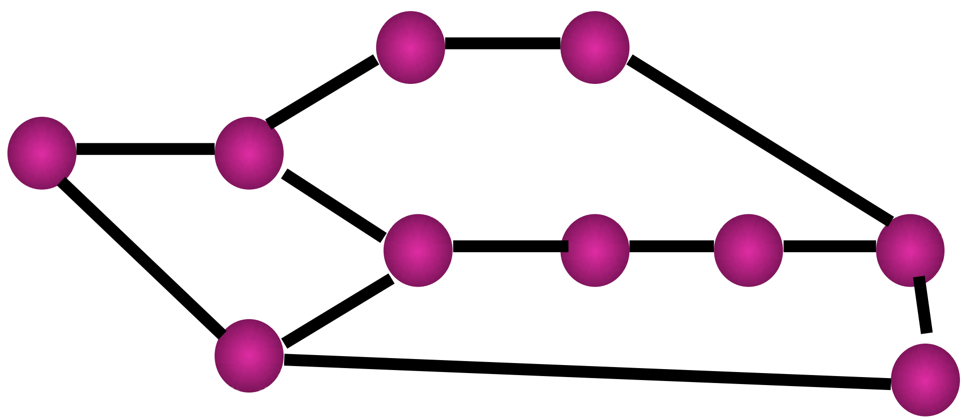


And, easy to solve equations in L_H by elimination

The Stretch of Spanning Trees

Boman-Hendrickson '01: $L_G \preccurlyeq \mathrm{st}_G(T) L_T$

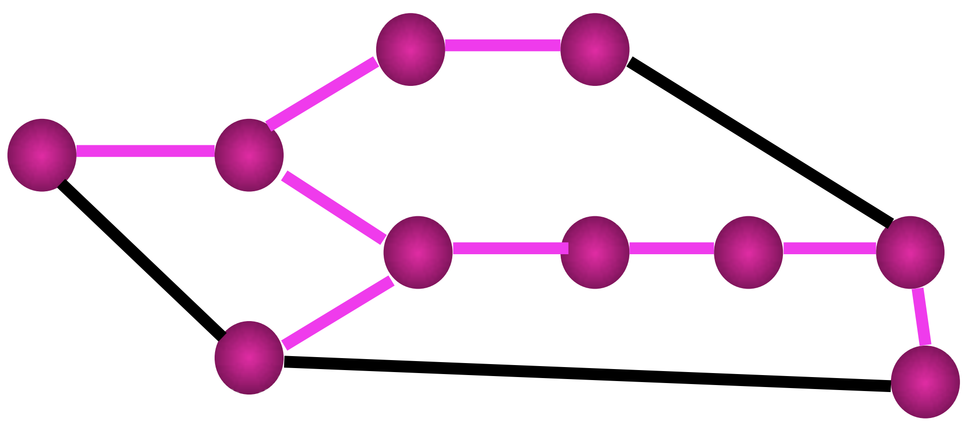
Where $\mathrm{st}_T(G) = \sum_{(u,v) \in E} \text{path-length}_T(u,v)$



The Stretch of Spanning Trees

Boman-Hendrickson '01: $L_G \preccurlyeq \mathrm{st}_G(T) L_T$

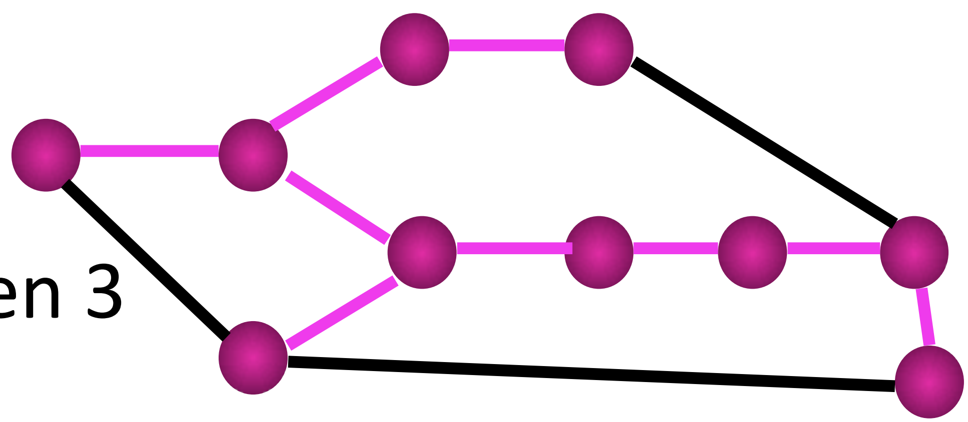
Where $\mathrm{st}_T(G) = \sum_{(u,v)\in E} \text{path-length}_T(u,v)$



