Introduction to fast summation methods

Gunnar Martinsson

The University of Oxford

Slides available at: `http://people.maths.ox.ac.uk/martinsson/`

We consider a basic summation problem such as

$$u_i = \sum_{j=1}^{N} G(\boldsymbol{x}_i - \boldsymbol{x}_j)\, q_j, \qquad i = 1,\, 2,\, \dots,\, N,$$

where

$\{\boldsymbol{x}_i\}_{i=1}^N$ is a given set of points in $\mathbb{R}^d$, where
$\{q_i\}_{i=1}^N$ is a given set of real numbers which we call *sources,* and where
$\{u_i\}_{i=1}^N$ is a sought set of real numbers which we call *potentials.*

We have in mind the "kernel" G being a classical fundamental solution such as

$$G(\boldsymbol{x}) = -\frac{1}{2\pi}\log|\boldsymbol{x}|, \quad \text{or} \quad G(\boldsymbol{x}) = \frac{1}{4\pi|\boldsymbol{x}|}, \quad \text{or} \quad G(\boldsymbol{x}) = \frac{e^{i\kappa|\boldsymbol{x}|}}{4\pi|\boldsymbol{x}|}, \quad \text{or} \quad , \dots$$

"Evaluation of N potentials induced by N sources."

We consider a basic summation problem such as

$$u_i = \sum_{j=1}^{N} G(\boldsymbol{x}_i - \boldsymbol{x}_j)\, q_j, \qquad i = 1,\, 2,\, \dots,\, N,$$

where

$\{\boldsymbol{x}_i\}_{i=1}^N$ is a given set of points in $\mathbb{R}^d$, where
$\{q_i\}_{i=1}^N$ is a given set of real numbers which we call *sources,* and where
$\{u_i\}_{i=1}^N$ is a sought set of real numbers which we call *potentials.*

We have in mind the "kernel" G being a classical fundamental solution such as

$$G(\boldsymbol{x}) = -\frac{1}{2\pi}\log|\boldsymbol{x}|, \quad \text{or} \quad G(\boldsymbol{x}) = \frac{1}{4\pi|\boldsymbol{x}|}, \quad \text{or} \quad G(\boldsymbol{x}) = \frac{e^{i\kappa|\boldsymbol{x}|}}{4\pi|\boldsymbol{x}|}, \quad \text{or} \quad , \dots$$

"Evaluation of N potentials induced by N sources."

We seek to:

- Reduce the computational complexity from $O(N^2)$ to $O(N)$. (Or $O(N \log^p N)$.)
- Meet any specified accuracy ε at tolerable cost. (Say $\varepsilon = 10^{-3},\, 10^{-6},\, 10^{-9}, \dots$)

We consider a basic summation problem such as

$$u_i = \sum_{j=1}^{N} G(\boldsymbol{x}_i - \boldsymbol{x}_j)\, q_j, \qquad i = 1,\, 2,\, \dots,\, N,$$

where

$\{\boldsymbol{x}_i\}_{i=1}^N$ is a given set of points in $\mathbb{R}^d$, where
$\{q_i\}_{i=1}^N$ is a given set of real numbers which we call *sources,* and where
$\{u_i\}_{i=1}^N$ is a sought set of real numbers which we call *potentials.*

We have in mind the "kernel" G being a classical fundamental solution such as

$$G(\boldsymbol{x}) = -\frac{1}{2\pi}\log|\boldsymbol{x}|, \quad \text{or} \quad G(\boldsymbol{x}) = \frac{1}{4\pi|\boldsymbol{x}|}, \quad \text{or} \quad G(\boldsymbol{x}) = \frac{e^{i\kappa|\boldsymbol{x}|}}{4\pi|\boldsymbol{x}|}, \quad \text{or} \quad , \dots$$

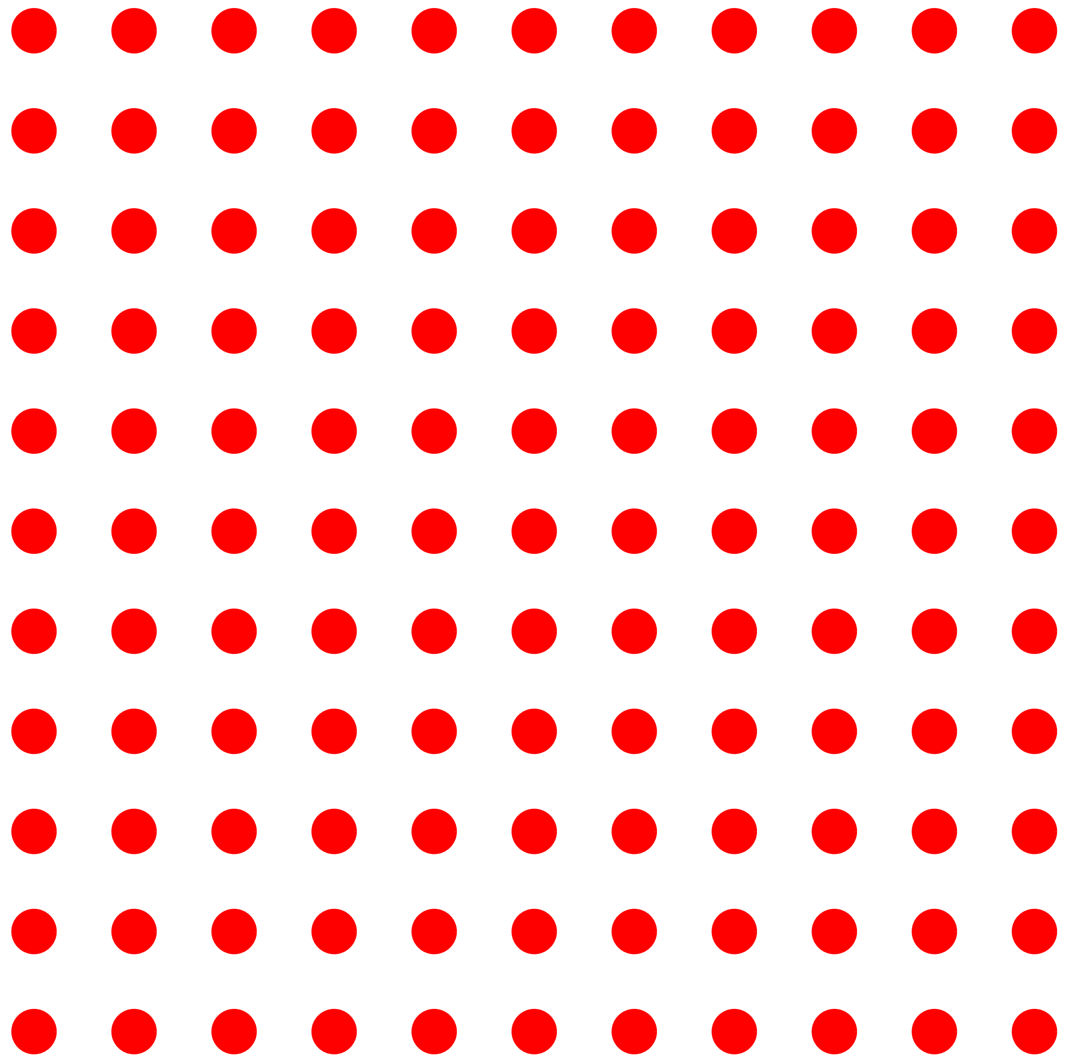
"Evaluation of N potentials induced by N sources."

