Preconditioning

Next Lecture: time $\tilde{O}(m \lg n \lg 1/\varepsilon)$
today $\tilde{O}(m^{4/3} \lg n \lg 1/\varepsilon)$

Solve equations in A by solving systems in a "preconditioner" B, and multiplying by A.

CG, Richardson, Chebyshev just used mults by A.

\# of iterations depends on $\kappa(A,B) = \dfrac{\lambda_{max}(B^{-1}A)}{\lambda_{min}(B^{-1}A)}$,

orthogonal to common nullspace, if there is one.

Lem $\lambda_{max}(B^{-1}A) = \min\{\beta : A \preccurlyeq \beta B\}$

$\lambda_{min}(B^{-1}A) = \max\{\alpha : \alpha B \preccurlyeq A\}$

proof $A \preccurlyeq \beta B$ iff $\forall x \quad x^T A x \leq \beta x^T B x$

$$\iff \frac{x^T A x}{x^T B x} \leq \beta$$

$$\iff \frac{y^T B^{-1/2} A B^{-1/2} y}{y^T y} \leq \beta$$

$$\iff \lambda_{max}(B^{-1/2} A B^{-1/2}) = \lambda_{max}(B^{-1}A) \leq \beta$$

So, $\kappa(A,B) \leq \beta/\alpha$ where $\alpha B \preccurlyeq A \preccurlyeq \beta B$

Cor. $A \preccurlyeq B$ iff $B^{-1} \preccurlyeq A^{-1}$

Recall, $\tilde{x}$ is ε-approx sol to $Ax = b$ if

$$\|\tilde{x} - x\|_A \leq \varepsilon \|A\|.$$

A very good precon would be B s.t. $(1-\varepsilon) B \preccurlyeq A \preccurlyeq (1+\varepsilon) B$

By cor, and a result from last lecture,

$$\|B^{-1}Ax - x\|_A \leq \varepsilon \|A\| \text{ for all } x.$$

Iterative Refinement is a procedure for increasing accuracy.

Set $x_1 = B^{-1}b = B^{-1}Ax$

residual $r_1 = b - Ax_1$

If $Ay = r_1$, then $A(x_1 + y) = Ax_1 + b - Ax_1 = b$

So, set $x_2 = B^{-1} r_1$

Now, $\|x - (x_1 + x_2)\|_A = \|(x - x_1) - x_2\|_A$

$$\leq \varepsilon \|(x - x_1)\|_A$$

$$\leq \varepsilon^2 \|x\|_A$$

If repeat k times, can get an ε^k approx solution

after k iterations

each using a mult by A

solve in B

So, once can get an ε-approx for some constant ε

can achieve $\ln 1/\varepsilon$ dependence

Origin: going from low to high precision.

Most preconditioners are not that good.

But, if can solve B quickly, then apply in CG or Chebyshev.

Let $\lambda_1 \dots \lambda_n$ be eigs of $B^{-1}A$ = eigs of $A^{1/2}B^{-1}A^{1/2}$

If $q(x) = 1 - x p(x)$ is a poly s.t. $|q(\lambda_i)| \le \varepsilon \ \forall i$,

then $\tilde{x} \triangleq p(B^{-1}A) B^{-1} b$

is an ε-approx solution to $Ax = b$

<u>proof:</u>

$$\|x - \tilde{x}\|_A = \|A^{1/2}x - A^{1/2}\tilde{x}\|$$
$$= \|A^{1/2}x - A^{1/2}p_t(B^{-1}A)B^{-1}Ax\|$$
$$= \|A^{1/2}x - A^{1/2}p_t(B^{-1}A)B^{-1}A^{1/2} \cdot A^{1/2}x\|$$
$$\le \|I - A^{1/2}p_t(B^{-1}A)B^{-1}A^{1/2}\| \cdot \|x\|_A$$

$$I - A^{1/2}p_t(B^{-1}A)B^{-1}A^{1/2} = I - p_t(A^{1/2}B^{-1}A^{1/2})A^{1/2}B^{-1}A^{1/2}$$
$$= q_t(A^{1/2}B^{-1}A^{1/2})$$

As $A^{1/2}B^{-1}A^{1/2}$ is symmetric,

$$\|q(A^{1/2}B^{-1}A^{1/2})\| \le \varepsilon$$

Preconditioned CG - PCG finds the best polynomial.
But, because polynomial depends on input, is not a linear operator.

Precon Cheby - needs to know α and β, but is a linear operator.

