Solving Laplacian linear equations

$L\underline{x} = \underline{d}$

electrical voltage
(flow routes demand $\underline{d}$)

Given L, the Laplacian of an undirected connected graph

$G = (V, E, \underline{w}) \quad \underline{w} \in \mathbb{R}^E_{(+)} \quad |V| = n, \quad |E| = m$

$\underline{d} \in \mathbb{R}^V$, demand vector, $\underline{d} \perp \underline{1}$

$\boxed{\text{GOAL: find } \underline{x}}$

Defn An ε-approximate solution $\underline{\tilde{x}}$ to $L\underline{x}^* = \underline{d}$ is $\underline{\tilde{x}}$ s.t.

$$\|\underline{\tilde{x}} - \underline{x}^*\|_L^2 \leq \varepsilon \|\underline{x}^*\|_L^2$$

$\varepsilon = n^{-100} \qquad \log\left(\frac{1}{\varepsilon}\right)$

Defn $\|y\|_L = \sqrt{y^T L y}$

Electrical Energy $\|\underline{f}^*\|_R^2 = \|\underline{x}^*\|_L^2 = \|\underline{d}\|_{L^+}^2$

Exercise

$$f^T R f = x^T L x = \underline{d}^T L^+ \underline{d}$$

Theorem Spielman-Teng 2004

We can find an ε-approximate solution $\tilde{x}$ to $L\underline{x}^* = \underline{d}$ in time $O(m \log^c n \log 1/\varepsilon)$.

The algorithm is randomized and succeeds w.h.p.

$\boxed{\text{ORIG ALGO} : c \approx 70}$

$$L\underline{x} = \underline{d}$$

$$Z \approx_{0.5} L^{-1}$$

$$0.5\, L^{-1} \preceq \underset{B^{-1}}{Z} \preceq 1.5\, L^{-1}$$

$$\|x\|_A = \sqrt{x^T A x}$$

Preconditioned Richardson Iteration

- Want to (approx) solve $A\underline{x}^* = \underline{b}$, $A \succ 0$,
- Can quickly invert B where $\frac{A}{K} \preceq B \preceq A$

$T = K\log(\frac{1}{\varepsilon})$

GOAL $\|\tilde{\underline{x}} - \underline{x}^*\|_A \leq \varepsilon \|\underline{x}^*\|_A$

$\leq e^{-\frac{2T}{K+1}}$

ALGO

$$\underline{x}_0 = \underline{0}$$

$$\underline{x}_{i+1} = \underline{x}_i - \alpha B^{-1}(A\underline{x}_i - \underline{b})$$

$\left(1 - \frac{2}{K+1}\right)^T$

Lemma $\|\underline{x}_{i+1} - \underline{x}^*\|_A \leq \|I - \alpha B^{-1}A\|_{A \to A} \|\underline{x}_i - \underline{x}^*\|_A$

$\|\cdot\|_c \quad \|\cdot\|_d$

$$\underline{x}_{i+1} - \underline{x}^* = \underline{x}_i - \underline{x}^* - B^{-1}(A\underline{x}_i - \underline{b})$$

$$= (I - B^{-1}A)(\underline{x}_i - \underline{x}^*)$$

$\|M\|_{c \to d} = \max_{\substack{x \\ \|x\|_c \leq 1}} \|Mx\|_d$

$$\|\underline{x}_{i+1} - \underline{x}^*\|_A \leq \|I - \alpha B^{-1}A\|_{A \to A} \|\underline{x}_i - \underline{x}^*\|_A$$

$\max_{\substack{x \\ \|x\|_A \leq 1}} \|Mx\|_A$

Lemma $\|I - \alpha B^{-1}A\|_{A \to A} \leq \max(1 - \alpha, |\alpha K - 1|)$

when $\frac{A}{K} \preceq B \preceq A$, $0 < \alpha < 1$

Cor Choose α well, get $\|\cdot\| \leq \frac{K-1}{K+1} = \left(1 - \frac{2}{K+1}\right)$

GAUSSIAN ELIMINATION

- A.k.a. CHOLESKY DECOMPOSITION: Gaussian Elimination when applied to a positive semi-definite matrix.

