Outline:

- PDE solvers based on integral equations.
  - The basic idea.
  - When should they be used?
  - Current state of research.

- Fast Multipole Methods.
  - The basic idea.
  - What are they good for?
  - Current state of research.

- A matrix factorization and its consequences.

Boundary methods for solving PDEs

Suppose that we wish to find a solution of the boundary value problem

$$\text{(BVP)} \qquad \begin{cases} -\Delta v(x) = 0, & x \in \Omega, \\ v(x) = f(x), & x \in \Gamma. \end{cases}$$

We make the Ansatz

$$v(x) = \int_\Gamma \big(n(y) \cdot \nabla_y \log|x-y|\big) u(y)\, ds(y), \qquad x \in \Omega,$$

where $n(y)$ is the outward pointing unit normal of Γ at y. Then v solves (BVP) if and only if u solves the boundary integral equation

$$\text{(BIE)} \qquad \frac{1}{2}u(x) + \int_\Gamma \big(n(y) \cdot \nabla_y \log|x-y|\big) u(y)\, ds(y) = f(x), \qquad x \in \Gamma.$$

- (BIE) and (BVP) are in a strong sense equivalent.
- (BIE) is appealing mathematically (2^{nd} kind Fredholm equation).

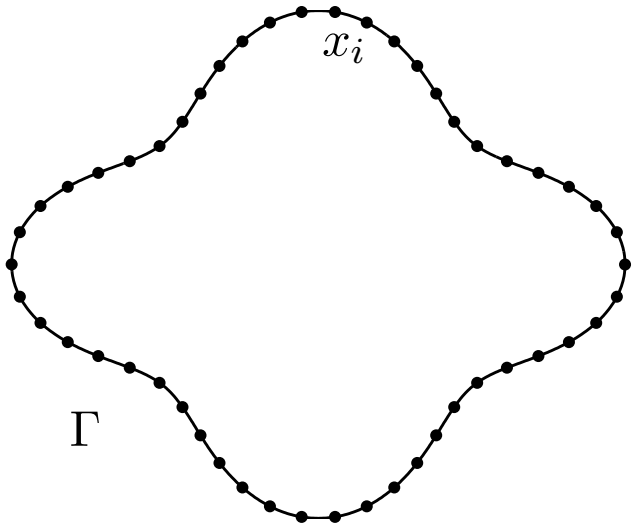
Numerical methods for BIEs

Suppose that we wish to numerically solve the integral equation

$$u(x) + \int_\Gamma K(x,y)u(y)\,ds(y) = f(x), \qquad x \in \Gamma.$$

We first discretize the contour into n points

$$\Gamma \sim [x_1, \ldots, x_n].$$



Then the operator

$$\int_\Gamma K(x,y)u(y)\,ds(y)$$

turns into a matrix A with entries (sort of)

$$A_{ij} = K(x_i, x_j), \qquad i,j = 1, \ldots, n.$$

Since A is **dense**, it appears that

the cost for constructing A is $O(n^2)$ (with a large constant),

the cost for solving $(I + A)u = f$ is $O(n^3)$.

Fast solution of boundary integral equations

We let A denote the dense $n \times n$ matrix discretizing the operator

$$\int_\Gamma K(x,y)u(y)\,ds(y).$$

There exist $O(n \log^q n)$ algorithms ($q = 0, 1, 2$) that evaluate the map

$$u \mapsto Au.$$

These include the Fast Multipole Method, Panel Clustering, multigrid, wavelets,...
Developed circa 1980 – 1985.

Using iterative methods (GMRES seems popular), the equation

$$(I + A)u = f$$

can then be solved using $O(\sqrt{\kappa} \cdot n \log^q n)$ operations, where κ is the condition number of $I + A$.

BIE formulations exist for many classical BVPs

Laplace $\quad -\Delta u = f,$

Elasticity $\quad \frac{1}{2} E_{ijkl} \left(\frac{\partial^2 u_k}{\partial x_l \partial x_j} + \frac{\partial^2 u_l}{\partial x_k \partial x_j} \right) = f_i,$

Stokes $\quad \Delta \mathbf{u} = \nabla p, \qquad \nabla \cdot \mathbf{u} = 0,$

Helmholtz $\quad (-\Delta - k^2) u = f,$

Schrödinger $\quad (-\Delta + V)\, \Psi = i \frac{\partial \Psi}{\partial t},$

Maxwell $\quad \begin{cases} \nabla \cdot \mathbf{E} = \rho \qquad \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \cdot \mathbf{B} = 0 \qquad \nabla \times \mathbf{B} = \mathbf{J} + \frac{\partial \mathbf{E}}{\partial t} \end{cases}$

Boundary formulations are frequently optimal solvers:

Accurate: The computational error should be roughly $\kappa\,\varepsilon$, where ε is the machine precision and κ is the actual condition number.

Efficient: The CPU time requirement should be roughly proportional to N, the actual complexity of the problem.

