An $\tilde{O}(m \log^2 n \log \frac{1}{\varepsilon})$ Laplacian solver

(Koutis/Miller/Peng 2010)

CS 292F
June 2, 2021
Lecture 17

Solve $Gx = b$ (weighted graph Laplacian.)

m edges, n vtxs

Vaidya 1991: MST, tree + a few extra edges, GE.

$O(m^{1.75})$

Boman-Hendrickson 2003: Low-stretch tree. (no extra edges)

$O(m^{1.33})$

Spielman-Teng 2004: Recursive precond. Spectral sparsifiers.

$O(m \log^{15} n)$

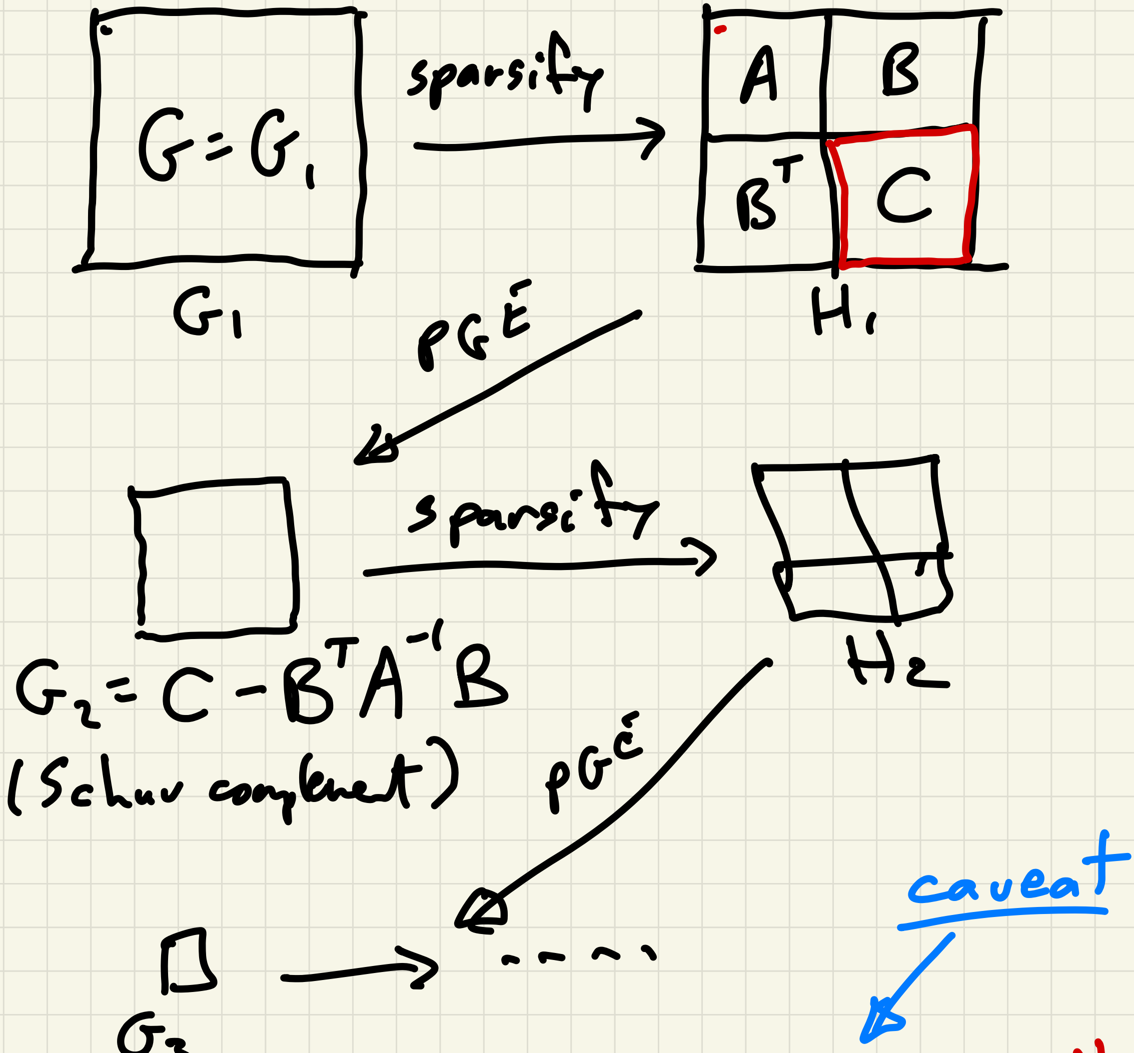
Koutis/Miller/Peng 2010:

$O(m \log^2 n \log\log^c n)$

KMP Key Ideas

1. Recursive preconditioning
2. (Partial) Gaussian elimination (PGE)
3. (Incremental) Sparsifiers.

Recursive precond. $\qquad Gx = b$



To solve with G_1, precondition CG with H_1.

To solve with H_1, solve recursively with G_2.