We seek to:

- Reduce the computational complexity from $O(N^2)$ to $O(N)$. (Or $O(N\log^p N)$.)
- Meet any specified accuracy ε at tolerable cost. (Say $\varepsilon = 10^{-3},\ 10^{-6},\ 10^{-9}, \dots$)

☞ Having a knob that controls accuracy is very useful.

Easy special case. Suppose the points $\{\boldsymbol{x}_i\}_{i=1}^N$ form a uniform lattice:

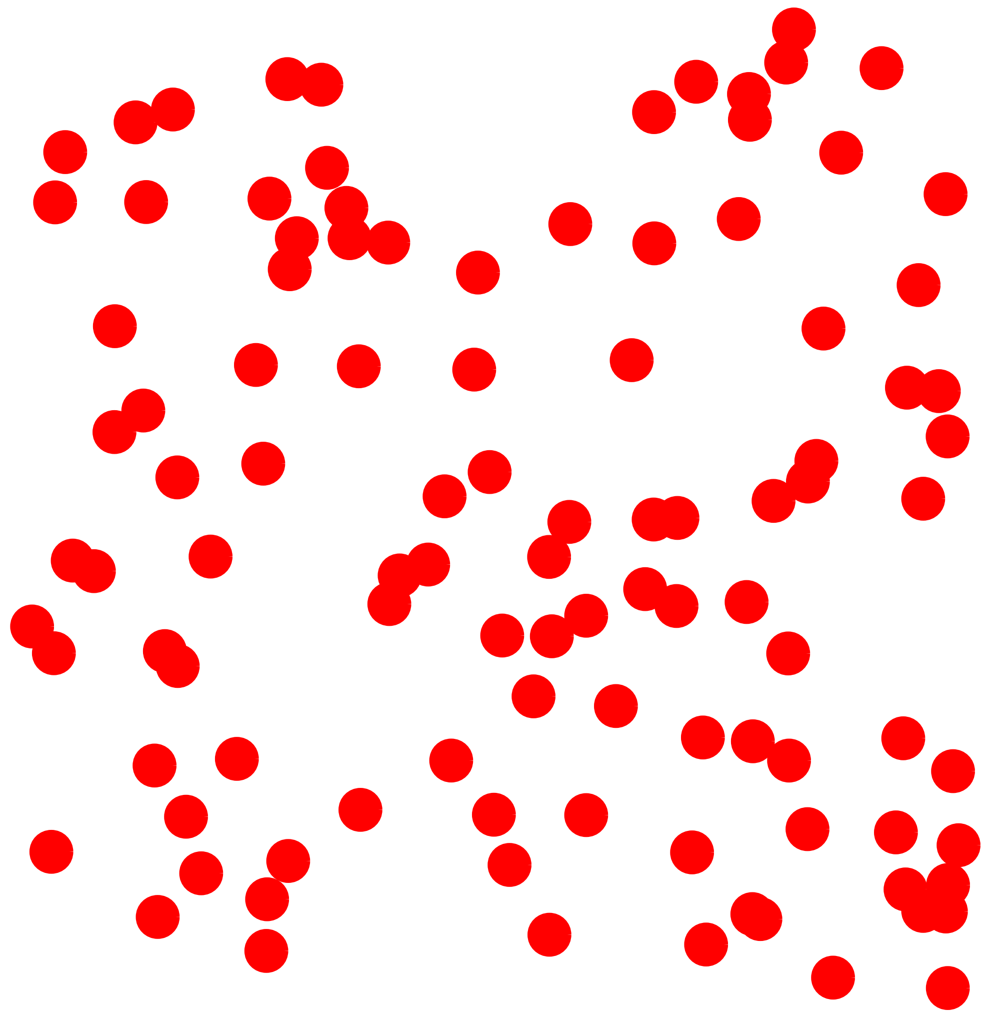


Then the sum simplifies to a convolution on a lattice, which is diagonalized by the discrete Fourier transform F. In other words, we have $u = G * q$ in physical space, so $F(u) = F(G)\,F(q)$ in Fourier space, and consequently (up to multiplicative factors)