Robust: The computation should be black-box with no need for fine-tuning, parameter-selection, pre-conditioning *et c.* In particular, delicate mesh-requirements currently form a major obstacle.

Under some conditions, methods exist that satisfy these criteria — FEM+multigrid, FFT, BIE + FMM, *et c.*

When should boundary methods be used?

The technique of solving boundary value problems (BVP) via fast iterative techniques for solving boundary integral equations (BIE) works best when

(1) there is no body force,

(2) the differential operator has constant coefficients
(or piece-wise constant coefficients),

(3) the BVP is linear,

(4) the BIE that is used is well-conditioned.

Over the last 20 years, much progress has been made in extending the technique to overcome the apparent obstreperousness of problems violating conditions (1), (2) and (3).

The recent development of **direct** solvers has done much to overcome obstacle (4).

A problem with periodic micro-structure

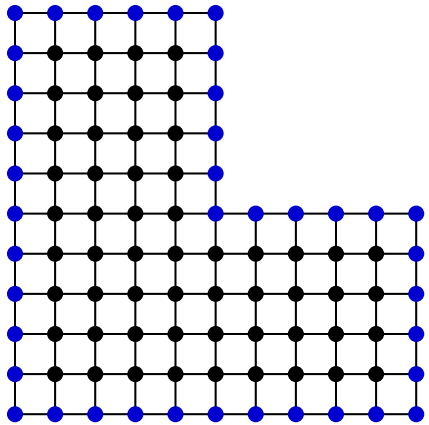
The lattice Laplace equation

$$\begin{cases} -\Delta \mathbf{u}(m) = 0, & m \in \Omega \subset \mathbb{Z}^2, \\ \mathbf{u}(m) = \mathbf{g}(m), & m \in \Gamma = \partial\Omega. \end{cases}$$

where $m = (m_1, m_2)$ is an integer index and

$$\begin{aligned} \Delta \mathbf{u}(m_1, m_2) =& \mathbf{u}(m_1 - 1, m_2) - 2\mathbf{u}(m_1, m_2) + \mathbf{u}(m_1 + 1, m_2) + \\ & \mathbf{u}(m_1, m_2 - 1) - 2\mathbf{u}(m_1, m_2) + \mathbf{u}(m_1, m_2 + 1), \end{aligned}$$

has a well-conditioned boundary formulation on Γ.



Example:

Implementation of BIE + FMM for modeling radar scattering problems at Boing led to an increase in problem size of 1–2 orders of magnitude, and an increase in accuracy from 10^{-2} to 10^{-9}.

Let us look at three different stealth aircraft:

- F-117 Nighthawk, developed 1978 – 1983.
- B-2 Spirit, developed 1985 – 1993.
- F-22 Raptor, developed 1994 – ?.

(Incidentally, a single B-2 costs about 2.2 billion dollars; while the budget for the F-22 program at this point exceeds 72 billion dollars.)

The Fast Multipole Method

Given a set of points $\{x_n\}_{n=1}^N \subset \mathbb{R}^2$ and a set of "charges" $\{q_n\}_{n=1}^N$, consider the problem of evaluating the harmonic potential u caused by the charges

$$u(x_n) = \sum_{\substack{m=1 \\ m \neq n}}^{N} \log|x_n - x_m| q_m, \qquad \text{for } n = 1, \ldots, N.$$

If direct summation is used, $O(N^2)$ floating point operations are required.

The Fast Multipole Method performs this task in $O(N)$ operations.

The method works for a wide class of kernels, but for simplicity, we will use the logarithmic kernel in 2D in all formulas.

What the FMM is good for:

- Evaluation of potential fields (gravitational fields in cosmology, electric force fields in molecular dynamics, etc).

- Application of integral operators: Given a function σ that is defined on a set Γ, evaluate

$$u(x) = \int_{\Gamma} K(x, y)\sigma(y)\, dy.$$

  Useful for rapid solution of integral equations.

- Application of a unitary matrix to another matrix. (Actually; yes.) Useful in constructing *very* fast methods for computing SVDs, etc.

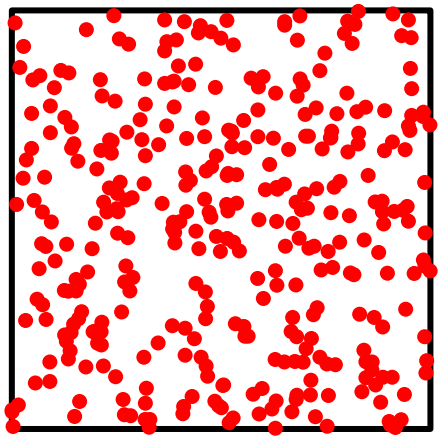
- Non-uniform FFT, interpolation, expansion in orthogonal bases (such as spherical harmonics), fast Gauss transform, ...