- Want to solve $M\underline{x} = \underline{d}$, $\quad M \succeq 0$

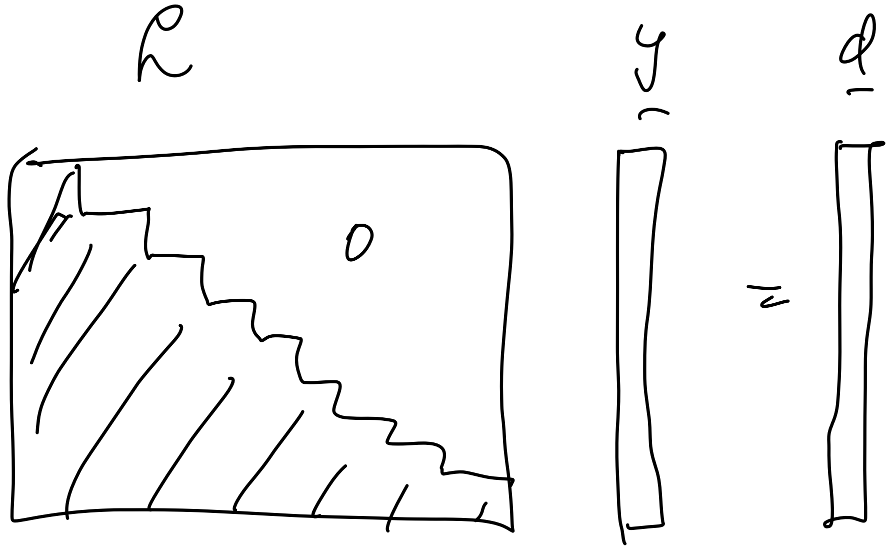
Given $M \succ 0$,

Theorem there exists a factorization $M = \mathcal{L}\mathcal{L}^T$ s.t. $\mathcal{L}$ is lower triangular i.e.

$$\mathcal{L}_{ij} = 0 \qquad i<j$$

$M\underline{x} = \underline{d}$ in two steps:

$$\mathcal{L}\mathcal{L}^T x = \underline{d} \quad \Rightarrow \quad \mathcal{L}\underline{y} = \underline{d}, \quad \mathcal{L}^T\underline{x} = \underline{y}$$



Lemma If $\mathcal{L}$ is lower triangular or upper triangular AND invertible, then

$$\mathcal{L}\underline{y} = \underline{b} \quad \text{can be solved in}$$

$$O(\mathrm{nnz}(\mathcal{L})) \text{ time.}$$

$$\underline{y} = \mathcal{L}^{-1} b$$

"Forward/Backward Substitution"

GAUSSIAN ELIMINATION ALGORITHM

$S_0 \leftarrow M$

for $i = 1, \ldots, n$

$$\underline{l}_i = \frac{1}{S(i,i)^{1/2}} S_{i-1}(:, i)$$

$$S_i = S_{i-1} - \underline{l}_i \underline{l}_i^T$$

end

$\mathcal{L} = [\underline{l}_1, \ldots, \underline{l}_n]$

Note: we can eliminate variables in other orders.

$\Rightarrow$ Then $\mathcal{L}$ is "lower triangular" w.r.t. the elimination order, which is sufficient.

S_i : the $\boxed{\text{Schur Complement}}$ of M
wrt. elimination of variables $1, 2, \dots, i$

CLAIM The Schur complement S_i does not depend on the order of elimination of the variables $1, \dots, i$

3

GAUSSIAN ELIMINATION ON LAPLACIANS

CLAIM When L is Laplacian (of a connected graph), then each S_i is the Laplacian of a connected graph on vertices $\{i+1, \dots, n\}$.