The Stretch of Spanning Trees

Boman-Hendrickson '01: $L_G \preccurlyeq \mathrm{st}_G(T) L_T$

Where $\mathrm{st}_T(G) = \sum_{(u,v) \in E} \text{path-length}_T(u,v)$

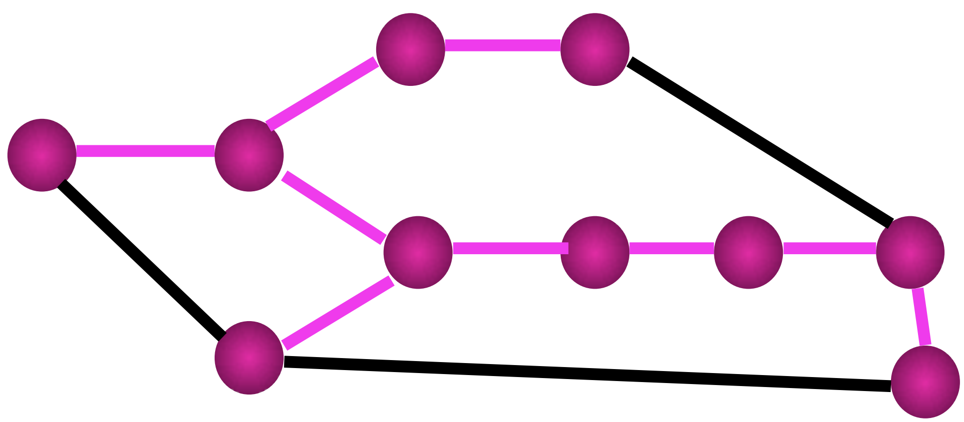


path-len 3

The Stretch of Spanning Trees

Boman-Hendrickson '01: $L_G \preccurlyeq \mathrm{st}_G(T) L_T$

Where $\mathrm{st}_T(G) = \sum_{(u,v) \in E} \text{path-length}_T(u,v)$

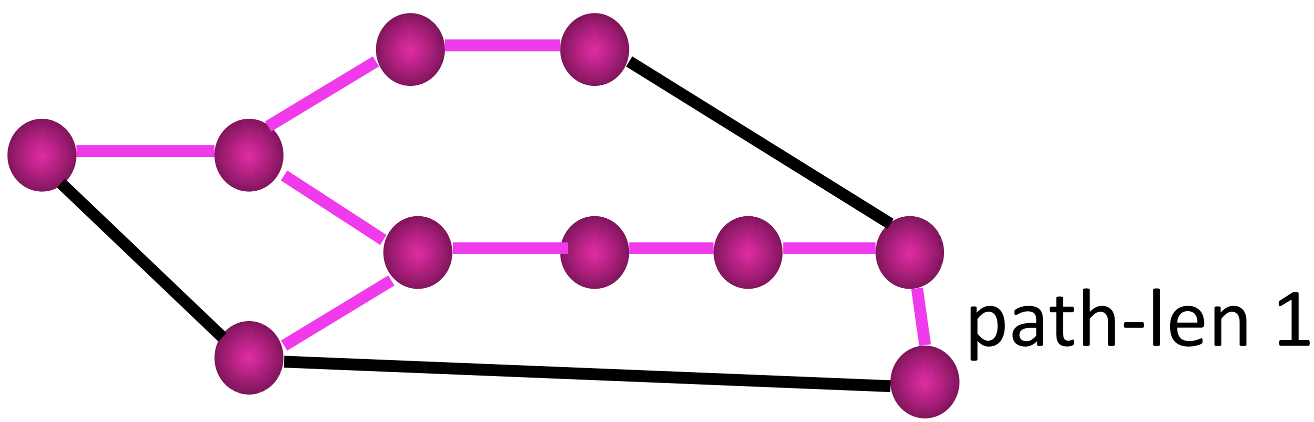


path-len 5

The Stretch of Spanning Trees

Boman-Hendrickson '01: $L_G \preccurlyeq \mathrm{st}_G(T) L_T$

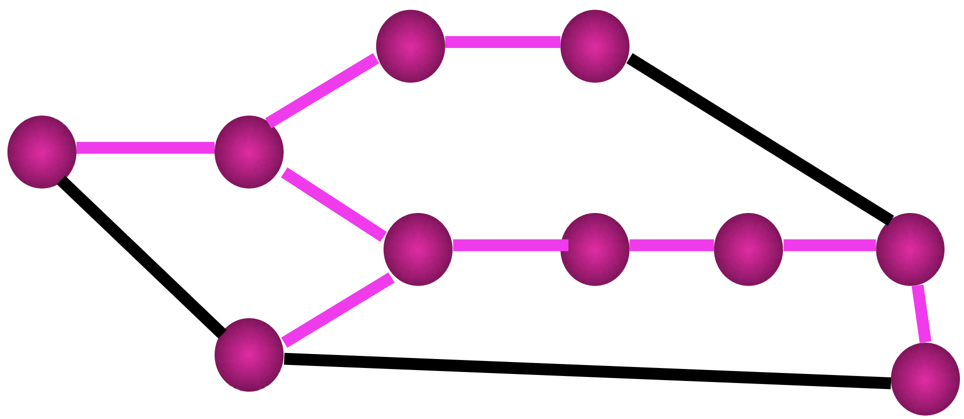
Where $\mathrm{st}_T(G) = \sum_{(u,v) \in E} \text{path-length}_T(u,v)$



The Stretch of Spanning Trees

Boman-Hendrickson '01: $L_G \preccurlyeq \mathrm{st}_G(T) L_T$

Where $\mathrm{st}_T(G) = \sum_{(u,v) \in E} \text{path-length}_T(u, v)$



In weighted case, measure resistances of paths

Low-Stretch Spanning Trees

For every G there is a T with

$$\mathrm{st}_T(G) \leq m^{1+o(1)}$$ where $m = |E|$

(Alon-Karp-Peleg-West '91)

$$\mathrm{st}_T(G) \leq O(m \log m \log^2 \log m)$$

(Elkin-Emek-S-Teng '04, Abraham-Bartal-Neiman '08)

$$\boxed{\text{Solve linear systems in time } O(m^{3/2} \log m)}$$

Low-Stretch Spanning Trees

For every G there is a T with

$$\mathrm{st}_T(G) \leq m^{1+o(1)}$$ where $m = |E|$

(Alon-Karp-Peleg-West '91)

$$\mathrm{st}_T(G) \leq O(m \log m \log^2 \log m)$$

(Elkin-Emek-S-Teng '04, Abraham-Bartal-Neiman '08)

If G is an expander $\mathrm{st}_T(G) \geq \Omega(m \log m)$

Expander Graphs

Infinite family of d-regular graphs (all degrees d)
satsifying $\lambda_2 \geq \mathrm{const} > 0$

