

Today:

- $\underbrace{\text{matrix}}_{\uparrow \text{integral operators}} \rightarrow O(h^2)$

Matvec: A Slow Algorithm

Matrix-vector multiplication: our first 'slow' algorithm.
 $O(N^2)$ complexity.

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

Assume A dense.

Matrices and Point Interactions

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

$\swarrow \in \mathbb{R}^3$
 $\nwarrow \in \mathbb{R}^3$



Does that actually change anything?

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

$x_i \rightarrow$ Targets / Observation points

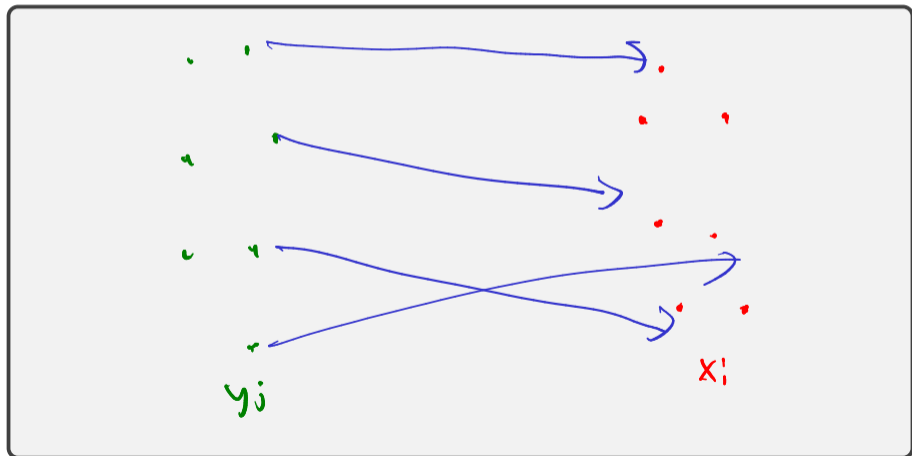
$y_j \rightarrow$ Sources

$G(x, y) \rightarrow$ kernel

Matrices and Point Interactions

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

Graphically, too:



Matrices and point Interactions

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

infininitely tall matrix

This *feels* different.

$$\varphi(x) = \int_{\Omega} G(x, y) \varphi(y)$$

Q: Are there enough matrices that come from globally defined G to make this worth studying?

Point Interaction Matrices: Examples (I)

Interpolation: $\tilde{\varphi}(x) = \sum_{j=1}^N \underbrace{l_j(x)}_{\text{Lagrange basis}}, \varphi(y_j)$

Interpolation error: $\varphi(x) - \tilde{\varphi}(x) = \varphi(x) - \sum_{j=1}^N \underbrace{l_j(x)}_{\text{Lagrange basis}}, \varphi(y_j)$

Num diff: $\tilde{\varphi}'(x) = \sum_{j=1}^N \underbrace{l_j'(x)}_{\text{Lagrange basis derivative}}, \varphi(y_j)$

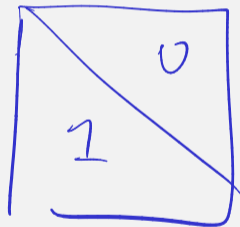
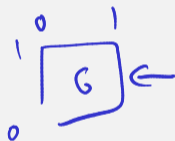
Identifying:

$$\varphi(x) = \int_{\mathbb{R}} \underbrace{\delta(x-y)} \varphi(y) dy$$

$$z_i = \sum A_{ij} z_j \quad (\Delta)$$

Point Interaction Matrices: Examples (II)

$$\chi_{\infty} \in [0,1] = \int_0^x f(y) \lambda_{xy} = \int_0^1 \underbrace{1}_{\uparrow} f(y) \lambda_{xy}$$



Point Interaction Matrices: Examples (III)

Convolutions

$$\hookrightarrow \varphi(x) = \int G(x-y) \psi(y) dy$$

\hookrightarrow circulant matrices

$$G(x,y) = C \cdot \frac{1}{|x-y|} \leftarrow \begin{matrix} \Delta u = 0 \\ (s_0) \end{matrix} \text{ fund. solution}$$

$$C \cdot \log(|x-y|) \leftarrow \mathbb{R}^2$$

So yes, there are indeed lots of these things.

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?

- sparse
- circulant / Toeplitz / convolutions
- low rank

Fast Dense Matvecs

Consider

$$A_{ij} = u_i v_j,$$

let $\mathbf{u} = (u_i)$ and $\mathbf{v} = (v_j)$.

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

Can we compute $A\mathbf{x}$ quickly? (for a vector \mathbf{x})

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

Fast Dense Matvecs (II)

$$A = \mathbf{u}_1 \mathbf{v}_1^T + \cdots + \mathbf{u}_K \mathbf{v}_K^T$$

$$A \in \mathbb{R}^{N \times N}$$

Does this generalize? What is K here?

$$\text{Cost: } O(NK)$$

Low-Rank Point Interaction Matrices

Usable with low-rank complexity reduction?