Star(v, S) = Laplacian of edges incident to v in S.

$$\text{Clique}(v, S) = \text{star}(v, S) - \underline{\ell}\,\underline{\ell}^T$$

where $\underline{\ell} = \frac{1}{S(v,v)^{1/2}} S(:, v)$

<u>Lemma</u> clique(v, S) is a Laplacian. EXERCISE

GAUSSIAN ELIMINATION ALGORITHM ON LAP.

$S_0 = L$

for $i = 1, \dots, n$

$$\underline{\ell}_i = \frac{1}{S_{i-1}(i, i)^{1/2}} S(:, i)$$

$$\cancel{S_i = S_{i-1} - \underline{\ell}_i \underline{\ell}_i^T}$$

$$\boxed{S_i = S_{i-1} - \text{Star}(i, S_{i-1}) + \text{Clique}(i, S_{i-1})}$$

end

$\mathcal{L} = [\underline{\ell}_1, \dots, \underline{\ell}_n]$

Gaussian Elimination Running Time

- Compute $\underline{l}_i$: $\deg(i)$ in current graph S_{i-1}
- Compute S_i : $\deg(i)^2$ time
- Even in graph w. $m = O(n)$, $\deg(i)$ can reach n

→ Thus total time n^3 worst case

APPROXIMATE GAUSSIAN ELIMINATION ALGORITHM ??!

$$S_0 = L$$

for $i = 1, \dots, n$

$$\underline{l}_i = \frac{1}{S_{i-1}(i,i)^{1/2}} S(:,i)$$

$$S_i = S_{i-1} - \mathrm{Star}(i, S_{i-1}) + \cancel{\mathrm{Clique}(i, S_{i-1})}$$

CliqueSample(i, S_{i-1}) ?

end

$$\mathcal{L} = [\underline{l}_1, \dots, \underline{l}_n]$$

APPROXIMATE GAUSSIAN ELIMINATION ALGORITHM

$S_0 = L$

$\boxed{\pi \leftarrow \text{random permutation of } 1, \dots, n}$

for $i = 1, \dots, n$

$$\ell_i = \frac{1}{S_{i-1}(\pi(i), \pi(i))^{1/2}} S(:, \pi(i))$$

$$S_i = S_{i-1} - \mathrm{Star}(\pi(i), S_{i-1}) + \mathrm{CliqueSample}(\pi(i), S_{i-1})$$

end

$$\mathcal{L} = [\ell_1, \dots, \ell_n], \ \pi$$

THEOREM Kyng & Sachdeva 2016

We can find $\mathcal{L}\mathcal{L}^T \approx_{0.5} L$

s.t. $\mathcal{L}$ is lower triangular w.r.t.

the elimination ordering π

and $\mathrm{nnz}(\mathcal{L}) = O(m \log^3 n)$.

The algorithm is randomized and runs in time $O(m \log^3 n)$,

and succeeds whp.

$Ax = d$

B^{-1}

$(\mathcal{L}\mathcal{L}^T)^{-1}$

in $m\sqrt{\log n}$ Cohen, Kyng, Pachocki, Peng, Rao (?) … using Expectation

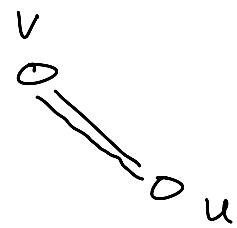
Cohen Kyng ... in Expectation

COROLLARY We can find an ε-approximate solution $\tilde{x}$ to $Lx^* = d$ in time $O(m \log^3 n \log(\frac{1}{\varepsilon}))$ whp.

IDEA Sample to approximate $\mathrm{Clique}(v, S)$.

EDITORIALIZE...

NB: We allow multiedges!