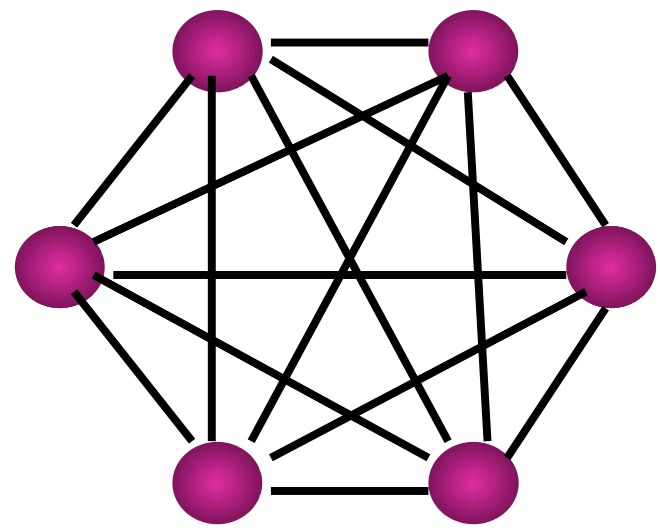
Spectrally best are Ramanujan Graphs
(Margulis '88, Lubotzky-Phillips-Sarnak '88)
all eigenvalues inside $d \pm 2\sqrt{d-1}$

Fundamental examples

Amazing properties

Expanders Approximate Complete Graphs

Let G be the complete graph on n vertices
(having all possible edges)



All non-zero eigenvalues of L_G are n

$$x^T L_G x = n \quad \text{for all} \quad x \perp \mathbf{1}, \|x\| = 1$$

Expanders Approximate Complete Graphs

Let G be the complete graph on n vertices

$$x^T L_G x = n \quad \text{for all} \quad x \perp \mathbf{1}, \|x\| = 1$$

Let H be a d-regular Ramanujan Expander

$$(d - 2\sqrt{d-1}) \leq x^T L_H x \leq (d + 2\sqrt{d-1})$$

$$\boxed{\kappa(L_G, L_H) \leq \frac{d + 2\sqrt{d-1}}{d - 2\sqrt{d-1}} \to 1}$$

Sparsification

Goal: find sparse approximation for every G

S-Teng '04: For every G is an H with
$O(n \log^7 n / \epsilon^2)$ edges and $\kappa(L_G, L_H) \leq 1 + \epsilon$

Sparsification

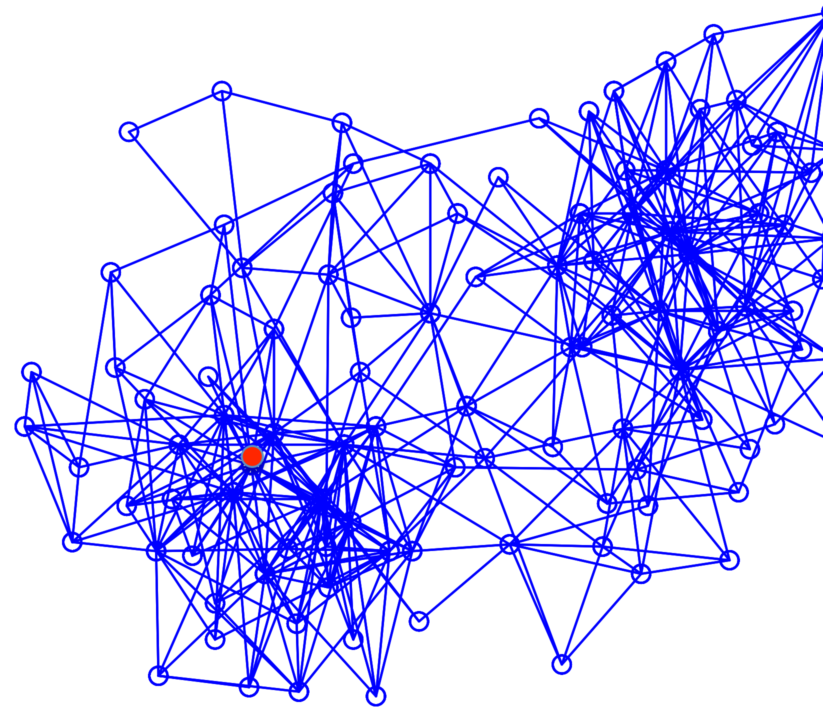
S-Teng '04: For every G is an H with
$O(n \log^7 n/\epsilon^2)$ edges and $\kappa(L_G, L_H) \leq 1 + \epsilon$

