Welcome to Ma221! (Apr 7)

Model Problem: Poisson Equation

1D: Discretize ODE with Dirichlet boundary conditions

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

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

Recall Lecture 10: discretize to get

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

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

stencil 

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: trig (HW Q6.1)

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

is orthogonal

closely related to FFT

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

for small j, large N up to $\lambda_N \sim 4$

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

2D Poisson with Dirichlet Boundary Cond.

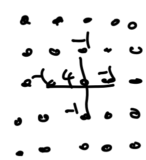
$$-\frac{\partial^2 v(x,y)}{\partial x^2} - \frac{\partial^2 v(x,y)}{\partial y^2} = f(x,y) \text{ on } [0,1]^2$$

with $v(x,y) = 0$ on boundary

discretize as before:

$$v_{ij} = v(i \cdot h, j \cdot h) \qquad h = \frac{1}{N+1}$$

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

2D stencil 

$V =$ $N \times N$ matrix of v_{ij} unknowns

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

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

N^2 equations in N^2 unknowns

Sylvester Equation (Q 4.6)

Evals and Evecs of 2D Poisson

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

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

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

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

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

Want to express V as vector to generalize to higher dimensions

3×3 case: write V columnwise, left to right

$$v = [V_{11}, V_{21}, V_{31}, V_{12}, \ldots V_{33}]^T$$

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

multiply $V_{i,j\pm1}$

multiply $V_{i\pm1,j}$

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

generalize to larger N, higher dimensions using Kronecker Product

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

(Matlab: reshape(X, m·n, 1))

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

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

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

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) $I_n \otimes A = \mathrm{diag}(\underbrace{A, A, \cdots A}_{n})$

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

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

2) similar: HW Q6.4

3) Apply 1) to $T_N\cdot V$ and
2) to $V\cdot T_N$ $(T_N = T_N^T)$

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

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

Lemma (HW Q6.4)

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

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

2) A and B invertible then

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

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

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

Then eigen decomp of

$$I_N \otimes T + T \otimes I_N =$$

(*) $$(Z \otimes Z) \underbrace{(I_N \otimes \Lambda + \Lambda \otimes I_N)}_{\substack{\text{diagonal with} \\ (i-1)N+j \text{ entry is } \lambda_i + \lambda_j}} (Z^T \otimes Z^T)$$

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

proof: multiply out (*)

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

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

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

Poisson equation in 3 (or any) D

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

with eigenvalue matrix Λ

$$(\Lambda_N \otimes I_N \otimes I_N) + (I_N \otimes \Lambda_N \otimes I_N) + (I_N \otimes I_N \otimes \Lambda_N)$$

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

eigenvector matrix $Z \otimes Z \otimes Z$

Solving Poisson with FFT

FFT direct method, not iterative

Start with 2D Poisson

$$T_N \cdot V + V \cdot T_N = F$$

$$T_N = Z \cdot \Lambda \cdot Z^T$$

$$Z^T\left((Z \cdot \Lambda \cdot Z^T \cdot V + V \cdot Z \cdot \Lambda \cdot Z^T) = F\right) Z$$

$$\Lambda (Z^T V Z) + (Z^T V Z) \cdot \Lambda = Z^T F Z$$

$$\Lambda V' + V' \cdot \Lambda = F'$$

Λ diagonal $\Rightarrow$ diagonal Sylvester Eqn

$$(\Lambda V')_{ij} + (V'\Lambda)_{ij} = (F')_{ij}$$

$$\lambda_i V'_{ij} + \lambda_j V'_{ij} = F'_{ij}$$

$$(*) \quad V'_{ij} = F'_{ij} / (\lambda_i + \lambda_j)$$

main cost

1) compute $F' = Z^T F Z$

2) solve $(*)$ for V'

3) compute $V = Z V' Z^T$

if 1) and 3) done using dense matmul

cost $= O(N^3)$, but can do it

in $O(N^2 \log N)$ using FFT

using relationship between Z and FFT

(Z = imaginary part of FFT $\Rightarrow$

all optmizations for FFT work for Z)

$$FFT(i,j) = e^{2\pi\sqrt{-1}\, i \cdot j / N}$$

Extend to higher dimensions using

Kronecker product:

$$(I_N \otimes T_N + T_N \otimes I_N)^{-1}$$

$$= ((Z \otimes Z)(I_N \otimes \Lambda + \Lambda \otimes I_N)(Z^T \otimes Z^T))^{-1}$$

$$= (Z \otimes Z)(\quad \text{diagonal} \quad)^{-1}(Z^T \otimes Z^T)$$

for 3D poisson:

$$(Z \otimes Z \otimes Z)(\text{diagonal})^{-1}(Z^T \otimes Z^T \otimes Z^T)$$

entries $\lambda_i + \lambda_j + \lambda_k$

note: cond(d dimensional Poisson)

$$= \frac{\lambda_{max}}{\lambda_{min}} = \frac{d \cdot \lambda_{max}(1D)}{d \cdot \lambda_{min}(1D)} = \frac{\lambda_{max}(1D)}{\lambda_{min}(1D)}$$

