Fast Algorithms

and
Integral Equation Methods

CS 598 APK - Fall 2017

Andreas Kloeckner

Today:

- Syllabus
- Advertisement
- Matrices → Integral operator

What's the point of this class?

- Starting point: Large-scale scientific computing
- Many popular numerical algorithms: $O(n^{\alpha})$ for $\alpha > 1$ $O(n^3)$

  (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 $\leftarrow$
- 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

Survey

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

Class web page

`bit.ly/fastalg-f17`

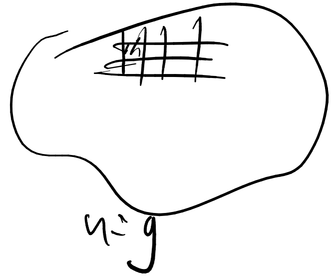
contains:

- Class outline
- Assignments
- Virtual Machine Image
- Piazza
- Grading
- Video
- HW1

$\Delta u = 0$

$\partial_x^2 u + \partial_y^2 u = 0$

Why study this at all?



- Finite difference/element methods are inherently
  - ill-conditioned
  - tricky to get high accuracy with
- Build up a toolset that does *not* have these flaws
- Plus: An interesting/different analytical and computational point of view
  - If you're not going to use it to solve PDEs, it (or the ideas behind it) will still help you gain insight.

FD/FEM: Issues

Idea of these methods:

1. Take differential equations
2. Discretize derivatives
3. Make linear system
4. Solve

So what's wrong with doing that?

$Ax = b$

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

rel. error $x \le \kappa(A) \cdot$ rel. error(b)

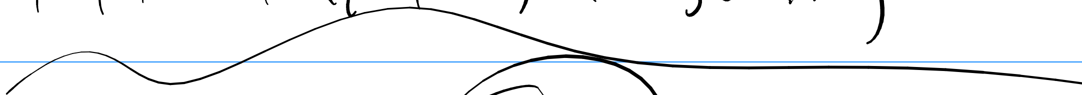
fundamental sol.

$$G(x) = \frac{1}{\|x\|_2}$$

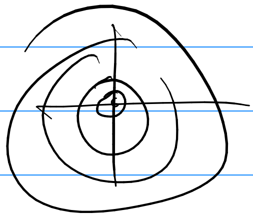
$\Delta u = 0$

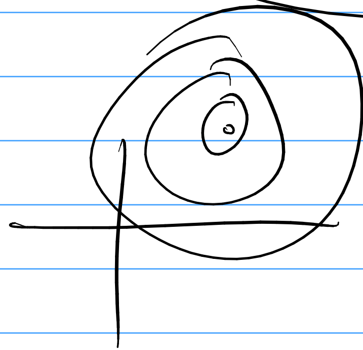
ΔG

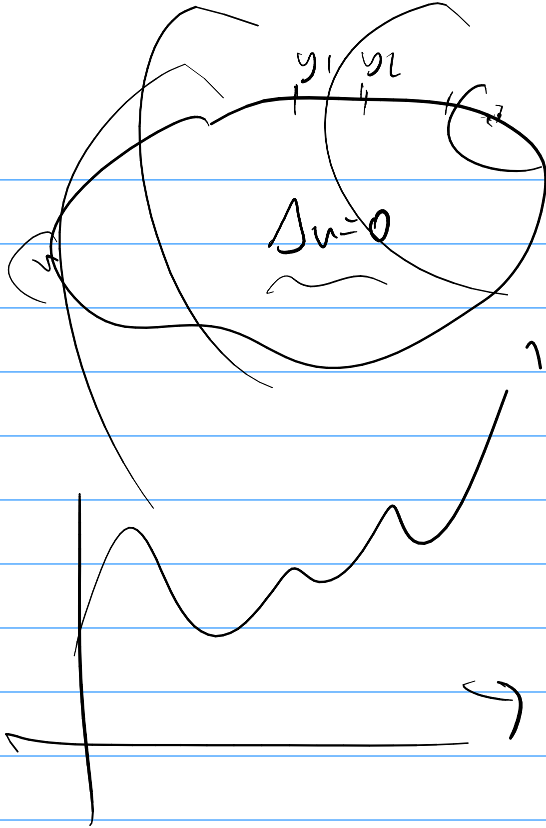
$$\alpha_1 G(\;) + \alpha_2 G(\ldots) + \alpha_3 G(\ldots)$$