How it works: Suppose first that we have

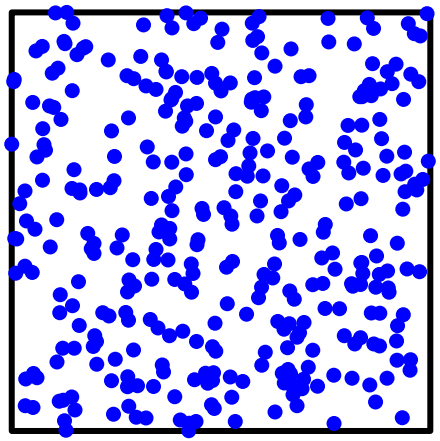
- M source points $\{y_m\}_{m=1}^{M} \subset \Omega_{\mathrm{S}} \subset \mathbb{C}$,
- N target points $\{x_n\}_{n=1}^{N} \subset \Omega_{\mathrm{T}} \subset \mathbb{C}$,

and that Ω_{S} and Ω_{T} are separate sets.
Using complex arithmetic, we want to evaluate

$$u(x_m) = \sum_{n=1}^{N} \log(x_m - y_n)\, q_n, \qquad \text{for } m = 1, \ldots, M.$$



Ω_{S}

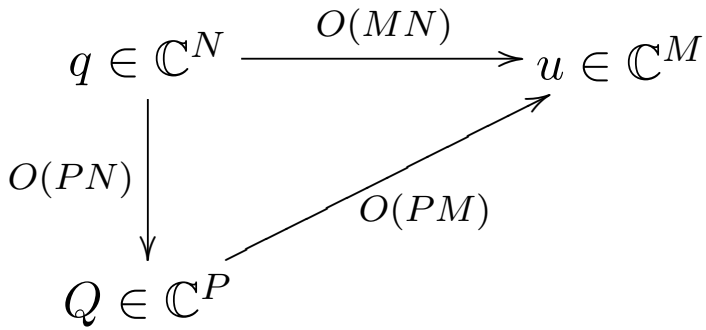


Ω_{T}

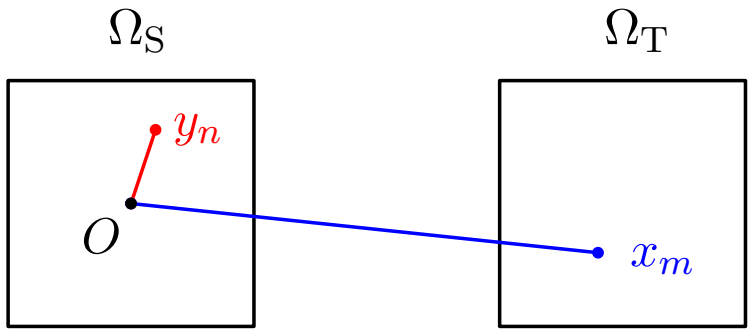
Let $Q = [Q_0, Q_1, \ldots, Q_P]$ denote the P-term multipole expansion of the charges in Ω_{S}. (In other words, Q_0 is the total charge, Q_1 is the dipole moment of the charge, Q_2 is the quadrupole moment, etc.)

Then divide the computation into two steps:

1. Compute a multipole expansion $Q \in \mathbb{C}^P$ from $q \in \mathbb{C}^N$.
2. Evaluate $u \in \mathbb{C}^M$ from $Q \in \mathbb{C}^P$.

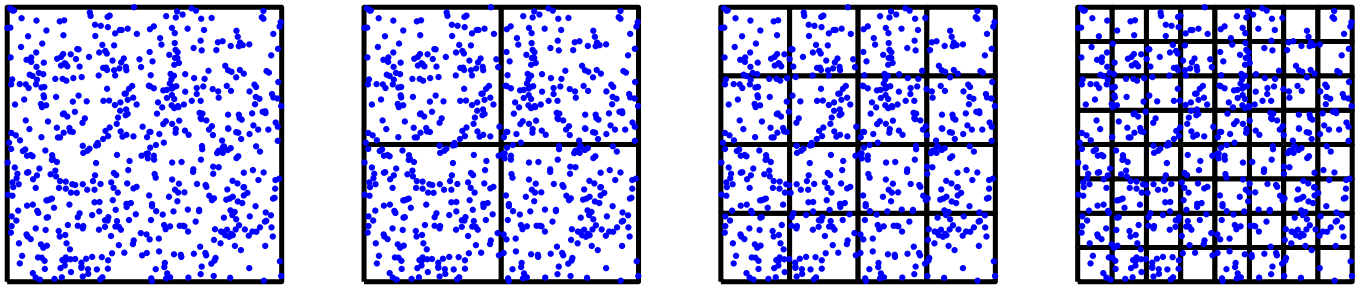


The total cost is proportional to $PM + PN$.
$P = O(|\log \varepsilon|)$, so the total cost is $O(|\log \varepsilon|(M + N))$.