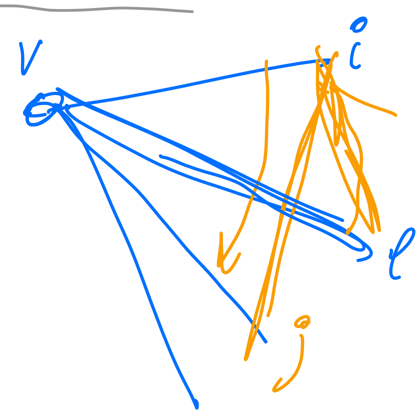
$\sum_e w(e) b_e b_e^T$

ALGO $\mathrm{CliqueSample}(v, S)$

$Y_v \leftarrow 0_{n \times n}$

for each multiedge (v, i)

→ pick a random multiedge (v, j) w. probability $\dfrac{w(v,j)}{\sum_{j \sim v} w(v,j)}$ (allow $j = i$)

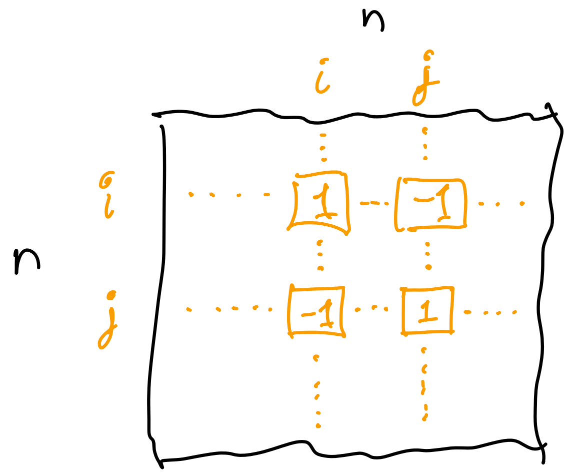


→ If $j \neq i$

$$Y_v \leftarrow Y_v + \frac{w(i,v)\, w(j,v)}{w(i,v) + w(j,v)} b_{ij} b_{ij}^T$$

return Y_v

$\underline{b}_{ij}\,\underline{b}_{ij}^T$:

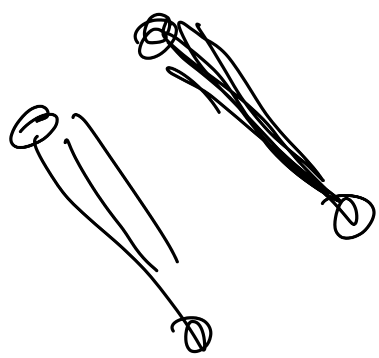


Baby Laplacian
(Four non-zeros)

Q: Can you compute each sample in $O(1)$ time?

A: Yes, with $O(\deg(v))$ preprocessing time.

"Walker's Method" / "The Alias Method"



- I cheated!
- There is one more step to the algorithm.
- Before we start, we split each edge into K multi-edges w. $\frac{1}{K}$ times the original weight.

APPROXIMATE GAUSSIAN ELIMINATION ALGORITHM

Finally ☺

$S_0 \leftarrow L$ with edges split into $K = 10\log^2 n$ copies w. $w(\hat{e}) = \frac{w(e)}{K}$

$\pi \leftarrow$ random permutation of $1, \dots, n$

for $i = 1, \dots, n$

$$\underline{l}_i = \frac{1}{S_{i-1}(\pi(i), \pi(i))^{1/2}} S(:, \pi(i))$$

$$S_i = S_{i-1} - \mathrm{Star}(\pi(i), S_{i-1}) + \mathrm{CliqueSample}(\pi(i), S_{i-1})$$

end

$\mathcal{L} = [\underline{l}_1, \dots, \underline{l}_n], \pi$

ANALYSIS PLAN

1) Sample Count & Running time

$\mathbb{E}\,\tilde{L}$

2) Expected value of output

   A) clique sample

   B) overall : linearity → martingale

3) The Tail Bound from Matrix Trace Exponentials

4) Edge Sample Norms (Individual Samples)

5) Clique Norm & Variance (Vertex Sampling)

PART 1: Sample count & running time

- Observe: S_i has $\leq Km$ multiedges

  Proof: we never create more edges than we remove in each elimination.

