

Fast Algorithms and Integral Equation Methods

CS598APK

Andreas Kloeckner

Fall 2026

You've been lied to

$O(n^2)$ is fine?

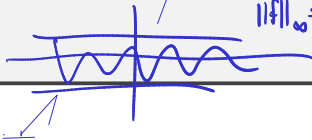
Conditioning

rel. error in output $\leq \kappa \cdot$ rel. error in input

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

$$\partial_x: C^1 \rightarrow C^0$$
$$\|\cdot\|_\infty$$

$$\|\partial_x\| = \sup_{\|f\|_\infty=1} \|\partial_x f\|$$



Derivatives are unbounded.

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

Integral Operators

$$\underline{A \vec{x} = \vec{b}} \rightarrow K \vec{\sigma} = \vec{f}$$

$$\sum_j k_{ij} \sigma_j = f_i$$

$$\rightarrow \int_{\mathcal{R}'} K(x, y) \sigma(y) dy = f(x)$$

$$\begin{aligned} i &\leftrightarrow x \\ j &\leftrightarrow y \end{aligned}$$

$$\sum K(\vec{x}, y_j) \sigma_j$$

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

Efficiency
 $O(n^2)$

Solvability
 $\rightarrow$ Theory

Applications
 $\rightarrow$ PDE solvers

Solvability

 $\sigma(y)$ $\sigma: \Omega' \rightarrow \mathbb{C}$

$$(A\sigma)(x) = \int_{\Omega'} k(x,y) \sigma(y) dy \quad (x \in \Omega)$$

If k is smooth, A is a compact operator.

A smooth function is
one that has a rapidly
convergent Taylor expansion

↓
well-approx. by an operator
with finite-dim. range

Solvability

$$A \sigma = f \quad \leftarrow \text{no known}$$

compact

integral equation of the first kind

$$(I + A) \sigma = f \quad \leftarrow \text{second-kind equation}$$

Fredholm alternative

Efficiency

- low-rank approximation
 - smooth functions $\rightarrow$ short expansion

$\rightarrow$ low rank

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

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

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

$$\int k(x-y) \sigma(y) dy \leftarrow \text{convolution}$$

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}$$

convolution structure

Applications: (Elliptic) PDE solving (I)



Applications: (Elliptic) PDE solving (II)



And one more thing...

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



Survey

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

Class web page

<https://tinyurl.com/fastalg-f26>

contains:

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

Open Source <3

These notes (and the accompanying demos) are open-source!

Bug reports and pull requests welcome:

<https://github.com/inducer/fast-alg-ie-notes>

Copyright (C) 2013 – 26 Andreas Kloeckner

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Sources

- ▶ [Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions](#) by Halko/Martinsson/Tropp
- ▶ Carrier, Greengard, Rokhlin: [A Fast Adaptive Multipole Algorithm for Particle Simulations](#)
- ▶ Rainer Kress: [Linear integral equations](#). (second edition)
- ▶ David Colton and Rainer Kress: [Inverse Acoustic and Electromagnetic Scattering Theory](#). (3rd edition)