$$
\begin{aligned}
u(x_m) &= \sum_{n=1}^{N} \log(x_m - y_n) q_n \\
&= \sum_{n=1}^{N} q_n \big(\log x_m + \log(1 - y_n/x_m)\big) \\
&\approx \sum_{n=1}^{N} q_n \big(\log x_m + \sum_{p=1}^{P} \frac{-1}{p} \left(\frac{y_n}{x_m}\right)^p\big) \\
&= \underbrace{\left(\sum_{n=1}^{N} q_n\right)}_{=:Q_0} \log x_m + \sum_{p=1}^{P} \underbrace{\left(\sum_{n=1}^{N} q_n y_n^p\right)}_{=:Q_p} \frac{-1}{p\, x_m^p} \\
&= Q_0 \log x_m + \sum_{p=1}^{P} Q_p \frac{-1}{p\, x_m^p}.
\end{aligned}
$$

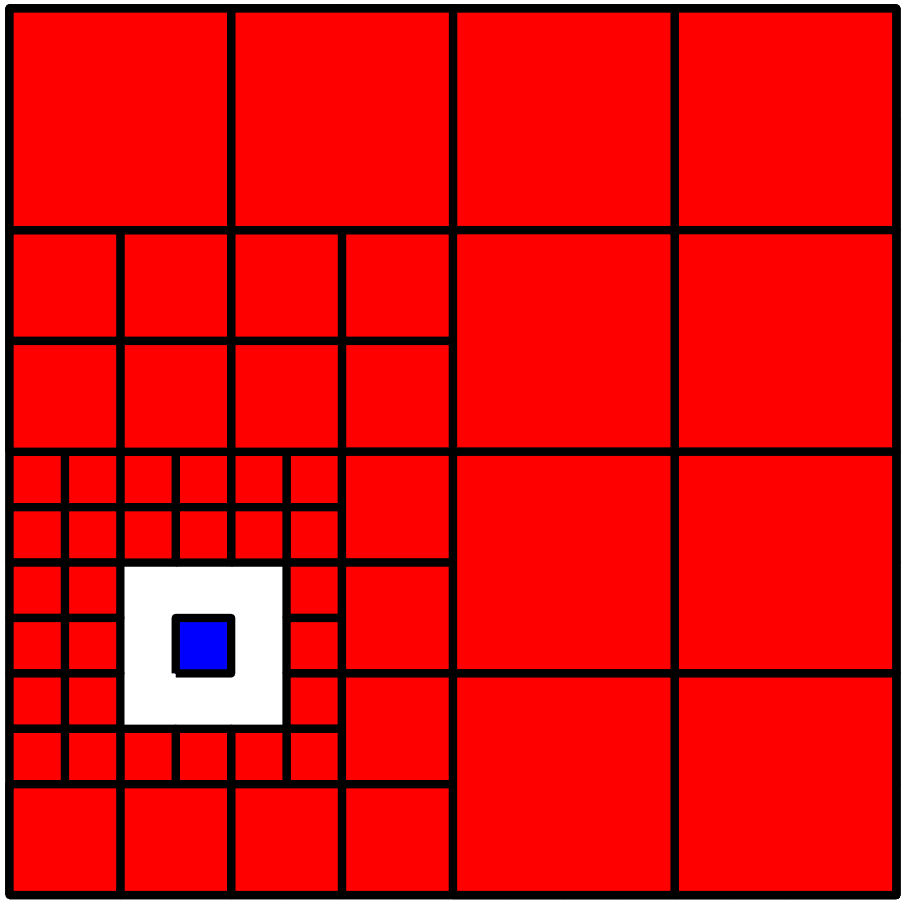
In the case where the target and the source regions are identical, we hierarchically subdivide the computational domain into boxes.



Level 1 Level 2 Level 3 Level 4

Step 1: For each box on every level, compute the multipole expansion of the charges inside it. Cost $\sim P\,N\,\log N$.

Step 2: For each particle, evaluate its interaction with the remaining particles by replacing charges in far-away boxes by as large a box as possible. Cost $\sim P\,N\,\log N$.



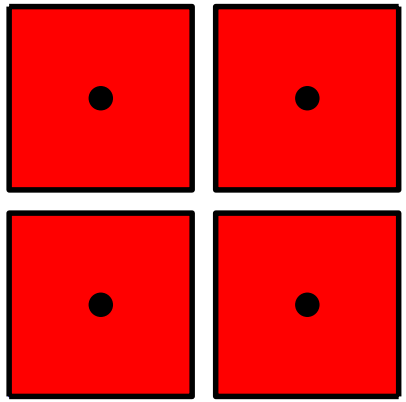
The algorithm just described is called Barnes-Hut.

It reduces the computational cost from $O(N^2)$ to $O(N \log N)$.

