Welcome to Ma221! Lecture 21, Fall24

Chap 6: Iterative Methods for $Ax=b$
(and $Ax=\lambda x$ Chap 7)

Model Problem: Poisson Eqn (see Lecture 10)

Goals: Contrast direct + iterative methods

- for $Ax=b$ or least squares: Use iterative methods when direct method too slow, or uses too much memory, or you need less accuracy
- for $Ax=\lambda x$ or SVD: same reasons as above, or when you need only a few evals/evecs

Choosing best iterative method depends on math. structure of $A \Rightarrow$ large diversity of algs and software (lots of links on class webpage)

Illustrate using Model Problem (Poisson)
arises in electrostatics, heat flow, quantum mechanics, fluid mechanics, graph theory

Present methods from simple to complicated,
(simple ones are building blocks for complicated ones)

Methods covered

(1) "Splitting Methods" for $Ax=b$

"Split" $A = M - K$, M nonsingular

so $Ax = b$ becomes $Mx = Kx + b$, given x_0

Iterate: $M x_{i+1} = K x_i + b$ ($\Rightarrow M$ "simple")

Convergence:
$$
\begin{aligned}
M x_{i+1} &= K x_i + b \\
M x &= K x + b \\
\hline
M(x_{i+1} - x) &= K(x_i - x) \qquad e_i = x_i - x \\
e_{i+1} &= M^{-1} K e_i \\
&= (M^{-1}K)^{i+1} e_0
\end{aligned}
$$

How fast does $(M^{-1}K)^i \to 0$?

Consider 3 Methods:

Jacobi, Gauss-Seidel,
Successive Overrelaxation (SOR)

eg: Jacobi $\Rightarrow M = \text{diag}(A)$

(2) Krylov Subspace Methods (KSMs)

What can you do if all you can do is multiply $A \cdot x$ for any x?

Overview of how KSMs: Given x_0

1) $x_1 = Ax_0$, $x_2 = Ax_1$, ... $x_k = Ax_{k-1}$
   (or a similar set, spanning same subspace)

2) Choose a linear combination of these vectors that gives "best" solution in that space (for some definition of "best")

3) If approx not good enough after k steps, increase k and repeat

Depending on
- How x_i are computed
- Properties of A (eg sym, spd, ...)
- def of "best"

lots of methods, all used in practice, will cover

any A: GMRES (Generalized Minimum Residual)
spd A: Conjugate Gradients (CG)

Same idea for EG, $Ax = \lambda x$ (Chap 7)

Lanczos, Arnoldi, .. in chap 7

(3) Preconditioning: How fast do these algs converge?
Complicated, but in general depends on $\kappa(A)$: faster if $\kappa(A)$ smaller

Suppose we can find a matrix M where

(1) Multiplying by M cheap

(2) MA better conditioned than A

$\Rightarrow$ apply all methods to $(MA)x = Mb$

$(AM)(M^{-1}x) = b$: maybe harder, need M^{-1}

Finding good M depends on structure of A

eg: conjugate gradients MA not symmetric, but CG can still work

(4) Multigrid: Most effective preconditioner apply to Poisson:

Idea: if A arises from approximating some physical problem, then "straight forward" to find a "coarse" approx = smaller approx to approximate solution

Eg: Poisson on 2D mesh $n \times n$

$\Rightarrow n^2$ unknowns

Approximate Poisson by $\frac{n}{2} \times \frac{n}{2}$ mesh use smaller solution as approximation, but $\left(\frac{n}{2}\right)^2$ unknowns still large

Use same idea recursively

$\Rightarrow$ Solve Poisson in $O(\#\text{unknowns})$

= lower bound

(5) Domain Decomposition:

How to hybridize all above methods, using different methods in different domains ~ submatrices, depending on which is fastest

How to optimize communication:

simple implementation of $A \cdot x$ memory bound

Lots of recent work on improving this:

Can multiply $x_1 = A x_0,\ x_2 = A x_1, \ldots, x_k = A x_{k-1} = A^k x_0$

but only read A once from memory:

assumes A sparse enough, some structure

Model Problem: Poisson Eqn

1D: Discrete ODE with Dirichlet Boundary Conditions

$$\frac{d^2}{dx^2} v(x) = f(x) \quad \text{on} \quad x \in [0,1]$$

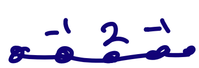
$$v(0) = v(1) = 0$$

discretize (Lecture 10) to get

$$T_N \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} = T_N v = h^2 \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} = h^2 f \qquad h = \frac{1}{N+1}$$

$h = \frac{1}{5},\ N = 4$

stencil:



$$T_N = \begin{bmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ 0 & & \ddots & \ddots & \ddots \end{bmatrix}$$

