Welcome to Ma221! Lecture 2, Aug 25

Last time: 4 axes of design space

- Math problem
- Structure of A
- Desired accuracy
- Target platform

Math Problem

- $Ax = b$
- Least Squares
- Eigen problems
  - One Matrix: $x'(t) = A x(t)$
  - Multiple Matrices
    $M \cdot x''(t) + K \cdot x(t) = 0$
  - Ex: $x(t)$: position
    - M: Mass
    - K: Stiffness
    - $x(t)$: currents in circuit
    - M: Inductances
    - K: reciprocals of capacitances
  - plug in $x(t) = e^{\lambda t} x(0)$
    $\lambda^2 M x(0) + K x(0) = 0$
    - $x(0)$ is a generalized evec
    - λ^2 is a " eval

usual def $\det(K-\lambda I)=0$ if λ eval of K

becomes $\det(K+\lambda' M)=0$, $\lambda'=\lambda^2$

All ideas for one matrix generalize:

Jordan Form → Weierstrass Form

Schur Form → Generalized Schur

Why not find evals of $M^{-1}K$?

What if M singular?

Arises in "differential algebraic eqns"

ODEs with linear constraints

Nonlinear Eigenproblems:

$$Mx''(t)+D\cdot x'(t)+Kx(t)=0$$

D: damping matrix if $x(t)$ position

resistances if $x(t)$ current

plugin $x(t)=e^{\lambda t}x(0)$, get

$$\lambda^2 Mx(0)+\lambda Dx(0)+Kx(0)=0$$

will reduce to linear problem of

2x size

Not all eigenproblem square:

Singular eigenproblems: Control Theory

$$x'(t)=Ax(t)+Bu(t)$$

$$A^{n\times n},\ B^{n\times m}\quad m<n$$

$u(t)$ is a "control input" for $x(t)$

What subspace can $x(t)$ lie in and be controlled by $u(t)$?

Rectangular eigenproblem:

$$\begin{array}{c c} & \begin{array}{cc} m+n & \end{array} \\ n & [B, A] \end{array}, \begin{array}{c c} & \begin{array}{cc} m & n \end{array} \\ n & [0, I] \end{array}$$

Jordon form $\rightarrow$ Kronecker form

Partial Solution:

find subset of evals and evecs

invariant subspace

low rank approximation

only the largest singular values

– can be much cheaper

Updating Solutions

Given solution for A, cheaply update if A changes "a little"

change a few entries

" " " rows and columns

add a few " " "

add a low rank matrix to A

(standard solution for SVD uses this as building block)

(Not) Tensors

instead of 2D arrays (matrices) could be 3D, 4D, . .

Lots of problems generalize

matrix multiply $\rightarrow$ tensor contractions

low rank approximations

"Most tensor problems are NP-complete"

Axis 2: Structure of A

Office hours story

Student: "I need to solve an nxn

system $Ax=b$. What should I do?"

Prof: "Standard alg is Gauss. Elim costs $\frac{2}{3}n^3$ flops"

S: Too expensive

P: Tell me more about A

S: A is symmetric, real

P: Anything else?

S: Yes A is positive definite $x^T A x > 0$ if $x \neq 0$

P: Great! You can use Cholesky, costs only $\frac{1}{3}n^3$

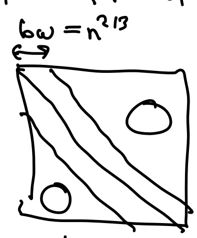
Prof. records conversation on board (Table 6.1 in text)

S: Still too expensive

P: Tell me more

S: A has lots of zeros, zero if farther than $n^{2/3}$ from diagonal.

P: Great! Band Matrix



version called band Cholesky only costs $O(bw^2 \cdot n) = O(n^{7/3})$

Much Cheaper!

S: Still too expensive

P: Tell me more

S: I need to solve $Ax=b$ for same A and many b.
So should I compute A^{-1} and multiply by it?

P: A^{-1} will be dense, so multiplying $A^{-1}\cdot b$ costs $2n^2$ flops.
but can reuse output of band Cholesky to solve for each b in $O(bw\cdot n) = O(n^{5/3})$ flops

S: Too expensive

P: Tell me more

S: Lot more zeros in A
just ≤ 7 nonzeros/row

P: Let's use an iterative alg. instead of a "direct" method:
Only need to multiply vector by A at each step, cost $O(n)$

S: How many matrix-vector multiplies do I need to do?

P: Can you tell me about range of evals of A? $\kappa(A) = \frac{\lambda_{max}}{\lambda_{min}}$ = "condition number"

S: $\kappa(A) = n^{2/3}$ too

P: You could use Conjugate Gradients (CG), needs $O(\sqrt{\kappa(A)})$ iterations

$\Rightarrow$ cost $O(\sqrt{\kappa(A)}) \cdot O(n) = O(n^{\frac{1}{3}} \cdot n) = O(n^{4/3})$

Happy yet?

S: No

P: Tell me more!

S: I know λ_{max}, λ_{min}, does that help?

P: You know alot about A.
What problem are you really trying to solve?

S: I have cube of metal, know temp at boundaries, want temp inside

P: Oh, you're solving 3D Poisson!
Best alg either

direct: FFT: Fast Fourier Trans
cost $O(n \log n)$ flops
iterative: Multigrid (MG)
cost $O(n)$, $O(1)$ per output
$\Rightarrow$ lower bound

S: Where do I download software?

Important: exploit Math Structure

Poisson one of best studied systems $Ax=b$, all algorithms more general

3) Desired accuracy:

1) "Guaranteed correct"
eg for $Ax=b$, need proof A is nonsingular, use Mathematica

2) "Backward Stability"
"get exact answer for slightly wrong problem"

Want $y = f(x)$

Get $alg(x) = f(x + \delta)$

where δ "small" compared to x

Det of "small" requires matrix norms

Size of δ should be proportional to error in arithmetic

"machine epsilon"

eg 10^{-16} in double precision

3) Residual as small as desired, for problems too big for a direct, backword stable method.

Iterate until "residual" as small as desired

Ex $\|Ax - b\|_2$ for $Ax = b$

equals true backword error

Lots of Algs: Chap 6 + 7

4) "Probably OK": randomized algs: replace a large problem with a small approximation by

"random projection" or
"random sampling",
then solve small problem with
deterministic alg:
Some iterative, control error
Some not!

Thm:

"The error is less than ε
with probability $1-\delta$ if
size of random approx big
enough, i.e size proportional
to $f(\delta,\varepsilon)$, gets larger as
δ, ε get smaller

Often $f(\delta,\varepsilon) = \Omega\left(\log\left(\frac{1}{\delta}\right)\cdot\frac{1}{\varepsilon^2}\right)$

$\Rightarrow$ if ε too small, use
deterministic or iterative
alg