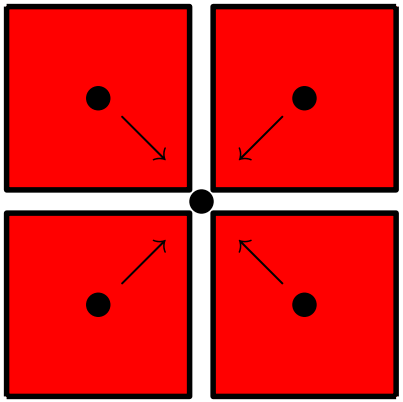
However, the constant is large.

We can accelerate Step 1 – the computation of the multipole moments – from $O(N \log N)$ to $O(N)$ by computing them hierarchically.

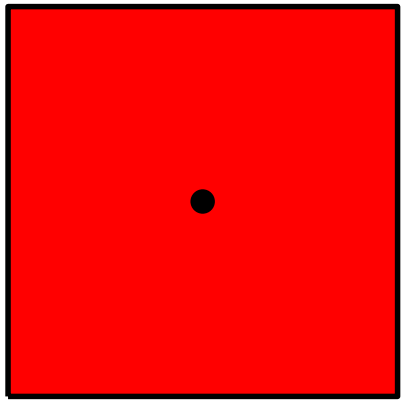
If we know the multipole expansions of four boxes that form a larger box on the coarser level, we can compute the multipole expansion of the larger box from the four expansions.



We know the mpole expansions for these four centers.



Merge the four expansions to form the new one.

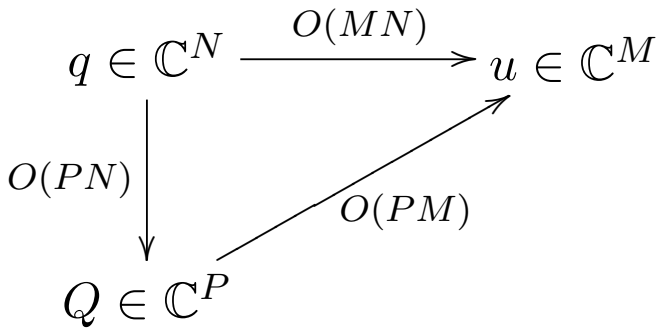


The new expansion.

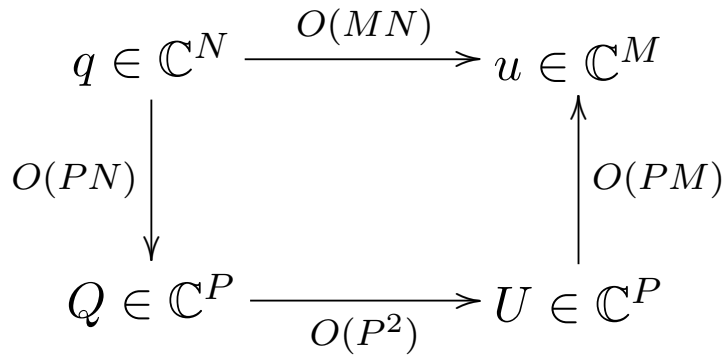
In the Fast Multipole Method, there is also a downward sweep through which the potentials are computed from the multipole expansions.

To make this work, we keep for each box track of both an outgoing representation Q (the multipole expansion), and an *incoming* representation U (an expansion of the potential in harmonic polynomials).

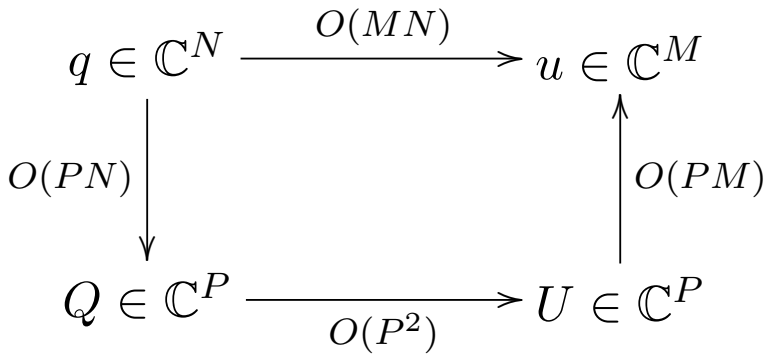
The total cost is $O(|\log \varepsilon| \, N)$.



Barnes-Hut



FMM



The scheme can be accelerated as follows:

- The representations Q and U can be formed in such a way that the map $Q \mapsto U$ is diagonal and requires only $O(P)$ operations.
  - Especially important in 3D.
  - Crucially important for oscillatory problems.
- Adaptive compression can be used to obtain optimally compressed representations (*i.e.* the smallest possible P is used).
  - The Singular Value Decomposition can be used.
  - Tabulation is better.

A (surprisingly?) simple way of representing functions.