$G(x-y)$







$u \approx g$

$$g(x) = u(x) = \cancel{\sum_{i=1}^{n}} G(x - y_i)\, \sigma_i$$

$$g(x) \overset{!}{=} u(x) = \int_\Gamma G(x-y)\, \sigma(y)\, dy$$

$$= \int_\Gamma \frac{1}{\|x-y\|}\, \sigma(y)\, dy$$

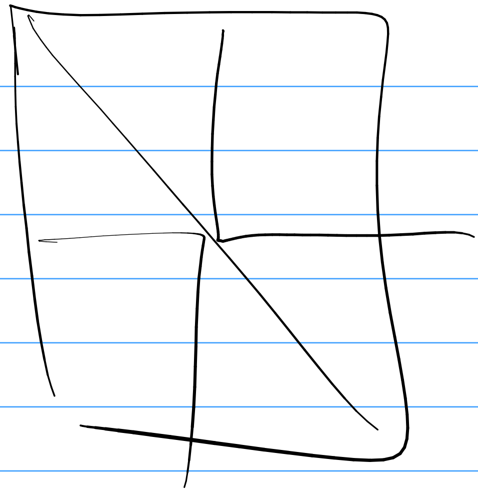
$$\to G(x) = \frac{1}{\|x\|_2}$$

$$E(x) = -\nabla u(x)$$

Problems:

- Integral defined?
- Quadrature
- dense matrices
- solvable

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



A

$$A = \vec{u}\,\vec{v}^T$$

$$A\vec{x} = \vec{u}\left(\vec{v}^T\vec{x}\right)$$

Ewald summation

PME

Barnes Hut

(Tree code)

FMM

Direct solver

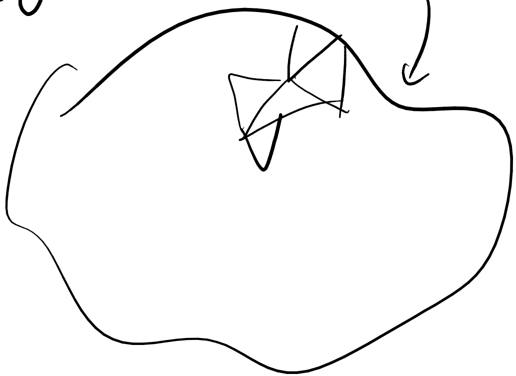
<u>IE methods: A brief outline</u>

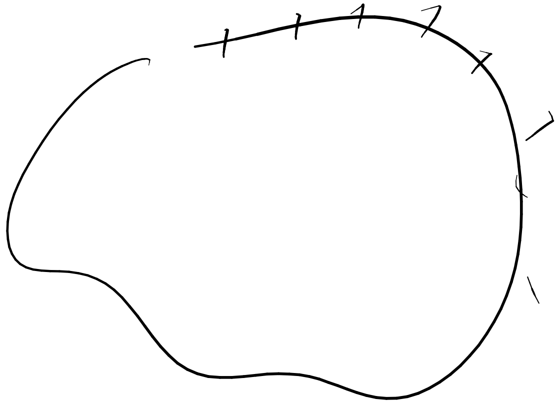
Bonus Advertising Goodie

> Both multigrid and fast/IE schemes ultimately are $O(N)$ in the number of degrees of freedom N.

$\Delta u = 0$

$u = g$





$O(\underline{n}) \Leftarrow n \sim \left(\frac{1}{h}\right)^d$

$\int G(x-y)\,\sigma(y)\,dy$

$O(n)$

$n \sim \left(\frac{1}{h}\right)^{d-1}$

1 Dense Matrices and Computation

Matvec: A slow algorithm

Matrix-vector multiplication: our first ‘slow’ algorithm.

$O(N^2)$ complexity.

$$\beta_i = \sum_{j=1}^{N} A_{ij}\alpha_i$$

Assume A dense.

Matrices and Point Interactions

$$A_{ij} = G(x_i, y_j)$$

Does that actually change anything?

Matrices and Point Interactions

$$A_{ij} = G(x_i, y_j)$$

Graphically, too:

Matrices and point interactions

$$\psi(x_i) = \sum_{j=1}^{N} G(x_i, y_j)\varphi(y_j)$$

This *feels* different.

Point interaction matrices: Examples

What kind of matrices, then?

Integral 'Operators'

Why did we go through the trouble of rephrasing matvecs as

$$\psi(x_i) = \sum_{j=1}^{N} G(x_i, y_j)\varphi(y_j)?$$

Cheaper Matvecs

$$\psi(x_i) = \sum_{j=1}^{N} G(x_i, y_j)\varphi(y_j)$$

So what can we do to make evaluating this cheaper?