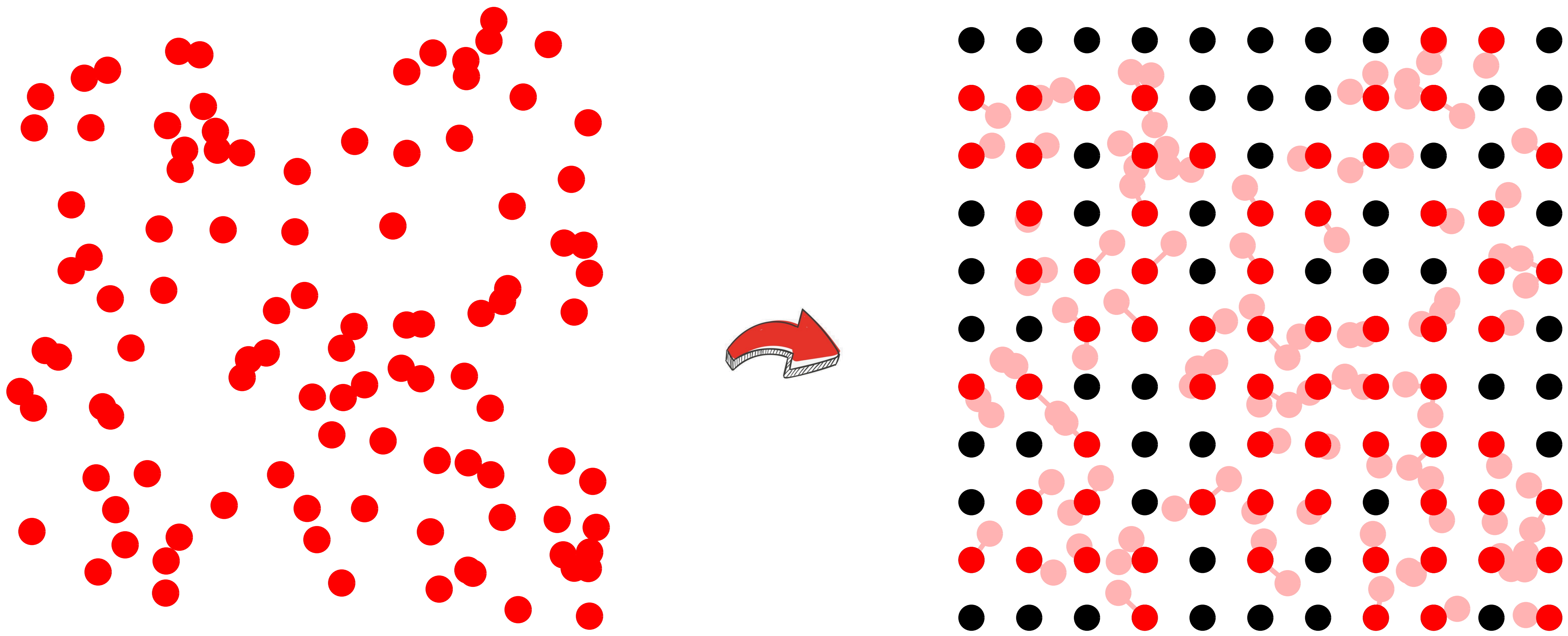
$$u = F^{-1}\left[F(G)\,F(q)\right].$$

F and F^{-1} can be applied rapidly via the FFT, and the resulting computational method is exact and very fast. (Some fussing is needed to get boundary conditions correct, etc.)

Fairly easy special case. Suppose the points $\{\boldsymbol{x}_i\}_{i=1}^N$ are "evenly" distributed in a box:



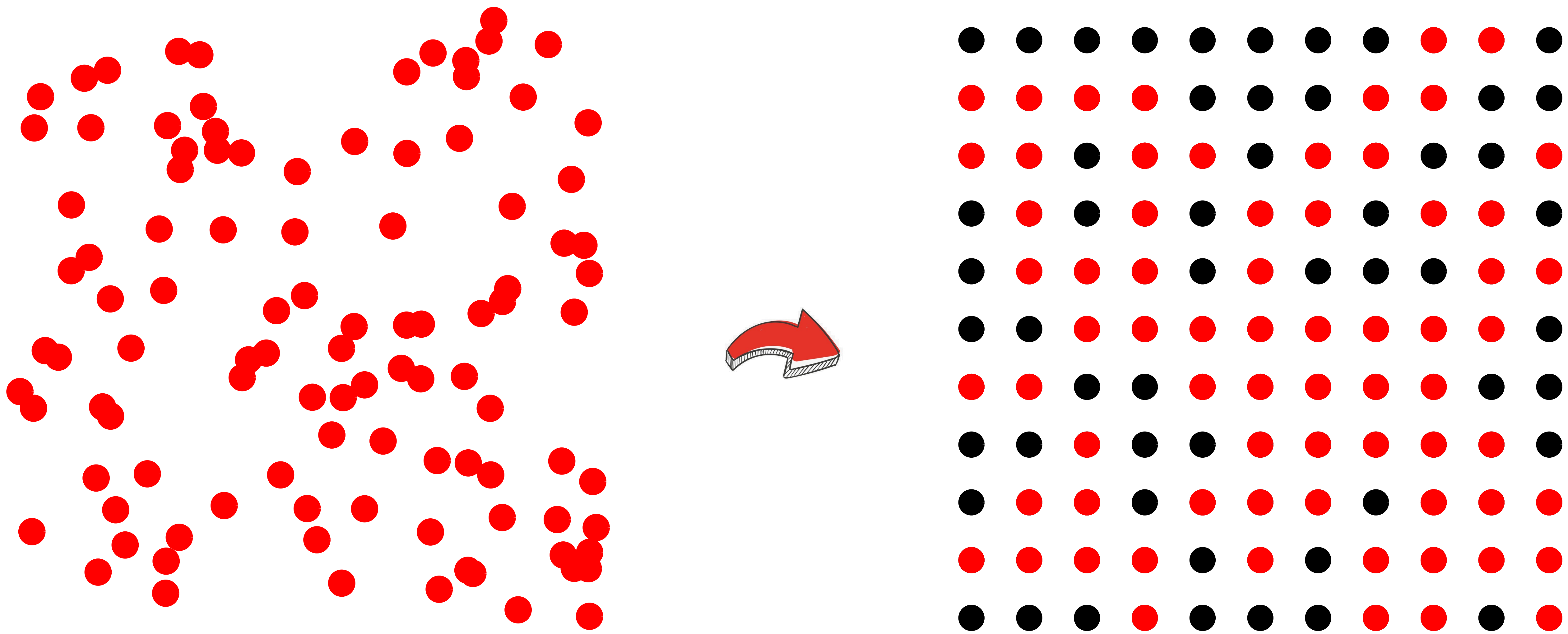
Fairly easy special case. Suppose the points $\{\boldsymbol{x}_i\}_{i=1}^N$ are "evenly" distributed in a box:



Here we could introduce an artificial uniform grid. Then we would map the actual charges to some "equivalent charges" on nearby grid points. Then use the FFT to evaluate long range interactions, and compute local corrections for near field interactions. Slightly messy, but works fine.