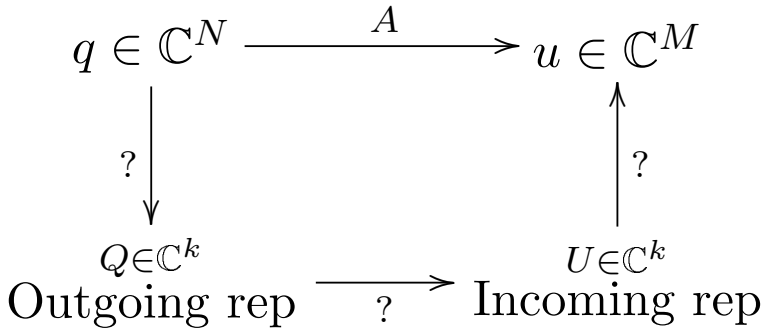
Again, let us assume that we are given

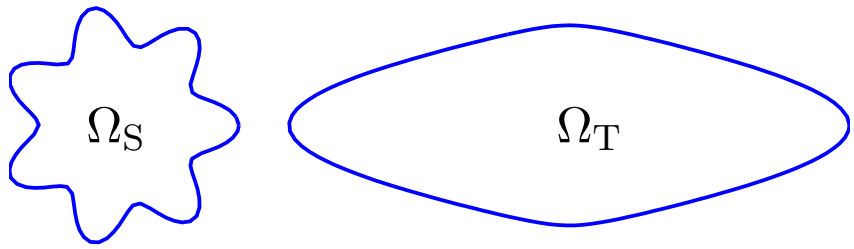
- M source points $\{y_m\}_{m=1}^M \subset \Omega_{\mathrm{S}} \subset \mathbb{C}$,
- N target points $\{x_n\}_{n=1}^N \subset \Omega_{\mathrm{T}} \subset \mathbb{C}$.

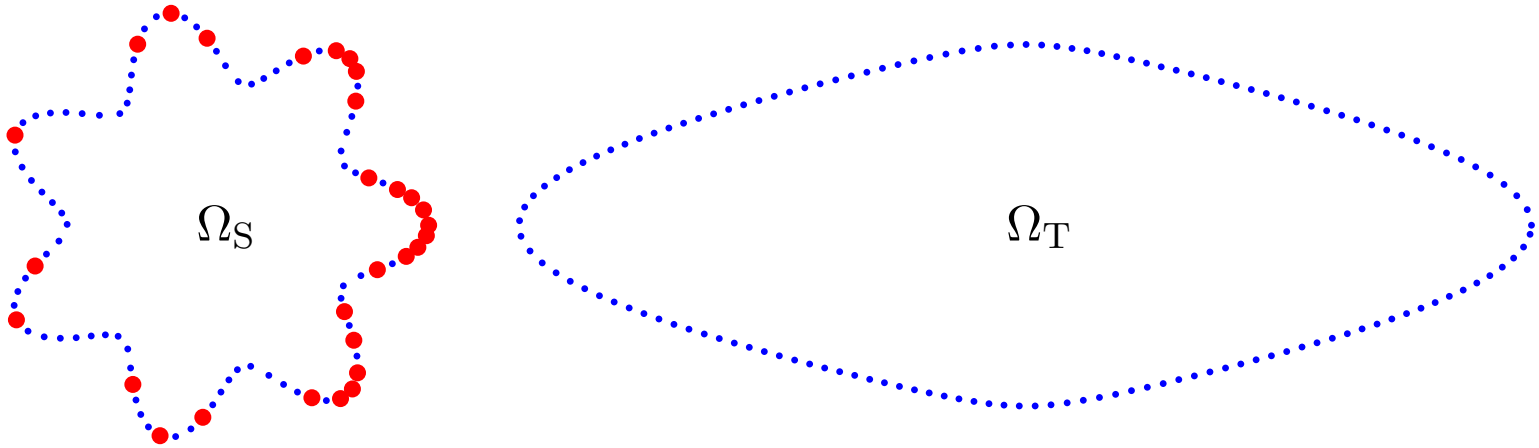
We want to evaluate

$$u(x_m) = \sum_{n=1}^{N} \log(x_m - y_n) q_n, \qquad \text{for } m = 1, \ldots, M.$$

Suppose that the $M \times N$ matrix A with entries $\log(x_m - y_n)$ has rank k. We want to find an efficient way of representing q and u:

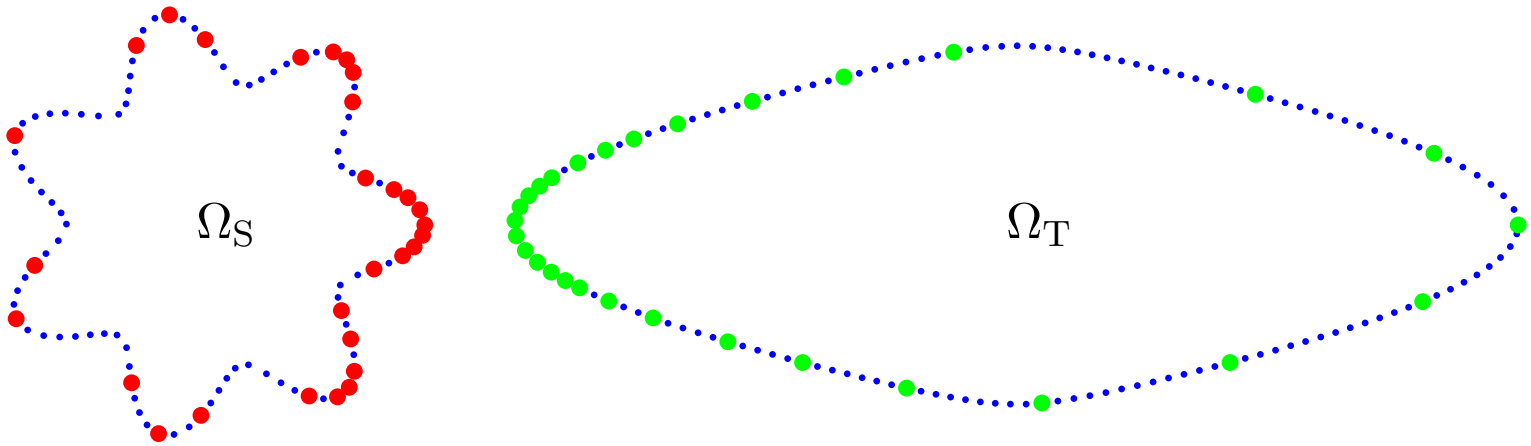




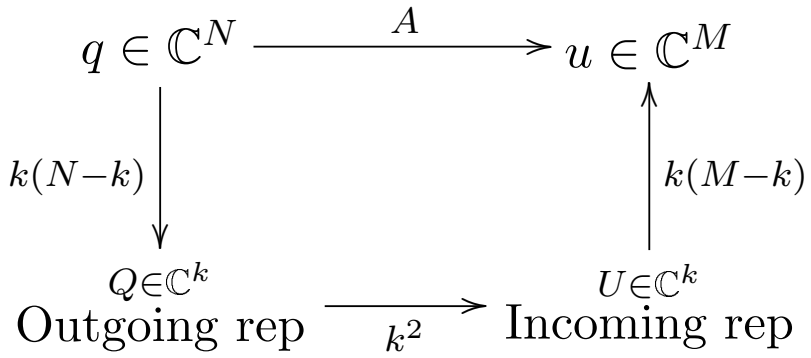


We can find k source points (marked with red dots) with the property that any potential on Ω_T caused by charges in Ω_S, can be reproduced by placing charges on these k points.

Important: The choice of the points depends only on the geometry (not on the given charge distribution).



Similarly, we can find k points in Ω_T with the property that any potential caused by charges on Ω_S can be interpolated locally if its values are known at the k green points.



Benefits:

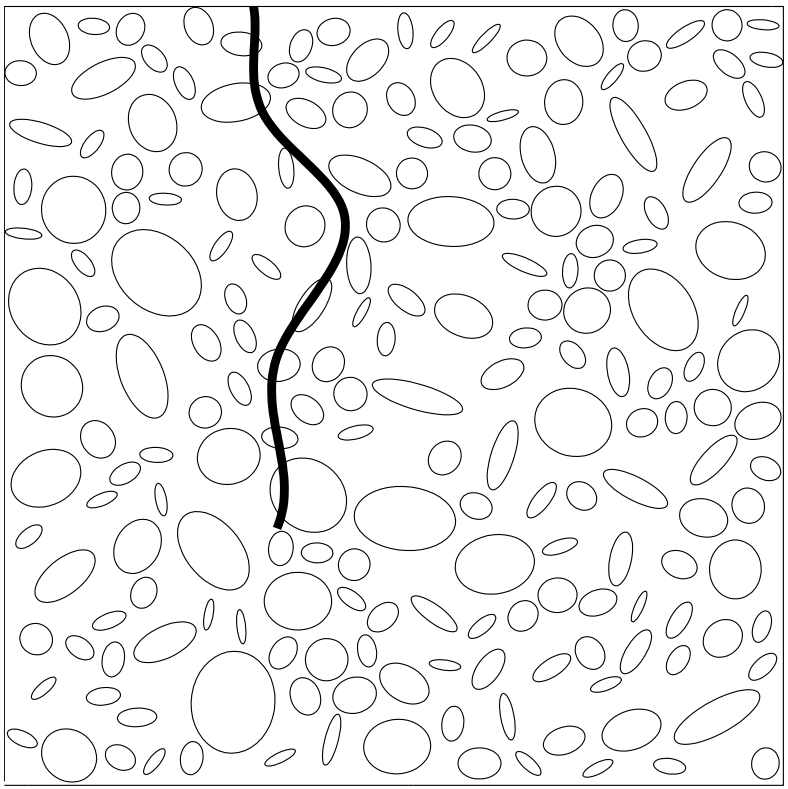
- The rank is optimal.
- The projection and interpolation are cheap.
- The projection and interpolation are well-conditioned.
- Finding the k points is cheap.
- The map $Q \mapsto U$ is simply a restriction of the original map.
- Interaction between **adjacent** boxes can be compressed (no buffering is required).
- *Very useful for constructing direct solvers.*