Summary of Performance of all algorithms on Poisson in 2D and 3D

Count # flops (all $O(\cdot)$ sense)

Memory needed (# words)

parallel steps on "perfect" parallel computer with as many processors as needed

Procs needed

(See CS267 for practical parallel algorithms)

Entries sorted in 2 orders:

from slowest to fastest on Poisson

(roughly) from most general to most specific (for Poisson)

"Explicit Inverse" = we compute and store A^{-1} ahead of time, don't count cost of A^{-1}, just multiplying by it

SOR = Successive Overrelaxation

SSOR/Chebyshev = Symmetric SOR with Chebyshev Acceleration

FFT = Fast Fourier Transform

BCR = Block Cyclic Reduction

Lower Bound = assume 1 flop per component of answer

SpMV = Sparse-Matrix-Multiply
cost(SpMV) = # nonzeros

For 2D Poisson on $n \times n$ mesh
$N = n^2$ = # unknowns

3D Poisson on $n \times n \times n$ mesh
$N = n^3$ = # unknowns

If table entry for 2D and 3D different, then 3D in parentheses

All entries in $O(\cdot)$ sense

Method	direct or iterative	#flops	Mem	#Parallel steps	#Procs
Dense Cholesky any spd matrix	D	N^3	N^2	N	N^2
Explicit Inverse anymatrix	D	N^2	N^2	$\log N$	N^2
Band Cholesky	D	N^2 $(N^{7/3})$	$N^{3/2}$ $(N^{5/3})$	N N	N $(N^{4/3})$

works on any band matrix, $\text{cost} = O(N \cdot bw^2)$

2D: $bw = n = N^{1/2}$

3D: $bw = n^2 = N^{2/3}$

Jacobi	I	N^2 $(N^{5/3})$	N N	N $(N^{2/3})$	N N

$$
\begin{aligned}
\#\text{flops} &= O(\#\text{flops\_per\_iteration} \cdot \#\text{iterations}) \\
&= O(\#\text{flops}(\text{SPMV}) \cdot \#\text{iterations}) \\
&= O(\#\text{nnz}(A) \cdot \#\text{iterations}) \\
&= O(\#\text{nnz}(A) \cdot \text{cond}(A))
\end{aligned}
$$

for Poisson

works for any diagonally dominant matrix

Gauss-Seidel	I	N^2 $(N^{5/3})$	N N	N $(N^{2/3})$	N N

cost analysis like Jacobi, better constants

Works on any diagonally dominant matrix or s.p.d. matrix

Sparse Cholesky	D	$N^{3/2}$	$N \cdot \log N$	$N^{1/2}$	N
		(N^2)	$(N^{4/3})$	$(N^{2/3})$	$(N^{4/3})$

assumes we use best reordering of rows and columns (nested dissection)

$\Longrightarrow$ bottleneck is dense Cholesky on trailing dense submatrix of size $n \times n$ in 2D, $n^2 \times n^2$ in 3D

Conjugate Gradients (CG)	I	$N^{3/2}$	N	$N^{1/2} \log N$	N
		$(N^{4/3})$	N	$(N^{1/3} \log N)$	N

$$\# \text{flops} = O(\#\text{flops\_per\_iteration} \cdot \#\text{iterations})$$
$$= O(\#(\text{SPMV}) \cdot \sqrt{\text{cond}(A)})$$

works on any s.p.d. matrix

SOR	I	$N^{3/2}$	N	$N^{1/2}$	N
		$(N^{4/3})$	N	$(N^{1/3})$	N

works on any s.p.d. matrix

\# steps depends on cond(A)

analogous cost analysis to CG

SSOR/ Chebyshev	I	$N^{5/4}$	N	$N^{1/4}$	N
		$(N^{7/6})$	N	$(N^{1/6})$	N

$\# flops = O(\# Spmv \cdot (cond(A))^{1/4})$

need to know λ_{max} and λ_{min}

works for Poisson, works on any spd matrix

FFT	D	$N \cdot \log N$	N	$\log N$	N

works on Poisson

BCR	D	$N \cdot \log N$	N	?	?

slightly more general than FFT

Multigrid	I	N	N	$\log^2 N$	N

many variant beyond Poisson

used for FEM, Elliptic PDEs

Lower Bound		N	N	$\log N$	