Alternative strategy: *Use a non-uniform FFT.*

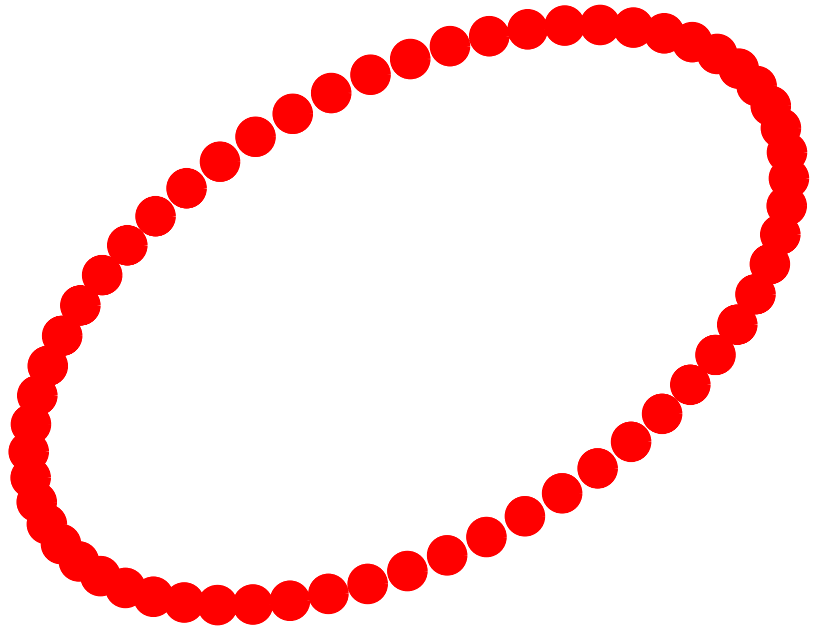
Fairly easy special case. Suppose the points $\{\boldsymbol{x}_i\}_{i=1}^{N}$ are "evenly" distributed in a box:



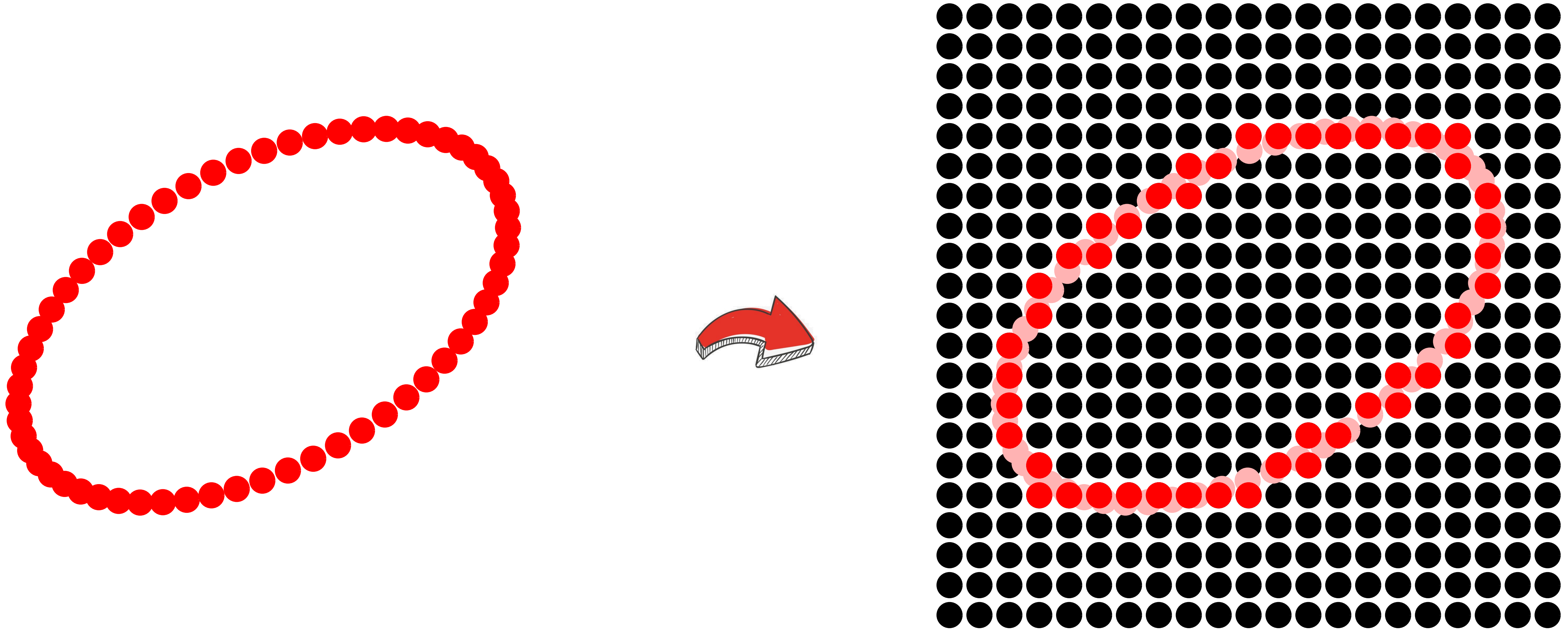
Here we could introduce an artificial uniform grid. Then we would map the actual charges to some "equivalent charges" on nearby grid points. Then use the FFT to evaluate long range interactions, and compute local corrections for near field interactions. Slightly messy, but works fine.

Alternative strategy: *Use a non-uniform FFT.*

Less easy case. What if the points are not evenly distributed at all ... ?



Less easy case. What if the points are not evenly distributed at all ... ?



Say we have n points along the curve shown. Then the uniform grid will need $N = O(n) \times O(n)$ points. So even using the FFT, we get complexity $\sim N \log N \sim n^2 \log n$. Geometries that involve local refinement would be even worse.

The Fast Multipole Method (early 1980's):

- Can handle non-uniform distributions.
- Linear complexity.
- Any requested tolerance ε can be met.

  The price to pay is that complexity will be

  $O((\log(1/\varepsilon))^p N)$.

L. Greengard

V. Rokhlin

Problem definition: Consider the task of evaluating the sum

$$(1) \qquad u_i = \sum_{j=1}^{N} G(\boldsymbol{x}_i - \boldsymbol{x}_j)\, q_j, \qquad i = 1,\,2,\,\dots,\,N,$$

where

$\{\boldsymbol{x}_i\}_{i=1}^N$	is a given set of points in a square Ω in the plane, where
$\{q_i\}_{i=1}^N$	is a given set of real numbers which we call *sources,* and where
$\{u_i\}_{i=1}^N$	is a sought set of real numbers which we call *potentials.*

The *kernel* G is given by

$$(2) \qquad G(\boldsymbol{x} - \boldsymbol{y}) = \begin{cases} \log(\boldsymbol{x} - \boldsymbol{y}), & \text{when } \boldsymbol{x} \neq \boldsymbol{y} \\ 0 & \text{when } \boldsymbol{x} = \boldsymbol{y}. \end{cases}$$

Note: A point $\boldsymbol{x} \in \mathbb{R}^2$ is represented by the complex number

$$\boldsymbol{x} = x_1 + i\, x_2 \in \mathbb{C}.$$

Then the kernel in (2) is a complex representation of the fundamental solution to the Laplace equation in $\mathbb{R}^2$ since

$$\log|\boldsymbol{x} - \boldsymbol{y}| = \text{Real}\big(\log(\boldsymbol{x} - \boldsymbol{y})\big).$$

(The factor of $-1/2\pi$ is suppressed.)