Conductance high
$\Longrightarrow$ λ_2 high (Cheeger)
$\Longrightarrow$ random sample good (Füredi-Komlós '81)

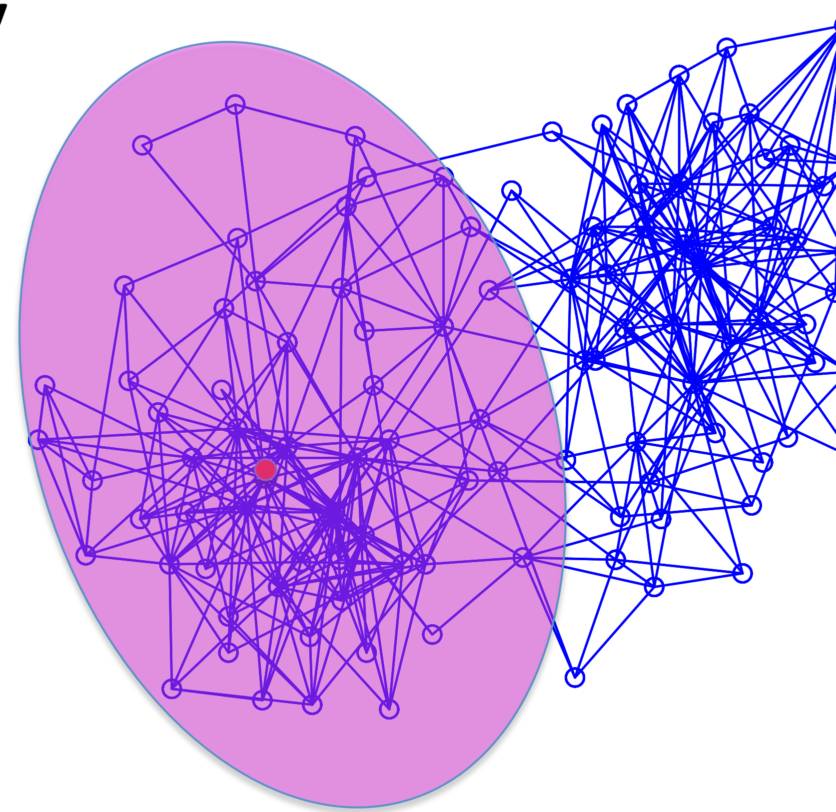
Conductance not high
$\Longrightarrow$ can split graph while removing few edges

Fast Graph Decomposition by local graph clustering

Given vertex of interest
find nearby cluster, small $\Phi(S)$,
in time $O(|S|)$



Fast Graph Decomposition by local graph clustering



Given vertex of interest
find nearby cluster, small $\Phi(S)$,
in time $O(|S|)$

S-Teng '04: Lovász-Simonovits
Andersen-Chung-Lang '06: PageRank
Andersen-Peres '09: evoloving set Markov chain

Sparsification

Goal: find sparse approximation for every G

S-Teng '04: For every G is an H with
$O(n \log^7 n/\epsilon^2)$ edges and $\kappa(L_G, L_H) \leq 1+\epsilon$

S-Srivastava '08: with $O(n \log n/\epsilon^2)$ edges
proof by modern random matrix theory
Rudelson's concentration for random sums

Sparsification

Goal: find sparse approximation for every G

S-Teng '04: For every G is an H with

$O(n \log^7 n/\epsilon^2)$ edges and $\kappa(L_G, L_H) \leq 1 + \epsilon$

S-Srivastava '08: with $O(n \log n/\epsilon^2)$ edges

Batson-S-Srivastava '09

dn edges and $\kappa(L_G, L_H) \leq \dfrac{d + 1 + 2\sqrt{d}}{d + 1 - 2\sqrt{d}}$

Sparsification by Linear Algebra

edges $\Rightarrow$ vectors

Given vectors $v_1, ..., v_m \in \mathbb{R}^n$ s.t. $\sum_e v_e v_e^T = I$

Find a small subset $S \subset \{1, ..., m\}$

and coefficients c_e s.t.

