Fast Algorithms and Integral Equation Methods

CS598APK

Andreas Kloeckner

Fall 2024

Outline

Outline

Outline

Numerics is lies.

<u>complexity</u>

$O(n^2)$ OK?

<u>conditioning</u>

condition # : κ

rel. error on output $\leq \kappa$ rel. error on input

$$\kappa(A) = \|A\| \, \|A^{-1}\|$$

$$D \approx \partial_x$$

$$D\hat{u} = f$$

really poorly conditioned

What's the point of this class?

- Starting point: Large-scale scientific computing
- Many popular numerical algorithms: $O(n^\alpha)$ for $\alpha > 1$
  (Think Matvec, Matmat, Gaussian Elimination, LU, ...)
- Build a set of tools that lets you cheat: Keep α small
  (Generally: probably not–Special purpose: possible!)
- Final goal: Extend this technology to yield PDE solvers
- But: Technology applies in many other situations
  - Many-body simulation
  - Stochastic Modeling
  - Image Processing
  - 'Data Science' (e.g. Graph Problems)
- This is class is about an even mix of math and computation

$A\vec{x} = \vec{b}$ → matrix-vector
↪ solve

$K\vec{\sigma} = \vec{f}$

$\sum K_{ij}\sigma_j = f_i$ $\quad i \leftrightarrow x$

$(x \in \Omega')$ $\int K(x,y)\,\sigma(y) = f(x)$ $\quad j \leftrightarrow y$

"columns" → Ω, Γ

↪ "integral operator"

Example:
$k(\vec{x},\vec{y}) = \frac{1}{\|x-y\|_2}$

① Efficiency
dense → $O(n^2)$

② Solvable?

③ Applications?
↪ PDE-slated.

① efficiency cont'd

$\bar{K}(\vec{r}) = \frac{1}{\|r\|_2}$

$$\int k(x,y)\sigma(y)\,dy = \int \bar{K}(x-y)\,\sigma(y)\,dy$$

convolution

$$= \bar{K} * \sigma$$

$$\mathcal{F}\{K * \sigma\} = \mathcal{F}(K) \cdot \mathcal{F}(\sigma)$$

Fourier transform : often $O(n \log n)$

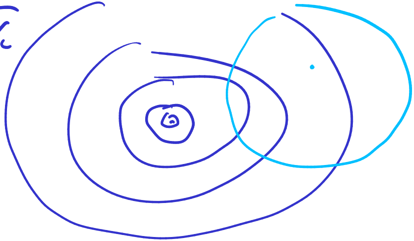
periodicity

"Fast" algorithm

regular grid

A smooth function is one that's well-approximated by a Taylor series.

$\bar{r}_c$



→ expand away from singularity/ diagonal

How about the solve?

iterative method for $A\vec{x} = \vec{b}$ (CG, GMRES)

- iterate to improve a guess until 'close enough' to solution
- each iteration: a matvec → Hope: $O(1)$
- Cost: # iterations · matvec cost $O(n)$

(2) <u>Solvable?</u> $\int_\Omega K(x,y)\,\sigma(y)\,dy = f(x) \quad (x\in\Omega)$

If K is a smooth function, is a compact operator.

$A : \sigma \mapsto f$ "operator"

$\sigma : \Omega \to \mathbb{R}$

$$(A\sigma)(x) = \int_\Omega K(x,y)\,\sigma(y)\,dy = f(x)$$

"compact" operator: well-approximated by a finite-dim. operator.

$$\int K(x,y)\,\sigma(y)\,dy = f(x)$$ "first kind"

$$\sigma(x) + \int_\Omega K(x,y)\,\sigma(y)\,dy = f(x)$$ "second kind"

$$(I + A)\sigma = f$$

will develop a solution theory.

3 Applications

$$\Delta u = f$$ "Poisson's equation"

solution $\Leftarrow$ how represented?
↳ polynomial
↳ point values

$$\Delta G = \delta$$ "δ-~~function~~ distribution":

"weakly": $\int \Delta G \varphi = \int \delta \varphi$

$\delta(x) = 0 \quad x \neq 0$

$\int_{\mathbb{R}^n} \delta(x)\, dx = 1$

"Free-space Green's function"

in 3D: $G(\vec{x}) = \dfrac{1}{\|x\|_2}$

$\Delta G = 0 \qquad x \neq 0 \qquad \Big| \qquad x = 0 : \quad ?$

representation of the soln.

$$u(\vec{x}) = \sum_j G(x - y_j)\,\sigma_j$$

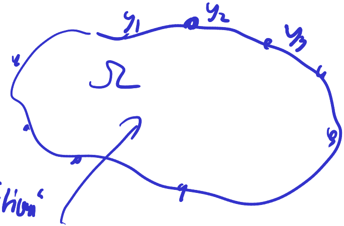
"Laplace"

BVP:

$$\boxed{\begin{array}{ll} \Delta u = 0 & \text{on } \Omega \\ u = g & \text{on } \partial\Omega \end{array}}$$

"PDE"

"boundary condition"



$\Delta u = 0$

$$\vec{\nabla} = \begin{pmatrix} \partial_x \\ \vdots \\ \partial_z \end{pmatrix}$$

$$\Delta = \partial_x^2 + \partial_y^2 + \partial_z^2 = \nabla \cdot \nabla$$

$$\underbrace{u(\vec{x}) = \sum {}' G(\vec{x} - \vec{y}_j)\ \sigma_j}_{\text{repn}} = g(\vec{x}) \qquad (\vec{x} \in \partial\Omega)$$

$u(x) = g(x)$ on $\partial\Omega$

"Upgrade" the representation:

From $u(\vec{x}) = \sum {}' G(x - y_j)\ \sigma_j$

do:

$$u(\vec{x}) = \int_{\partial\Omega} {}' G(x - y)\ \sigma(y)\, dy$$

"layer potential"

Survey

no class on Thu

- Home dept
- Degree pursued
- Longest program ever written
  - in Python?
- Research area
- Interest in PDE solvers

Class web page

https://bit.ly/fastalg-s24

contains:

- Class outline
- Notes
- Demos
- Assignments
- Discussion forum
- Grading
- Video