PGE

Solve $Hx = b$ $\quad H\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

$$H_1 = \left[\begin{array}{c|c} A & B \\ \hline B^T & C \end{array}\right] = \underbrace{\left[\begin{array}{c|c} L_1 & 0 \\ \hline L_2 & I \end{array}\right]}_{L} \left[\begin{array}{c|c} I & 0 \\ \hline 0 & G_2 \end{array}\right] \underbrace{\left[\begin{array}{c|c} L_1^T & L_2^T \\ \hline 0 & I \end{array}\right]}_{L^T}$$

$$G_2 = C - B^T A B$$

Solve:

① Solve $L\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$: $\quad L_1 y_1 = b_1$ (fwd subst), $\quad y_2 = b_2 - L_2 b_1$

② Solve $\left[\begin{array}{c|c} I & 0 \\ \hline 0 & G \end{array}\right]\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$: $\quad z_1 = y_1$ and $G_2 z_2 = y_2$ (recursion)

③ Solve $L^T\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$:

$$x_2 = z_2$$

and $L_1^T x_1 = z_1 - L_2^T z_2$ (back subst.)

TIME:

$O(nnz(L))$ + recursive call

KEY: make $nnz(L)$ small.

INCREMENTAL SPARSIFIER (IS):

H_1 will be a spanning tree of G, plus a <u>few</u> extra edges (with weights scaled cleverly)

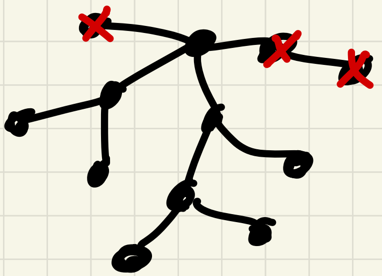
PGE:

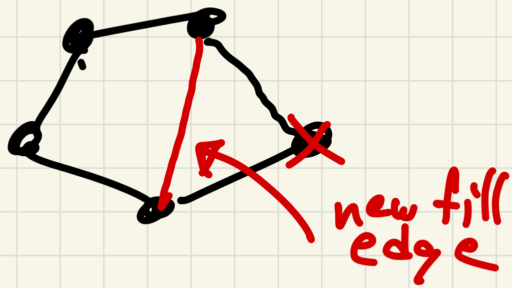
while H_1 has a vertex of degree 1 or 2
- eliminate it (one less vtx and one less edge)

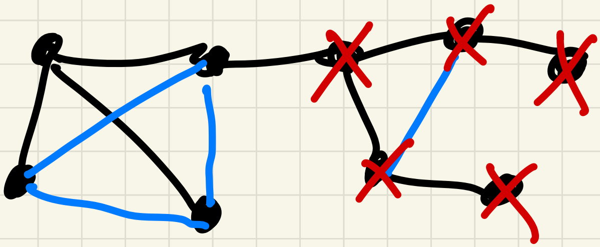
A = eliminated vtxs

C = remaining vertices

$G_2 = C - B^T A^{-1} B$ (Schur compl)







LEMMA

If H_i has $m = \underline{n-1} + \underline{j}$ edges

then $G_{i+1} = PGE(H)$ has

$\hat{n} \leq 2j-2$ vtxs & $\hat{m} \leq 3j-3$ edges.

PROOF

when $PGE(H)$ stops

$$\hat{m} - \hat{n} \leq j-1$$

and $\hat{m} \geq \frac{3\hat{n}}{2}$ so

$$\frac{3\hat{n}}{2} - \hat{n} \leq j-1 \implies \hat{n} \leq 2j-2$$

and $\hat{m} \leq \hat{n} + j - 1 \leq 3j-3$.

I.S. :

We saw a sparsifier that sampled edges of G with probability ~ leverage score $l_e = c(e) R_e^{eff}$.

$H = \text{sample}(G)$

<u>repeat</u> q <u>times</u>:

- sample $e \in E(G)$ with probability $p_e = \alpha\, c_G(e)\, R_G^{eff}(e)$
- add edge e to H with $c_H(e) = c_G(e)/p_e$

<u>Idea</u>: It's good enough to take any $p_e \geq \alpha\, c_G(e)\, R_G^{eff}(e)$. H is at least as good an approx, <u>but</u> might have more edges.

__IDEA__: Get a low-stretch tree $T \subseteq G$.

Then use $R^{eff}_{G}(e) \leq R^{eff}_{T}(e)$

$H = IS(G)$:

$$\begin{cases} T = \text{low-stretch tree}(G) \times K \\ H = \text{sample}(G, p_e = \alpha \cdot \text{stretch}_T(e)) \times 2 \end{cases}$$

Recall $\text{stretch}_T(e) = c_G(e) R^{eff}_T(e) = \dfrac{R^{eff}_T(e)}{r_G(e)}$

So T is low-stretch $\Rightarrow$

$C_G(e) R^{eff}_T(e)$ is not too much more than $C_G(e) R^{eff}_G(e)$.

__THM__: For any $K < m$, $H = IS(G)$ computes an H with $G \preceq H \preceq 3KG$ and H has $n-1 + O(\frac{m}{K} \log^2 n)$ edges.

SOLVER: ① Build chain of G_i, H_i's
② Recursive solve

BUILDCHAIN(G):

$G_1 = G$

for $i = 1, 2, \ldots$

$n_i := \#\text{vtxs}(G_i)$

$k_i \approx \Theta(\log^4 n_i)$

$H_i = \text{INC\_SPFY}(G_i, k_i)$

$G_i = \text{PGE}(H_i)$

Total time for iterations at all levels

is $O(m \log^2 n \log\log^c n \log \frac{1}{\varepsilon})$

independant of the λ's.

SINCE THEN: at least 2 quite different $\tilde{O}(m)$

and improvements to this $O(m \log^{1/2} n \log\log^c \log \frac{1}{\varepsilon})$