A highly optimized version of the FMM for one-dimensional non-oscillatory problems has been developed:

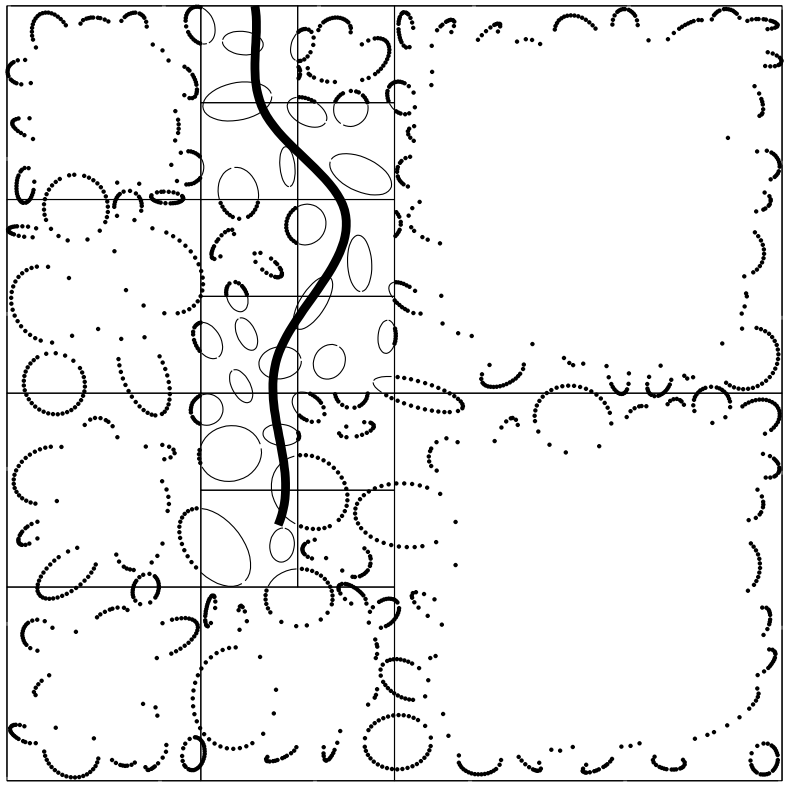
- Kernel independent.
- Entirely stable and with adaptive error control.

The break-even point at 14 digits of accuracy is 90.

Example: Crack in a bi-phase material



Uncompressed



Compressed

The claims above are actually pure linear algebra:

Lemma 1 (Gu & Eisenstat) *Let A be an $M \times N$ matrix of rank k. There exists a factorization*

$$A = X \circ \tilde{A} \circ Y, \tag{3}$$

where

- *X is a $k \times N$ matrix containing a $k \times k$ unit submatrix. No entry of X has magnitude larger than 1.*
- *Y is an $M \times k$ matrix containing a $k \times k$ unit submatrix. No entry of Y has magnitude larger than 1.*
- *$\tilde{A}$ is a $k \times k$ submatrix of A.*

Recall that

$$u = A\,q.$$

Setting

$$Q = Y\,q, \qquad \text{and} \qquad U = \tilde{A}\,Q,$$

we find that by virtue of (3), u can be reconstructed via

$$u = X\,U.$$

There is also an interesting connection to interpolation of functions:

Lemma 2 *Let A be a k-dimensional space of real-valued functions on a compact set Ω. There exists a set of k points $\{x_j\}_{j=1}^k \subset \Omega$, and a set of functions $\{\varphi_j\}_{j=1}^k$ such that*

$$u(x) = \sum_{j=1}^{k} u(x_j)\varphi_j(x),$$

for any $u \in A$. Moreover, for $j = 1, \ldots, k$,

$$|\varphi_j(x)| \leq 1, \qquad \forall x \in \Omega.$$

Direct solvers:

It has recently been demonstrated that in many environments there exist $O(N \log N)$ and $O(N)$ methods for *inverting* the boundary integral operator

$$u(x) \mapsto u(x) + \int_\Gamma K(x,y)u(y)\,dy.$$

Benefits:

- Increased robustness.
- Less sensitive to ill-conditioned problems.
  Important for solving near-resonant scattering problems.
- Very efficient for problems with multiple right-hand sides.
- Can be used to construct spectral decompositions of operators.

Work in progress.

Research directions:

- **Direct solvers.**
- Continued development of Fast Multipole Methods.
- Harmonic analysis; representation of function spaces, interpolation.
- Application of hierarchical techniques to multiscale modelling. The new understanding of techniques for representation of functions is crucial here.
- Application of Fast Multipole Methods to various fields (modelling of macro-molecules, semi-conductors, emulsions, molecular dynamics).
- Local collaborations!