- Hence $$\mathbb{E}_{\pi(i)} \deg(\pi(i)) \leq \frac{2Km}{n-(i+1)}$$

- Total expected running time

$$\sum_i \frac{2Km}{n-(i+1)} = O(Km \log n) = O(m \log^3 n)$$

- This also bounds $nnz(\mathcal{L}) = O(m \log^3 n)$

- Exercise: Make running time deterministic

  ~~NB: Not independent.~~

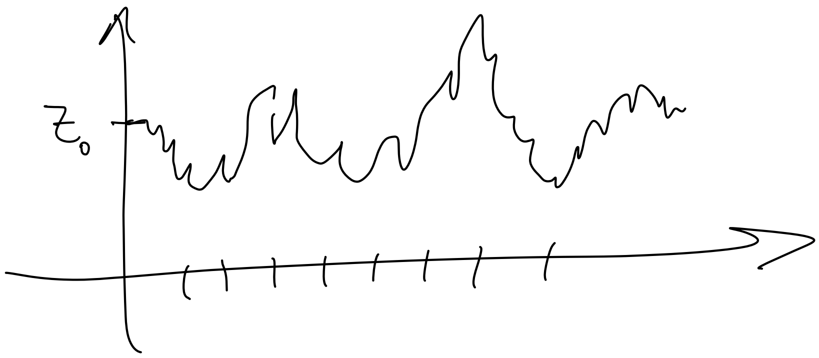
PART 2 : EXPECTATION

LEMMA $\mathbb{E}\,\text{CliqueSample}(v,S) = \text{Clique}(v,S)$

MARTINGALES

A sequence of RVs $z_1, \dots, z_k$ with

$$\mathbb{E}[z_i \mid \text{all variables before: } z_1, \dots, z_{i-1}] = z_{i-1}$$



NB: Both scalar and matrix-valued make sense.

We can also define $X_i = z_i - z_{i-1}$,
then we have $\mathbb{E}[X_i \mid \text{all before}] = 0$

GOAL Show concentration of $z_k = z_0 + \sum_{i=1}^{k} X_i$

A MATRIX MARTINGALE FOR APPROXIMATE G.E.

Let $L_i = S_i + \sum_{j=1}^{i} \ell_i \ell_i^\top$, $S_0 = L$ $\quad L_0 = L$

as created by Apx. G.E.

Then $\mathbb{E}[L_i \mid \text{all vars before}] = L_{i-1}$

Proof : exercise.

COROLLARY $\mathbb{E}\,\mathcal{L}\mathcal{L}^\top = \mathbb{E}L_n = L_0 = L$

PART ③ : Tail Bound from matrix trace exponentials

Algo outputs $\mathcal{L}\mathcal{L}^\top = L_n$ (represented as $\mathcal{L}$)

GOAL $L_n - L \preceq \frac{1}{2} L$ (and other side)

$\Phi(M) = L^{-1/2} M L^{-1/2}$ $\qquad L_n \approx_{\frac{1}{2}} L$

FACT $A \preceq B \iff \Phi(A) \preceq \Phi(B)$ when $\ker(L) \subseteq \ker(A) \cap \ker(B)$

GOAL $\Phi(L_n) - \pi \preceq \frac{1}{2}\pi$, $\pi = I - \frac{\mathbf{1}\mathbf{1}^\top}{n}$

- Break this into zero-mean martingale steps?

- Let $Y_{i,e}$ be the edge sample for the i^{th} vertex elimination and the e^{th} multi-edge sample.

Y_v

$S_i \preceq \sum_{j \leq} \ell_j \ell_j^T$

$\parallel$

L

- $\text{CCliqueSample}(\pi(i), S_{i-1}) = \sum_e Y_{i,e}$

- $X_{i,e} = \Phi(Y_{i,e}) - \mathbb{E}\left[\Phi(Y_{i,e}) \mid \text{all samples before } (i,e)\right]$

CLAIM $L_i = L + \sum_{j \leq i} \sum_e Y_{i,e} - \mathbb{E}[Y_{i,e} \mid \text{all before}]$

proof: Exercise.

