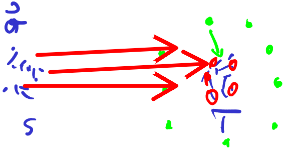
Announcements:

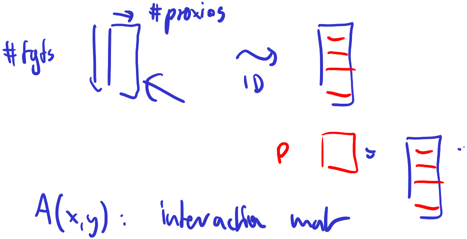
Goals:

Review:

"basis": expansion → targets

"coefficients": sources → expansion





A(x,y): interaction mat

$$P\,A(\text{skel},\text{src}) \approx A(\text{tgt},\text{src})$$

$$\rightarrow \boxed{P\,A(\text{skel},\text{src})}\,\vec{\sigma} \approx A(\text{tgt},\text{src})\,\vec{\sigma}$$

Proxies that lead to bad results:

- too close? increases the rank of A(skel,src)

- not covering the sphere? → scary 'cliff hanger
- too far? → Ok in exact arithmetic
  → "targets too similar" in FP,
- too few? → limits representable i,rank
- too many? → costly

Where are we now? (I)

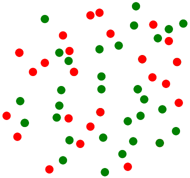
Summarize what we know about interaction ranks.

- We know that far interactions with a smooth kernel have low rank. (Because: short Taylor expansion suffices)
- If
  $$\psi(\boldsymbol{x}) = \sum_j G(\boldsymbol{x}, \boldsymbol{y}_j)\varphi(\boldsymbol{y}_j)$$
  satisfies a PDE (e.g. Laplace), i.e. if $G(\boldsymbol{x}, \boldsymbol{y}_j)$ satisfies a PDE, then that low rank is *even* lower.
- Can construct interior ('local') and exterior ('multipole') expansions (using Taylor or other tools).
- Can lower the number of terms using the PDE.
- Can construct LinAlg-workalikes for interior ('local') and exterior ('multipole') expansions.
- Can make those cheap using proxy points.

Where are we now? (II)

So we can compute interactions where sources are distant from targets (i.e. where the interaction is low rank) quite quickly.

Problem: In general, that's not the situation that we're in.



But: *Most* of the targets are far away from *most* of the sources.

(⇔ Only a few sources are close to a chosen 'close-knit' group of targets.)

So maybe we can do business yet–we just need to split out the near interactions to get a hold of the far ones (which (a) constitute the bulk of the work and (b) can be made cheap as we saw.)

Outline

Introduction

Dense Matrices and Computation

Tools for Low-Rank Linear Algebra

Rank and Smoothness

Near and Far: Separating out High-Rank Interactions
- Ewald Summation
- Barnes-Hut
- Fast Mutipole
- Direct Solvers
- The Butterfly Factorization

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

Outline

Introduction

Dense Matrices and Computation

Tools for Low-Rank Linear Algebra

Rank and Smoothness

Near and Far: Separating out High-Rank Interactions
- Ewald Summation
- Barnes-Hut
- Fast Mutipole
- Direct Solvers
- The Butterfly Factorization

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

Preliminaries: Convolution

$$(f * g)(x) = \int_{\mathbb{R}} \overset{\overset{\text{"}K\text{"}}{f(x-\xi)}\;\;\overset{\text{"}\sigma\text{"}}{g(\xi)}}{\cancel{f(\xi)g(x-\xi)}}\,d\xi.$$

- Convolution with shifted δ is the same as shifting the function;

$$[f * \underbrace{(\xi \mapsto \delta(\xi - a))}_{}](x) = f(x-a)$$

- Convolution is linear (in both arguments) and commutative.

$$\int_0^1 f(x)\,\delta(x-0.5)\,dx = f(0,5)$$

Preliminaries: Fourier Transform

$$\mathcal{F}(f)(\omega) = \int_{\mathbb{R}} f(x) e^{-2\pi i \omega x} dx$$

- Convolution turns into multiplication: $\mathcal{F}\{f * g\} = \mathcal{F}f \cdot \mathcal{F}g$,
- A single δ turns into: $\mathcal{F}\{\delta(x-a)\}(\omega) = e^{-ia\omega}$
- And a "train" of δs turns into:

$$\mathcal{F}\left\{\sum_{\ell \in \mathbb{Z}} \delta(x-\ell)\right\}(\omega) = \sum_{k \in \mathbb{Z}} \delta(\omega - 2\pi k).$$

What is $\mathcal{F}\{f(x-a)\}$? $x \mapsto f(x-a) = (\delta(x-a)) * f$

$$\boxed{\mathcal{F}\{f(x-a)\} = \mathcal{F}\{\delta(x-a) * f\} = \mathcal{F}\{\delta(x-a)\} \cdot \mathcal{F}\{f\} = \mathcal{F}\{f\} e^{-ia\omega}}$$

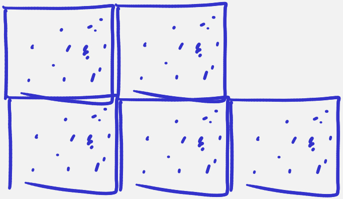
See e.g. [Décoret '04].