Laplacians

Vaidya: precondition L_G by L_H for $H \subseteq G$

Guarantees $L_H \preccurlyeq L_G$

Just need to know β s.t. $L_G \preccurlyeq \beta L_H$

Use a spanning tree T.

Can solve equations in L_T in time $O(n)$ -

elimination takes time $O(n)$ because

every tree has a vertex of degree 1,

and eliminating it gives another tree.

Will show is a tree T s.t. $\beta \leq c \cdot m \cdot \lg n \cdot \lg\lg n$,

for some constant c.

Now $\lambda_{max}(L_H^+ L_G) \leq Tr[L_H^+ L_G]$

lem. $Tr[L_H^+ L_G] = \sum_{(a,b) \in E} w_{a,b} \, Stretch_T(a,b)$

where view length of an edge as $\frac{1}{weight}$

$stretch_T(a,b)$ = length of path in T from a to b

mult by $w_{a,b}$ = divide by length of (a,b) edge proof

Write $L_G = \sum_{a \sim b} w_{a,b} (\delta_a - \delta_b)(\delta_a - \delta_b)^T$

$$\mathrm{Tr}\left[L_{T}^{+} L_G\right] = \sum_{a \sim b} w_{a,b} \, \mathrm{Tr}\left[L_{T}^{+} (\delta_a - \delta_b)(\delta_a - \delta_b)^T\right]$$

$$= \sum_{a \sim b} w_{a,b} (\delta_a - \delta_b)^T L_{T}^{+} (\delta_a - \delta_b)$$

$$= \sum_{a \sim b} w_{a,b} \, \mathrm{Reff}_T(a,b)$$

$\mathrm{Reff}_T(a,b) = \sum \frac{1}{w_i}$ sum over weights of edges on the unique path in T between a and b.

$$= \sum_{a \sim b} w_{a,b} \, \mathrm{stretch}_T(a,b)$$

Theorem Every weighted graph $G = (V, E, w)$ has a spanning Tree $T = (V, F, w)$ with $F \subseteq E$

s.t. $\sum_{(a,b) \in E} w_{a,b} \, \mathrm{stretch}_T(a,b) \le O(m \lg n \lg \lg n)$

And, can compute it in time $O(m \lg n \lg \lg n)$

$= \mathrm{stretch}_T(G)$

Given T, can compute $\mathrm{stretch}_T(G)$, use it to set β, and run Precon Cheby for $\leq O(\sqrt{\beta}\lg 1/\varepsilon)$ iterations to get ε-accurate solution.

Each iteration takes time $O(m)$ - mult by L_G

$O(n)$ - solve in L_H

so total is $O(m^{3/2}\sqrt{\lg n \lg\lg n}\,\lg 1/\varepsilon)$

Using PCG can do better, as there is a better polynomial.

Reason: there can be at most k eigenvalues $> \mathrm{Tr}(L_H^+ L_G)/k$

<u>Lem</u> let $1 \leq \lambda_1 \leq \dots \leq \lambda_n$

and $\lambda_{n-k} \leq \beta$.

Then for every $t \geq k$, $\exists$ poly $q(x)$ of degree t s.t.

$q(0) = 1$ and $|q(\lambda_i)| \leq 2\exp(-2(t-k)/\sqrt{\beta})$

use with $\beta = \mathrm{Tr}(L_H^+ L_G)^{2/3}$, $k = \mathrm{Tr}(L_H^+ L_G)^{1/3}$,

$t = (k+1)\ln(1/\varepsilon)$ to get ε-accurate solution

<u>proof</u> Let $r(x)$ be poly of degree $t-k$ s.t.

$r(0)=1$, $|r(x)| \leq 2\exp(-2(t-k)/\sqrt{\beta})$ $\forall$ $1 \leq x \leq \beta$

constructed last week.

Set $q(x) = \tau(x) \prod_{j=n-k+1}^{n} \left(1 - \frac{x}{\lambda_j}\right)$

so $q(0) = \tau(0) = 1$

$q(\lambda_i) = 0 \quad i > n-k$

for $i \leq n-k, \quad \lambda_i \leq \beta$

so $q(\lambda_i) = \underbrace{\tau(\lambda_i)}_{\leq \varepsilon} \prod_{j=n-k+1}^{n} \underbrace{\left(1 - \frac{\lambda_i}{\lambda_j}\right)}_{\in (0,1)}$

$|q(\lambda_i)| \leq \varepsilon$