$$|| \sum_{e \in S} c_e v_e v_e^T - I || \leq \epsilon$$

Sparsification by Linear Algebra

Given vectors $v_1, ..., v_m \in \mathbb{R}^n$ s.t. $\sum_e v_e v_e^T = I$

Find a small subset $S \subset \{1, ..., m\}$

and coefficients c_e s.t. $||\sum_{e \in S} c_e v_e v_e^T - I|| \leq \epsilon$

Rudelson '99 says can find $|S| \leq O(n \log n / \epsilon^2)$

if choose S at random*

* = $\Pr[e] \sim 1/\|v_e\|^2$

In graphs, are effective resistances

Sparsification by Linear Algebra

Given vectors $v_1, ..., v_m \in \mathbb{R}^n$ s.t. $\sum_e v_e v_e^T = I$

Find a small subset $S \subset \{1, ..., m\}$

and coefficients c_e s.t. $|| \sum_{e \in S} c_e v_e v_e^T - I|| \leq \epsilon$

Batson-S-Srivastava: can find $|S| \leq 4n/\epsilon^2$

by greedy algorithm

with Stieltjes potential function

Relation to Kadison-Singer, Paving Conjecture

Would be implied by the following strong version of Weaver's conjecture KS'$_2$

Exists constant α s.t. for $v_1, \ldots, v_m \in \mathbb{R}^n$ s.t. for

$$\|v_e\|^2 = \frac{n}{m} \leq \alpha \qquad \sum_e v_e v_e^T = I$$

Exists $S \subset \{1, \ldots, m\}, |S| = m/2$ $\left\| \sum_{e \in S} v_e v_e^T - \frac{1}{2} I \right\| \leq \frac{1}{4}$

Relation to Kadison-Singer, Paving Conjecture

Exists constant α s.t. for $v_1, \ldots, v_m \in \mathbb{R}^n$ s.t. for

$$\|v_e\|^2 = \frac{n}{m} \le \alpha \qquad\qquad \sum_e v_e v_e^T = I$$

Exists $S \subset \{1, \ldots, m\}, |S| = m/2$ $\left\| \sum_{e \in S} v_e v_e^T - \frac{1}{2} I \right\| \le \frac{1}{4}$

Rudelson '99: $\alpha = \text{const}/(\log n)$

Batson-S-Srivastava '09: true, but with coefficients in sum

Relation to Kadison-Singer, Paving Conjecture

Exists constant α s.t. for $v_1, ..., v_m \in \mathbb{R}^n$ s.t. for

$$\|v_e\|^2 = \frac{n}{m} \leq \alpha \qquad \sum_e v_e v_e^T = I$$

Exists $S \subset \{1, ..., m\}, |S| = m/2$ $\left\| \sum_{e \in S} v_e v_e^T - \frac{1}{2} I \right\| \leq \frac{1}{4}$

Rudelson '99: $\alpha = \mathrm{const}/(\log n)$

Batson-S-Srivastava '09: true, but with coefficients in sum

S-Srivastava '10: Bourgain-Tzafriri Restricted Invertability

Sparsification and Solving Linear Equations

Can reduce any Laplacian to one with $O(|V|)$ non-zero entries/edges.

S-Teng '04: Combine Low-Stretch Trees with Sparsification to solve Laplacian systems in time

$$O(m \log^{c} n \log \epsilon^{-1})$$

$$m = |E| \quad n = |V|$$

Sparsification and Solving Linear Equations

S-Teng '04: Combine Low-Stretch Trees with Sparsification to solve Laplacian systems in time

$$O(m \log^{c} n \log \epsilon^{-1})$$

$$m = |E| \quad n = |V|$$

Koutis-Miller-Peng '10: time $O(m \log^2 n \log \epsilon^{-1})$

Kolla-Makarychev-Saberi-Teng '09:
$O(m \log n \log \epsilon^{-1})$ after preprocessing

What’s next

Other families of linear equations:

from directed graphs

from physical problems

from optimization problems

Solving Linear equations as a primitive

Decompositions of the identity:

Understanding the vectors we get from graphs

the Ramanujan bound

Kadison-Singer?

Conclusion

It is all connected