Simple and Periodic: Ewald Summation

Want to evaluate potential from an infinite periodic grid of sources:

$$\psi(\boldsymbol{x}) = \sum_{\boldsymbol{m}\in\mathbb{Z}^d} \sum_{j=1}^{N_{\text{src}}} G(\boldsymbol{x}, \boldsymbol{y}_j + \boldsymbol{m})\varphi(\boldsymbol{y}_j)$$

$\mathbb{Z} = \{ \ldots, -2, -1, 0, 1, 2, \ldots \}$



Lattice Sums: Convergence

Q: When does this have a right to converge?

$$\boxed{\sum_n \frac{1}{n}}$$ no

$$\boxed{\sum_n \frac{1}{n^2}}$$ ok

Assume $G = O(\|\vec{x}\|_2^{-p})$ (as $\vec{x} \to \infty$)

$$\psi(\vec{0}) = \sum_{i=0}^{\infty} \sum_{\text{cells @ dist} \in [i, i+1)} O(i^{-p})$$

$$= \sum_{i=0}^{\infty} O(i^{d-1}) \, O(i^{-p}) = \sum_{i=0}^{\infty} O(i^{d-1-p})$$

$$d-1-p < -1 \iff p > d$$

Ewald Summation: Dealing with Smoothness

$G(x-(y_j+i))\ \varphi(y_j)$

$$\psi(\boldsymbol{x}) = \sum_{\boldsymbol{i}\in\mathbb{Z}^d}\sum_{j=1}^{N_{\text{src}}} G(\boldsymbol{x},\boldsymbol{y}_j+\boldsymbol{i})\underline{\varphi(\boldsymbol{y}_j)}$$

Clear: a discrete convolution. Would like to make use of the fact that the Fourier transform turns convolutions into products. How?

Issue: G is non-smooth.

Ewald Summation: Screens

σ density

screen

$$G(\vec{x}) = \underbrace{\sigma(x)\, G(x)}_{G_{LR}} + \underbrace{(1-\sigma(x))\, G(x)}_{G_{SR}}$$

For example,

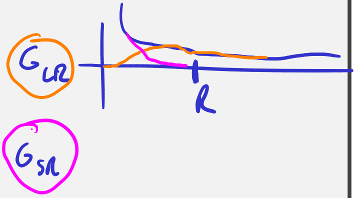
$$\frac{1}{\|x\|_2^4} = G(\vec{x}) = \sigma(\vec{x})\frac{1}{\|\vec{x}\|_2^4} + (1-\sigma(\vec{x}))\frac{1}{\|\vec{x}\|_2^4}$$

$\sigma(x) \in [0,1]$

$\sigma(\vec{x}) = O(\|\vec{x}\|_2^4) \quad (\vec{x} \to 0)$

$(1-\sigma)$ has bounded support.

$\sigma(\vec{x}) = 1 \quad$ if $\quad \|x\|_2 \geq R$



Ewald Summation: Field Splitting

We can split the computation (from the perspective of a unit cell target) as follows:

	G_{SR}	G_{LR} (smooth)
Close source	(nonsingular ⇒ eval. directly	Fourier
Far away ("dist $\ge R$")	0	Fourier

Ewald Summation: Summation (1D for simplicity)

Interesting bit: How to sum G_{LR}.

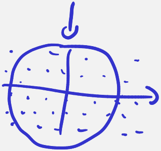
$$\mathcal{F}\{\psi\} - \mathcal{F}\{\psi_{SR}\} = \mathcal{F}\{\psi_{LR}\}$$

$$= \mathcal{F}\{G_{LR}\} \quad \mathcal{F}\Big\{x \mapsto \sum_{m\in\mathbb{Z}} \sum_{j=1}^{N_{src}} \delta(x - y_j - m)\Big\}$$

$$= \mathcal{F}\{G_{LR}\} \cdot \sum_{j=1}^{N_{src}} e^{-iy_j\omega} \cdot \mathcal{F}\Big\{\sum_{m\in\mathbb{Z}} \delta(x - m)\Big\}$$

$j=1$

$$= = \mathcal{F}\{G_{LR}\}(\omega) \cdot \sum_{j=1}^{N_{src}} e^{-iy_j\omega} \cdot \Big(\omega \mapsto \sum_{n\in\mathbb{Z}} \delta(\omega - 2\pi n)\Big)$$



Ewald Summation: Remarks

In practice: Fourier transforms carried out discretely, using FFT.