CLAIM

$\Pr[L_n - L \not\preceq 1.5 L] \leq e^{-\theta/2}\, \mathbb{E}\, \mathrm{tr}\left(\exp\left(\theta \sum_{i \leq n} \sum_e X_{i,e}\right)\right)$

LEMMA "Nested Moment Trick"

- Suppose that for conditional on $\pi(i)$ and everything before

$$\sum_e \log\left(\mathbb{E}_e \exp(\theta X_{i,e})\right) \preceq V_i \qquad \text{NB: } V_i \text{ random}$$

- Then

$$\mathbb{E}\, \mathrm{tr}\left(\exp\left(\theta \sum_{i \leq n} \sum_e X_{i,e}\right)\right) \leq \mathrm{tr}\left[\exp\left\{\sum_i \log\left(\mathbb{E}_{\pi(i)} \exp(V_i)\right)\right\}\right]$$

Conditional on samples before $\pi(i)$!

$\mathrm{tr}\left(\exp(H + \log(A))\right)$ concave in A

$\mathrm{tr}\left(\exp(H + \log(\mathbb{E}\exp(X)))\right) \geq \mathbb{E}\, \mathrm{tr}\exp(H + X)$

LEMMA $$\sum_e \log\left(\mathbb{E}_e \exp(\theta X_{i,e})\right) \preceq R\theta^2 \Phi\left[\text{clique}(\pi(i), S_{i-1})\right]$$

Provided $\theta R \le 1$

$\forall i$?

LEMMA $$\Phi\left(\text{clique}(\pi(i), S_{i-1})\right) \preceq \Phi(S_{i-1}) \preceq 2I$$

LEMMA $$\Phi\left(\mathbb{E}_{\pi(i)} \text{clique}(\pi(i), S_{i-1})\right) \preceq \Phi\left(\frac{2 S_{i-1}}{n+1-i}\right) \preceq 2I$$

conditional on samples before $\pi(i)$

"LEMMA" (NOT QUITE TRUE)

$$S_i \preceq 2L$$

Putting it together

$$\log\left(\mathbb{E}_{\pi(i)} \exp(V_i)\right)$$

$$\preceq \log\left(\mathbb{E}_{\pi(i)}(I + 2V_i)\right)$$

$$\preceq \log\left(I + \frac{4\,\theta^2 R\,\Phi(S_{i-1})}{n+1-i}\right)$$

$$\preceq \log\left(I + \frac{8\theta^2 R}{n+1-i} I\right)$$

$$\preceq \frac{8\theta^2 R}{n+1-i} I$$

$\exp(A) \preceq I + 2A$

provided $0 \preceq A \preceq I$

V_i

$= R\theta^2 \Phi(\text{clique}(v, S_{i-1}))$

$\preceq R\theta^2 \Phi(S_{i-1})$

$\preceq R\theta^2 \Phi(L)$

$\preceq R\theta^2 I \qquad R\theta^2 \leq 1$

$\log(I + A) \preceq A$

$-I \preceq A \preceq 0$

$$\sum_i \log\left(\mathbb{E}_{\pi(i)} \exp(V_i)\right)$$

$$\preceq 8\theta^2 R \left(\sum_i \frac{1}{n+1-i}\right) I$$

$$= 8\theta^2 R \log n \; I$$

$$\Pr[\ldots] \leq e^{-\theta/2} \cdot n \cdot e^{8\theta^2 R \log n}$$

$\theta R \leq 1$

optimize θ subject to $\theta^2 R \leq 1$

$0 < \theta$.

PART 4 EDGE NORMS

LEMMA Consider two Laplacians L and S with the same dimensions.