Evals and evecs of T_N

Lemma: $T_N \cdot z_j = \lambda_j z_j$ where $\|z_j\|_2 = 1$

$$\lambda_j = 2\left(1 - \cos \frac{\pi j}{N+1}\right)$$

$$z_j(k) = \sqrt{\frac{2}{N+1}} \sin(j \cdot k \cdot \pi / (N+1))$$

proof: (HW Q6.1)

Corollary: Z: $Z_{jk} = \sqrt{\frac{2}{N+1}} \sin(j \cdot k \cdot \pi / (N+1))$

orthogonal

closely related to FFT

Evals range from $\lambda_j \sim \left(\frac{\pi j}{N+1}\right)^2$ for j small

to $\lambda_N \sim 4$

$$\Rightarrow \text{cond}\,\# = \frac{\lambda_N}{\lambda_1} \approxeq \left(\frac{2(N+1)}{\pi}\right)^2 = \left(\frac{2}{\pi}\right)^2 h^{-2}$$

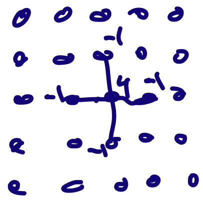
2D Poisson with Dirichlet Boundary Cond.

$$-\frac{\partial^2 v(x,y)}{\partial x^2} - \frac{\partial^2 v(x,y)}{\partial y^2} = f(x,y) \quad \text{on } [0,1]^2$$

$$v(x,y) = 0 \text{ on boundary}$$

discretize as above: $V_{ij} = v(i \cdot h, j \cdot h)$ $\quad h = \frac{1}{N+1}$

$$(*)\quad 4v_{ij} - \overset{\text{above}}{v_{i-1,j}} - \overset{\text{below}}{v_{i+1,j}} - \overset{\text{left}}{v_{i,j-1}} - \overset{\text{right}}{v_{i,j+1}} = h^2 f_{ij}$$



stencil

$V =$ $N \times N$ matrix of unknowns

$$(*)\quad \begin{cases} 2v_{ij} - \overset{\text{above}}{v_{i-1,j}} - \overset{\text{below}}{v_{i+1,j}} = (T_N \cdot V)_{ij} \\ 2v_{ij} - \overset{\text{left}}{v_{i,j-1}} - \overset{\text{right}}{v_{i,j+1}} = (V \cdot T_N)_{ij} \end{cases}$$

$$(*)\quad T_N \cdot V + V \cdot T_N = h^2 F$$

N^2 equations in N^2 unknowns

Sylvester equation (Q4.6)

Evals and Evecs of 2D Poisson

$$T_N V + V T_N = \lambda V$$

$\lambda =$ eval, $V =$ "evec"

let $V = z_i \cdot z_j^T$ where $T_N \cdot z_i = \lambda_i z_i$

$$\begin{aligned} T_N V + V T_N &= T_N(z_i z_j^T) + (z_i z_j^T) T_N \\ &= (T_N z_i) z_j^T + z_i (z_j^T T_N) \\ &= (\lambda_i z_i) z_j^T + z_i (\lambda_j z_j^T) \\ &= (\lambda_i + \lambda_j)(z_i z_j^T) \\ &= \underset{\text{eval}}{(\lambda_i + \lambda_j)} \underset{\text{"evec"}}{V} \end{aligned}$$

$\exists$ eval, evec for all pairs i,j

Want to express V as vector to generalize to higher dimensions

3×3 case: V columnwise, left to right

$$\mathrm{vec}(V) = v = \begin{bmatrix} v_{11} \\ v_{21} \\ v_{31} \\ v_{12} \\ \vdots \\ v_{33} \end{bmatrix}$$

want $\mathrm{vec}(T_N V + V T_N)$

$= T_{N \times N} \cdot \mathrm{vec}(V)$

multiply $v_{i\pm1,j}$

$$T_{N \times N} = \left[\begin{array}{ccc|ccc|ccc} 4 & -1 & & -1 & & & & & \\ -1 & 4 & -1 & & -1 & & & & \\ & -1 & 4 & & & -1 & & & \\ \hline -1 & & & 4 & -1 & & -1 & & \\ & -1 & & -1 & 4 & -1 & & -1 & \\ & & -1 & & -1 & 4 & & & -1 \\ \hline & & & -1 & & & 4 & -1 & \\ & & & & -1 & & -1 & 4 & -1 \\ & & & & & -1 & & -1 & 4 \end{array}\right]$$

multiply $v_{i,j\pm1}$

$$= \left[\begin{array}{c|c|c} T_N + 2I_N & -I_N & \\ \hline -I_N & T_N + 2I_N & -I_N \\ \hline & -I_N & T_N + 2I_N \end{array}\right]$$

generalize to larger N, higher dimensions, using Kronecker Product

Def: $X^{m \times n}$ then $\mathrm{vec}(X)$ defined as $m \cdot n \times 1$ vector gotten by stacking columns of X on top of each other left to right