Special case: Sources and targets are separate

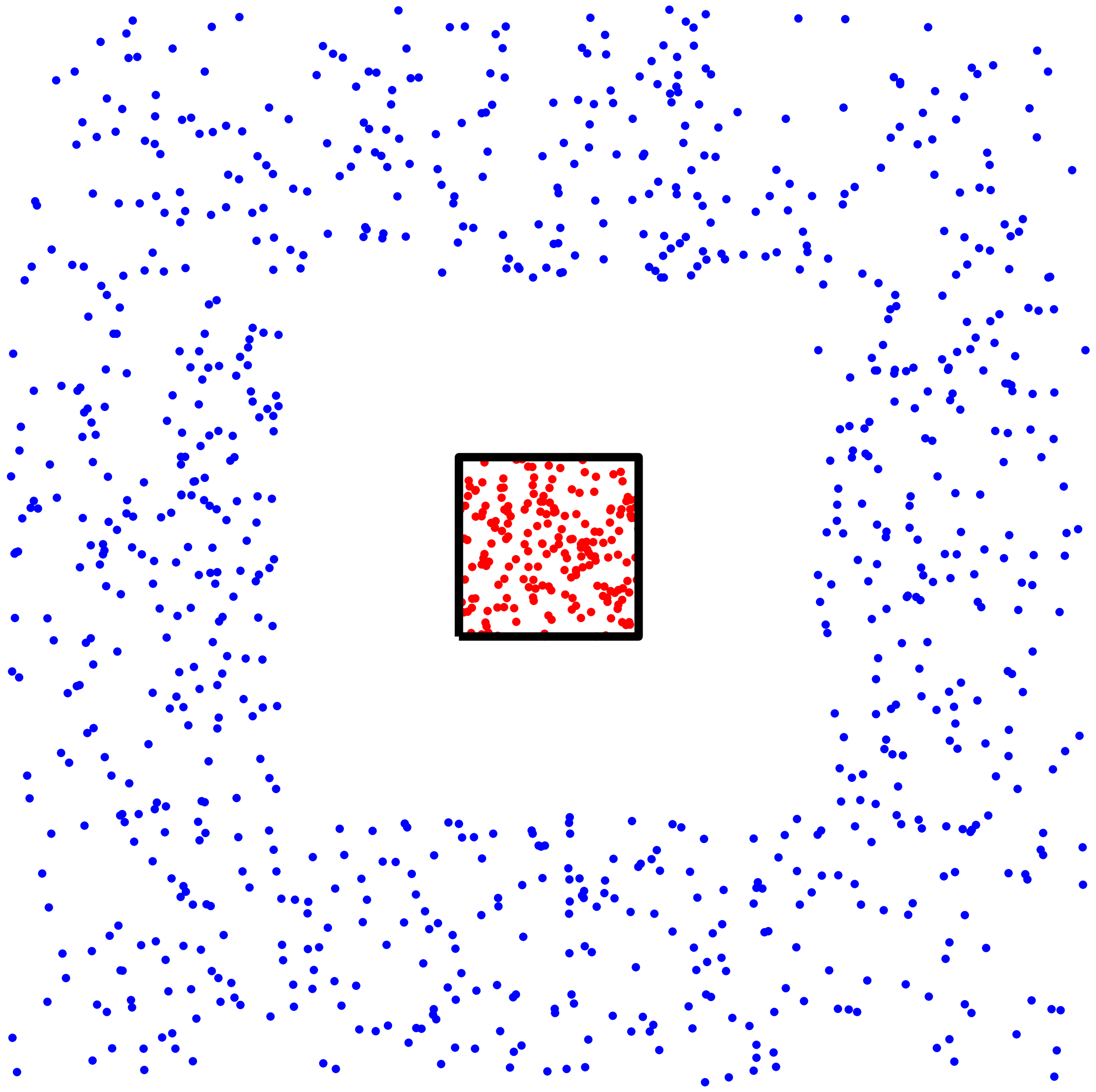
Charges q_j at locations $\{\boldsymbol{y}_j\}_{j=1}^{N}$.

Potentials u_i at locations $\{\boldsymbol{x}_i\}_{i=1}^{M}$.

$$u_i = \sum_{j=1}^{N} \log(\boldsymbol{x}_i - \boldsymbol{y}_j)\, q_j, \qquad i = 1,\,2,\,\dots,\,M$$

$$u_i = u(\boldsymbol{x}_i), \qquad i = 1,\,2,\,\dots,\,M$$

$$u(\boldsymbol{x}) = \sum_{j=1}^{N} \log(\boldsymbol{x} - \boldsymbol{y}_j)\, q_j$$