If each multiedge e of $\mathrm{Star}(v, S)$ has

$$\| L^{+/2} \underline{w}_S(e) \underline{b}_e \underline{b}_e^{\top} L^{+/2} \| \leq R$$

Then every possible multiedge sample $\hat{e}$ of $\mathrm{CliqueSample}(v, S)$ satisfies

$$\| L^{+/2} \underline{w}_{new}(\hat{e}) \underline{b}_{\hat{e}} \underline{b}_{\hat{e}}^{\top} L^{+/2} \| \leq R$$

Proof : exercise.

$w(e) \underline{b}_e \underline{b}_e^{\top} \preceq L$

$\Phi(\quad) \preceq I$

- Before we start, we split each edge into K multi-edges w. $\frac{1}{K}$ times the original weight.

LEMMA · Original edges have leverage score

$$\| w(e) \Phi(\underline{b}_e \underline{b}_e^{\top}) \| \leq 1.$$

· K-wise split multi-edges have leverage score

$$\| w(e) \Phi(\underline{b}_e \underline{b}_e^{\top}) \| \leq 1/K$$

PART 5 CLIQUE NORM & VARIANCE

LEMMA $\mathrm{Clique}(v,S) \preceq \mathrm{Star}(v,S)$

Pf $\mathrm{Star}(v,S) = \mathrm{Clique}(v,S) + \underline{\ell}\,\underline{\ell}^T \succeq \mathrm{Clique}(v,S)$

$$\left[\begin{array}{c|c} w & (-\underline{a}^T) \\ \hline -\underline{a} & \mathrm{diag}(\underline{a}) \end{array}\right] = \left[\begin{array}{c|c} 0 & \underline{0}^T \\ \hline \underline{0} & \mathrm{diag}(\underline{a}) - \frac{\underline{a}\,\underline{a}^T}{w} \end{array}\right] + \left[\begin{array}{c|c} w & \underline{a}^T \\ \hline -\underline{a} & \frac{\underline{a}\,\underline{a}^T}{w} \end{array}\right]$$

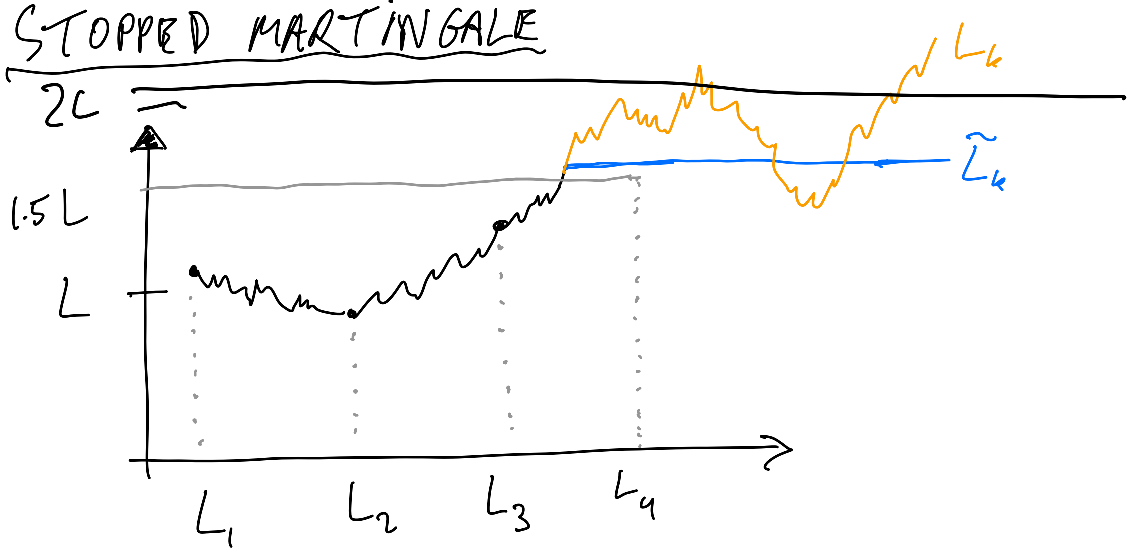
COROLLARY $\mathrm{Clique}(v,S) \preceq S$

LEMMA $\sum_v \mathrm{Star}(v,S) = 2S$

Proof Each edge appears in 2 stars.