Matlab: reshape$(X, m \cdot n, 1)$

Def: given $A^{m\times n}$, $B^{p\times q}$

$A \otimes B$ is $m\cdot p$ by $n \cdot q$

$$\begin{bmatrix} A_{11}\cdot B & A_{12}\cdot B & \cdots & A_{1n}\cdot B \\ A_{21}\cdot B & & & \\ \vdots & & & \\ A_{m1}\cdot B & & & A_{mn}\cdot B \end{bmatrix}$$

is Kronecker product of A and B

Matlab: kron(A, B)

Lemma: $A^{m\times m}$, $B^{n\times n}$, $X^{m\times n}$

1) $\mathrm{vec}(A\cdot X) = (I_n \otimes A)\cdot \mathrm{vec}(X)$

2) $\mathrm{vec}(X\cdot B) = (B^T \otimes I_m)\cdot \mathrm{vec}(X)$

3) 2D Poisson: $T_N \cdot V + V\cdot T_N = F$

can be written

$(I_N \otimes T_N + T_N \otimes I_N)\cdot \mathrm{vec}(V) = \mathrm{vec}(F)$

proof: 1) $I_n \otimes A = \mathrm{diag}(\overbrace{A, A, \ldots, A}^{n})$

$$(I_n \otimes A)\cdot \mathrm{vec}(x) = \begin{bmatrix} A & & & \\ & A & & \\ & & \ddots & \\ & & & A \end{bmatrix} \begin{bmatrix} X(:,1) \\ X(:,2) \\ \vdots \\ X(:,n) \end{bmatrix} = \begin{bmatrix} AX(:,1) \\ \vdots \\ AX(:,n) \end{bmatrix}$$

$= \mathrm{vec}(A\cdot X)$

2) similar (HW Q6.4)

3) 2D Poisson: Apply 1) to $T_N \cdot V$
2) to $V \cdot T_N$ $\quad T_N = T_N^T$

$$I_N \otimes T_N + T_N \otimes I_n$$

$$= \begin{bmatrix} T_N & & \\ & \ddots & \\ & & T_N \end{bmatrix} + \begin{bmatrix} 2 \cdot I_N & -I_N & & \\ -I_N & 2 \cdot I_N & \ddots & \\ & \ddots & \ddots & \ddots \\ & & \ddots & \end{bmatrix} = \begin{bmatrix} T_N + 2I_n & -I_N & & \\ -I_N & \ddots & \ddots & \\ & \ddots & \ddots & \ddots \\ & & \ddots & \end{bmatrix}$$

T.

Lemma (HW Q6.4)

1) assume $A \cdot C$ and $B \cdot D$ well defined

$$(A \otimes B) \circ (C \otimes D) = (A \cdot C) \otimes (B \cdot D)$$

2) A and B invertible $\Rightarrow$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

3) $$(A \otimes B)^T = A^T \otimes B^T$$

Prop: $T = Z \cdot \Lambda \cdot Z^T$ be eigen decomp of T
$N \times N$ symmetric

Then eigen decomp of

(*) $(I_N \otimes T) + (T \otimes I_N) =$

$$\underset{\text{orthog}}{(Z \otimes Z)} \underset{\text{diagonal with}}{(I_N \otimes \Lambda + \Lambda \otimes I_N)} \underset{\text{orthog}}{(Z^T \otimes Z^T)}$$

$((i-1)N+j)$th diagonal
$= \lambda_i + \lambda_j$

$Z \otimes Z$ orthogonal with $((i-1)N+j)$th column $= z_i \otimes z_j$

proof: multiply out (*)

$$= (Z \cdot I_n \otimes Z \cdot \Lambda + Z \cdot \Lambda \otimes Z \cdot I_N)(Z^T \otimes Z^T)$$

$$= (Z \cdot I_n \cdot Z^T \otimes Z \cdot \Lambda \cdot Z^T + Z \cdot \Lambda \cdot Z^T \otimes Z \cdot I \cdot Z^T)$$

$$= (I_N \otimes T_N + T_N \otimes I_N)$$

$$= T_{N \times N}$$

Poisson in 3 (or higher) dimensions

$$T_{N \times N \times N} = (T_N \otimes I_N \otimes I_N + I_N \otimes T_N \otimes I_N + I_N \otimes I_N \otimes T_N)$$

$$= \underbrace{(Z \otimes Z \otimes Z)}_{\text{orthog}} \underbrace{(\Lambda \otimes I_N \otimes I_N + I_N \otimes \Lambda \otimes I_N + I_N \otimes I_N \otimes \Lambda)}_{\text{diagonal}} \cdot (Z^T \otimes Z^T \otimes Z^T)$$

N^3 evals of form $\lambda_i + \lambda_j + \lambda_k$ for all triples (i,j,k)

same condition number in all dimensions