- Additional error contributions from interpolation (small if screen smooth enough to be well-sampled by mesh)
- $O(N \log N)$ cost (from FFT)
- Need to choose evaluation grid (‘mesh’)
- Resulting method called Particle-Mesh-Ewald (‘PME’)

Outline

Introduction

Dense Matrices and Computation

Tools for Low-Rank Linear Algebra

Rank and Smoothness

Near and Far: Separating out High-Rank Interactions
- Ewald Summation
- Barnes-Hut
- Fast Mutipole
- Direct Solvers
- The Butterfly Factorization

Outlook: Building a Fast PDE Solver

Going Infinite: Integral Operators and Functional Analysis

Singular Integrals and Potential Theory

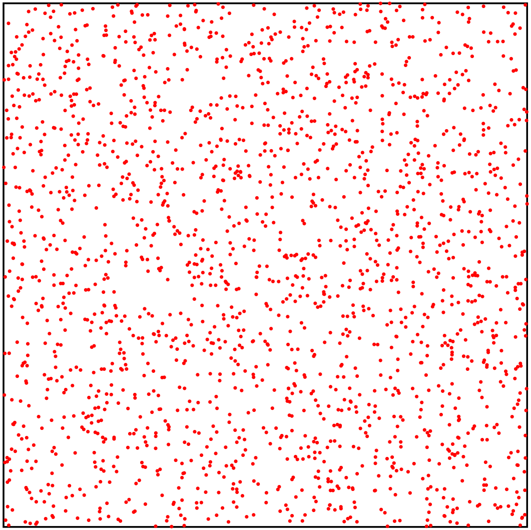
Boundary Value Problems

Back from Infinity: Discretization

Computing Integrals: Approaches to Quadrature

Going General: More PDEs

Barnes-Hut: Putting Multipole Expansions to Work



(Figure following G. Martinsson)

Barnes-Hut: The Task At Hand

Want: All-pairs interaction.

Caution:

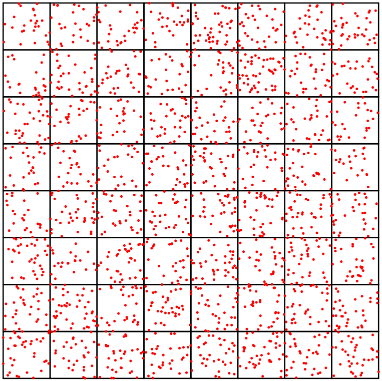
- In these figures: targets sources
- Here: targets and sources

$\vec{x}_i = \vec{y}_j$

$$\vec{u} = A\vec{q}$$

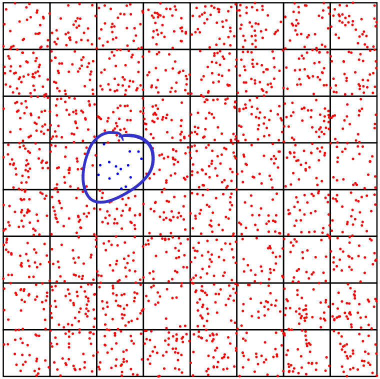
$$A_{ij} = \log(\|x_i - y_j\|_2)$$

Barnes-Hut: Putting Multipole Expansions to Work



(Figure following G. Martinsson)

Barnes-Hut: Putting Multipole Expansions to Work



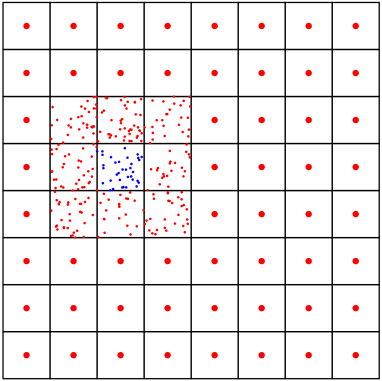
(Figure following G. Martinsson)

Barnes-Hut: Box Targets

For sake of discussion, choose one ‘box’ as targets.

Q: For which boxes can we then use multipole expansions?

Barnes-Hut: Putting Multipole Expansions to Work



(Figure following G. Martinsson)

complexity?

Barnes-Hut: Accuracy

With this computational outline, what's the accuracy?

Q: Does this get better or worse as dimension increases?