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

$$A = uv^T$$

$$G(x, y) = u(x) \cdot v(y)$$

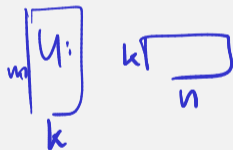
↳ simple "compact operator"

"compact" \Leftrightarrow "approximately finite-dim. range"

Numerical Rank

What would a *numerical* generalization of 'rank' look like?

A has rank k if $A \in \mathbb{R}^{m \times n}$

$$A = UV$$


The diagram illustrates the dimensions of the matrices in the equation $A = UV$. Matrix U is shown as a vertical rectangle with height m and width k . Matrix V is shown as a horizontal rectangle with height k and width n . The product UV results in a matrix of size $m \times n$.

$$\text{num rank}(A, \epsilon) = \min \{k : \exists U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{k \times n} \\ |A - UV|_2 \leq \epsilon\}$$

Eckart-Young-Mirsky Theorem

$$\|A\|_2 = \max |\sigma_i| \quad \|A\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$$

Theorem (Eckart-Young-Mirsky)

SVD $A = U\Sigma V^T$. If $k < r = \text{rank}(A)$ and

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T,$$

then

$$\min_{\text{rank}(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$$

\mathbb{F} $\mathbb{F} = \sum_{i=k+1}^{\min(m,n)} \sigma_i^2$

$$\|A\|_F = \sqrt{\sum \sigma_i^2}$$

Q: What's that error in the Frobenius norm?

So in principle that's good news:

- ▶ We can find the numerical rank.
- ▶ We can also find a factorization that reveals that rank (!)

Demo: Rank of a Potential Evaluation Matrix (Attempt 2)

Constructing a tool

There is still a slight downside, though.

build: $O(N^2)$

factorize: $O(N^3)$

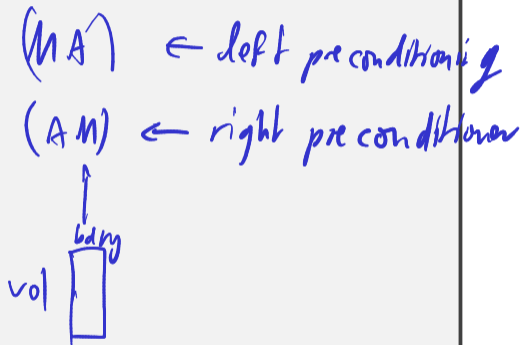
matvec: $O(Nk)$

Representation

$$Ax=b$$

$$M \approx A^{-1}$$

What does all this have to do with (right-)preconditioning?



Representation (in context)



Outline

Introduction

Dense Matrices and Computation

Tools for Low-Rank Linear Algebra

Low-Rank Approximation: Basics

Low-Rank Approximation: Error Control

Reducing Complexity

Rank and Smoothness

Near and Far: Separating out High-Rank Interactions

Outlook: Building a Fast PDE Solver

Going Infinite: Integral Operators and Functional Analysis

Singular Integrals and Potential Theory

Boundary Value Problems

Back from Infinity: Discretization

Computing Integrals: Approaches to Quadrature

Going General: More PDEs

Rephrasing Low-Rank Approximations

SVD answers low-rank-approximation ('LRA') question. But: too expensive. First, rephrase the LRA problem:

$$\therefore A = UV^T \leftarrow \text{factorization form}$$

$$A \approx \underbrace{Q}_{\substack{\approx \\ \text{rank } k}} \underbrace{Q^T A}_{\leftarrow} \leftarrow \text{projection}$$

Using LRA bases

$$A \approx QQ^T A$$

If we have an LRA basis Q , can we compute an SVD?

Complexity

1	$B = Q^T A$	$\square \square$	$N^2 k$
2	$B = \tilde{U} \Sigma V^T$	$\square = O D \square$	$k^2 N$
3	$A = QB = \underbrace{Q \tilde{U}}_U \Sigma V^T$		$k^2 N$

$$\begin{matrix} k \\ \boxed{} \\ N \end{matrix}$$

Finding an LRA basis

How would we *find* an LRA basis?

Goal: Find Q columns

$$\|A - QQ^T A\|_2 \leq \epsilon$$

- Fixed-rank approximation
- Adaptive CRA

Idea 1: SVD

Idea 2: Randomized algorithm for range finding

Giving up optimality

What problem should we actually solve then?

$$\|A - QQ^T A\|_2 = \min_{\text{rank}(X) \leq k} \|A - X\|_2 = \sigma_{k+1}$$

$$\|A - QQ^T A\|_2 = \min_{\text{rank}(X) \leq k+p} \|A - X\|_2 = \cancel{\sigma_{k+1}}^{\epsilon}$$

Recap: The Power Method

How did the power method work again?