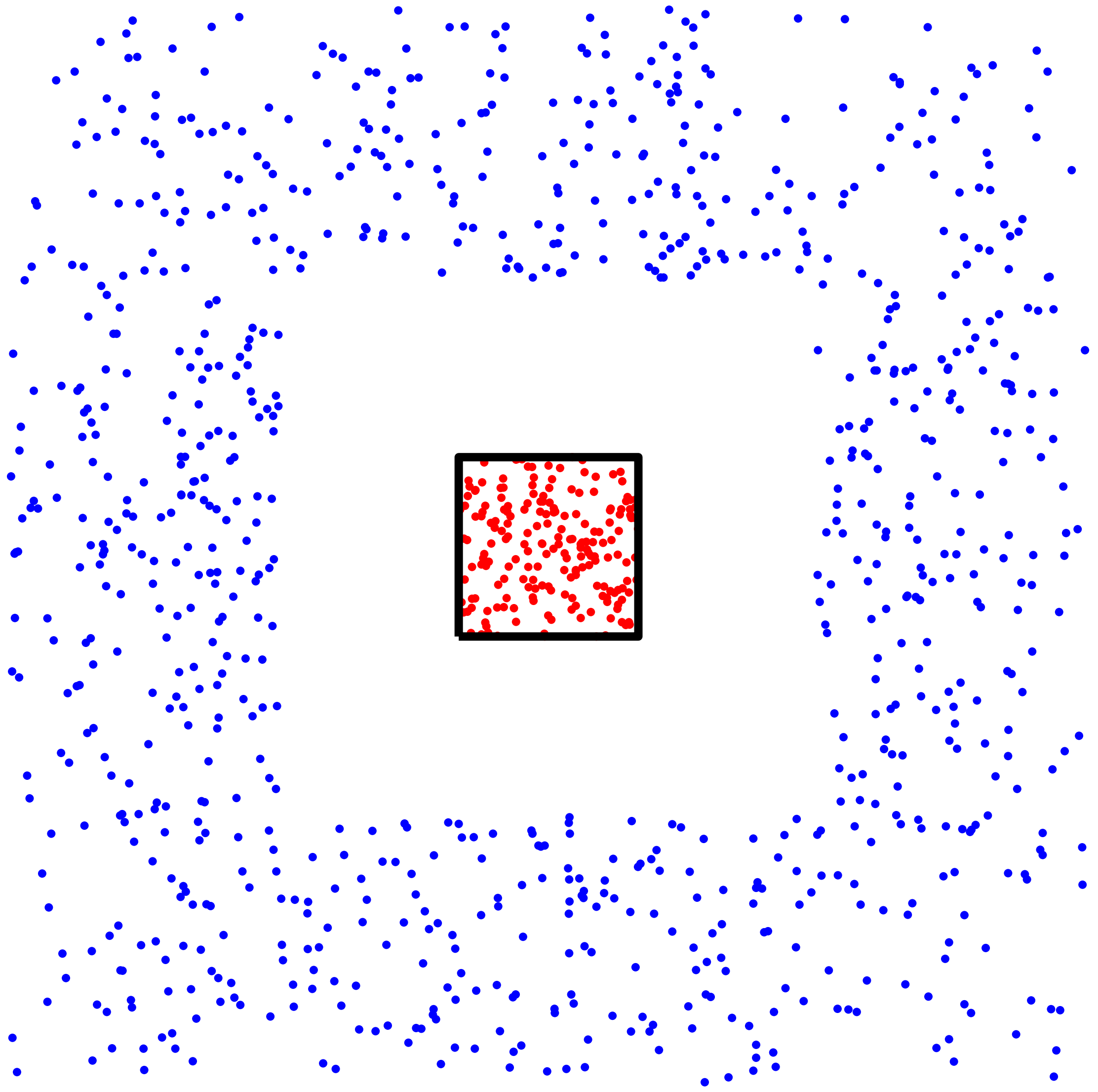
Direct evaluation

Cost is $O(M\,N)$.

Special case: Sources and targets are separate

Charges q_j at locations $\{\boldsymbol{y}_j\}_{j=1}^{N}$.

Potentials u_i at locations $\{\boldsymbol{x}_i\}_{i=1}^{M}$.



$$u_i = \sum_{j=1}^{N} \log(\boldsymbol{x}_i - \boldsymbol{y}_j)\, q_j, \qquad i = 1,\, 2,\, \dots,\, M$$

$$u_i = u(\boldsymbol{x}_i), \qquad i = 1,\, 2,\, \dots,\, M$$

$$u(\boldsymbol{x}) = \sum_{j=1}^{N} \log(\boldsymbol{x} - \boldsymbol{y}_j)\, q_j$$

Direct evaluation

Cost is $O(M\,N)$.

But recall that $\log(\boldsymbol{x} - \boldsymbol{y})$ admits the *separation of variables*

$$\log(\boldsymbol{x} - \boldsymbol{y}) = \log(\boldsymbol{x} - \boldsymbol{c})\,1 + \sum_{p=1}^{\infty} \frac{-1}{p}\frac{1}{(\boldsymbol{x} - \boldsymbol{c})^p}(\boldsymbol{y} - \boldsymbol{c})^p,$$

where $\boldsymbol{c}$ is the center of the source box.

Special case: Sources and targets are separate

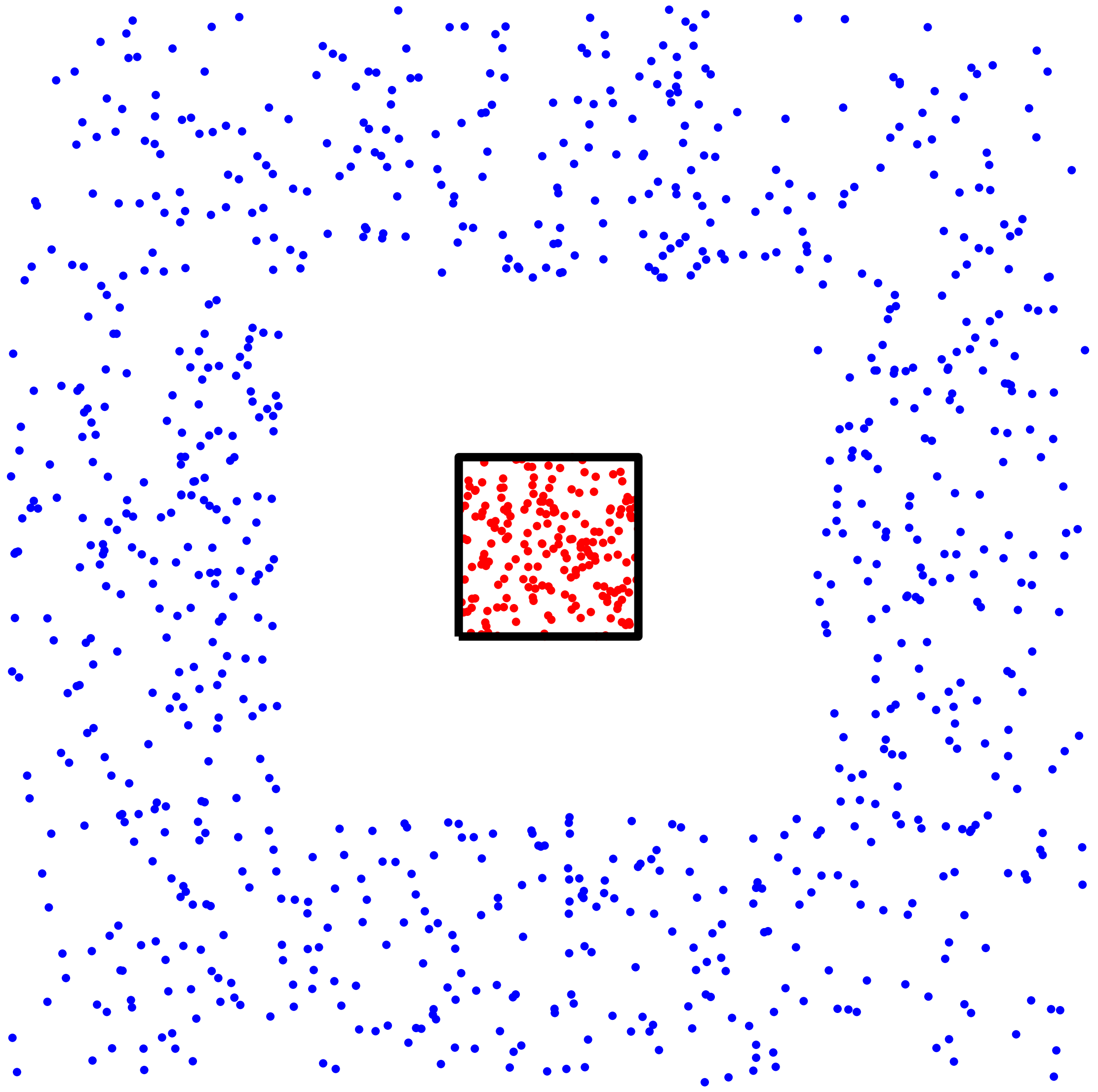
Charges q_j at locations $\{\boldsymbol{y}_j\}_{j=1}^N$.