COROLLARY $\mathop{\mathbb{E}}_{\pi(i)} \mathrm{Clique}(\pi(i), S_{i-1}) \preceq \dfrac{2S_{i-1}}{n+i-1}$

STOPPED MARTINGALE



$$L_{i,e} = L + \sum_{j \le i} \sum_{f \le e} \varphi_{j,f} - \mathbb{E}[\varphi_{j,f} \mid \text{all before}]$$

$$\tilde{L}_i = \begin{cases} L_i & \text{if} \quad L_{j,f} \lesssim 1.5L \ \text{for all } (j,f) \le (i, e_{last(i)}) \\ L_{i^*,e^*} & \text{o.w. where } i^*, e^* \text{ is the first } i,e \\ & \quad \text{such that } L_{i,e} \not\lesssim 1.5 \end{cases}$$

LEMMA

1) $\tilde{L}_i \lesssim 2L$

2) $0.5L \lesssim \tilde{L}_n \lesssim 1.5L$ (2A)

$\Downarrow$

$0.5L \lesssim L_n \lesssim 1.5L$ (2B)

3) $\{\tilde{L}_i\}$ is a martingale

Conclusion: It suffices to show (2A)

Gaussian elimination as optimization on a Laplacian quadratic form.

$\underline{x} \in \mathbb{R}^V \qquad G = (V, E, \underline{w}), \qquad L$

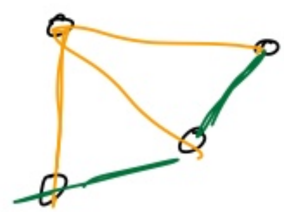
$\underline{d} \in \mathbb{R}^V \qquad \underline{d} \perp \underline{1}$

$$\mathcal{E}(\underline{x}) = -\underline{d}^T \underline{x} + \frac{1}{2} \underline{x}^T L \underline{x} \qquad L\underline{x} = \underline{d}$$

$$L = \begin{pmatrix} W & -\underline{a}^T \\ -\underline{a} & \mathrm{diag}(\underline{a}) + L_{-1} \end{pmatrix}$$

$$\underline{x} = \begin{pmatrix} y \\ \underline{z} \end{pmatrix} \qquad \underline{d} = \begin{pmatrix} b \\ \underline{c} \end{pmatrix}$$

$$\mathcal{E}\begin{pmatrix} y \\ \underline{z} \end{pmatrix} = \underline{-\begin{pmatrix} b \\ \underline{c} \end{pmatrix}^T \begin{pmatrix} y \\ \underline{z} \end{pmatrix} + \frac{1}{2} \begin{pmatrix} y \\ \underline{z} \end{pmatrix}^T L \begin{pmatrix} y \\ \underline{z} \end{pmatrix}}$$



$$\frac{d}{dy} \mathcal{E}\begin{pmatrix} y \\ \underline{z} \end{pmatrix} = 0$$

$$y = \frac{1}{W}\left(\underline{a}^T \underline{z} + b\right)$$

$$\underline{g} = \underline{c} + \frac{b}{W} \underline{a} \qquad S = \underbrace{L_{-1} + \mathrm{diag}(\underline{a}) - \frac{\underline{a}\underline{a}^T}{W}}$$

$$\mathcal{E}\begin{pmatrix} y \\ \underline{z} \end{pmatrix} \text{ w. } y = \frac{1}{W}\left(\underline{a}^T \underline{z} + b\right)$$

$$\min_y \mathcal{E}\begin{pmatrix} y \\ \underline{z} \end{pmatrix} = -\underline{g}^T \underline{z} + \frac{1}{2} \underline{z}^T S \underline{z} - \frac{1}{2} \frac{b^2}{W}$$

CLAIM 1) $\underline{g}^T \underline{1} = 0$ when $\underline{d}^T \underline{1} = 0$

2) S is a graph Laplacian