Potentials u_i at locations $\{\boldsymbol{x}_i\}_{i=1}^M$.

$$u_i = \sum_{j=1}^N \log(\boldsymbol{x}_i - \boldsymbol{y}_j)\, q_j, \qquad i = 1,\, 2,\, \dots,\, M$$

$$u_i = u(\boldsymbol{x}_i), \qquad i = 1,\, 2,\, \dots,\, M$$

$$u(\boldsymbol{x}) = \sum_{j=1}^N \log(\boldsymbol{x} - \boldsymbol{y}_j)\, q_j$$



Multipole expansion:

It follows that $u(\boldsymbol{x}) = \log(\boldsymbol{x} - \boldsymbol{c})\hat{q}_0 + \sum_{p=1}^{\infty} \frac{1}{(\boldsymbol{x} - \boldsymbol{c})^p}\hat{q}_p$.

where $\hat{q}_0 = \sum_j^n q_j$ and $\hat{q}_p = -\frac{1}{p}\sum_{j=1}^N (\boldsymbol{y}_j - \boldsymbol{c})^p\, q_j$.

Special case: Sources and targets are separate

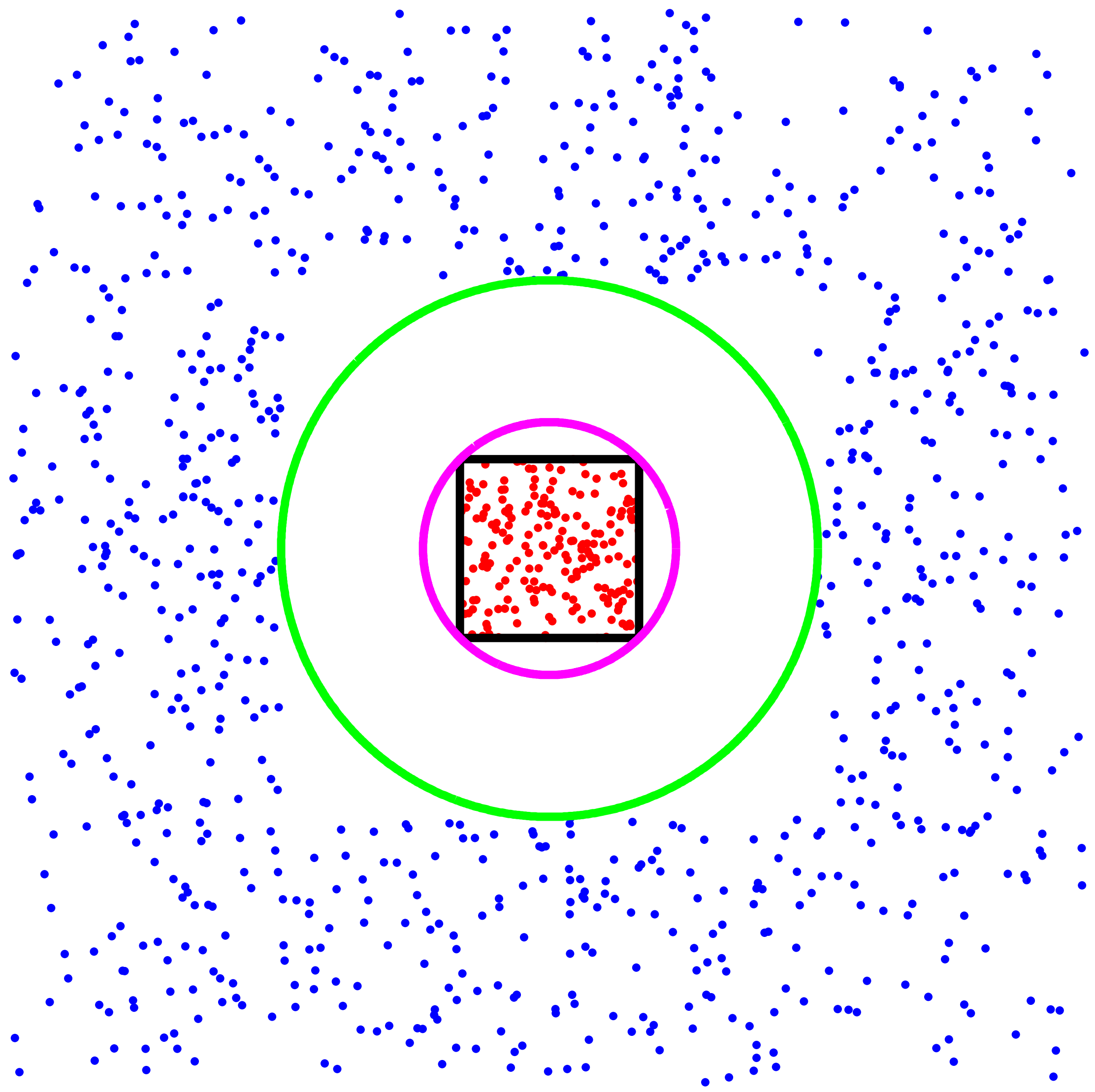
Charges q_j at locations $\{\boldsymbol{y}_j\}_{j=1}^N$.

Potentials u_i at locations $\{\boldsymbol{x}_i\}_{i=1}^M$.

$$u_i = \sum_{j=1}^N \log(\boldsymbol{x}_i - \boldsymbol{y}_j)\, q_j, \qquad i = 1, 2, \dots, M$$

$$u_i = u(\boldsymbol{x}_i), \qquad i = 1, 2, \dots, M$$

$$u(\boldsymbol{x}) = \sum_{j=1}^N \log(\boldsymbol{x} - \boldsymbol{y}_j)\, q_j$$



Multipole expansion — truncated to $P+1$ terms:

It follows that $u(\boldsymbol{x}) = \log(\boldsymbol{x} - \boldsymbol{c})\hat{q}_0 + \sum_{p=1}^P \frac{1}{(\boldsymbol{x}-\boldsymbol{c})^p}\hat{q}_p + E_P$ where $\hat{q}_p = -\frac{1}{p}\sum_{j=1}^N (\boldsymbol{y}_j - \boldsymbol{c})^p\, q_j$.

The approximation error E_P scales as $E_P \sim \left(\frac{r}{R}\right)^P = \left(\frac{\sqrt{2}a}{3a}\right)^P = \left(\frac{\sqrt{2}}{3}\right)^P$, where

$r = \sqrt{2}a$ is the radius of the magenta circle, and $R = 3a$ is the radius of the green circle.

Special case: Sources and targets are separate

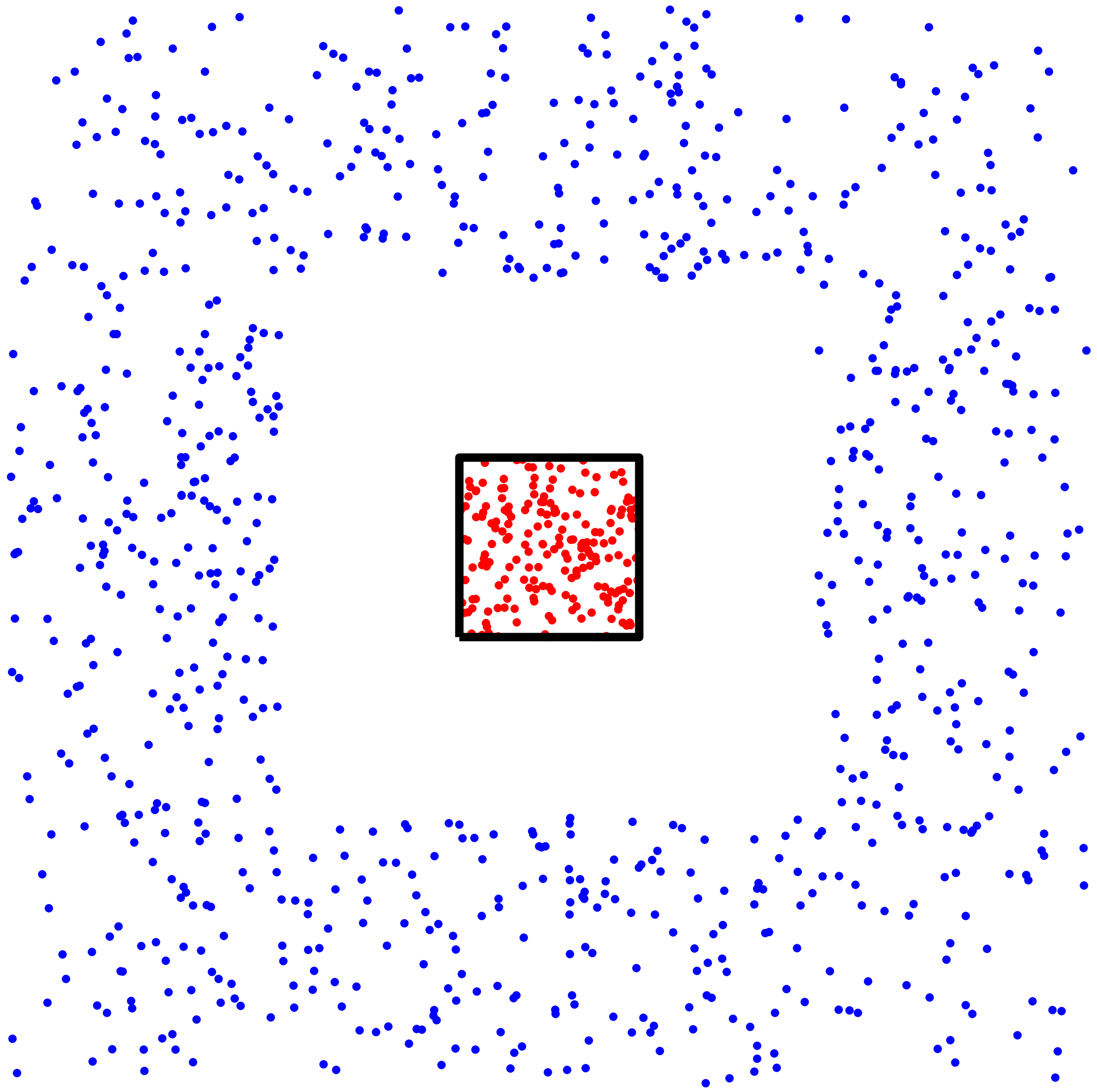
Charges q_j at locations $\{\boldsymbol{y}_j\}_{j=1}^{N}$.

Potentials u_i at locations $\{\boldsymbol{x}_i\}_{i=1}^{M}$.

$$u_i = \sum_{j=1}^{N} \log(\boldsymbol{x}_i - \boldsymbol{y}_j)\, q_j, \qquad i = 1,\, 2,\, \dots,\, M$$

$$u_i = u(\boldsymbol{x}_i), \qquad i = 1,\, 2,\, \dots,\, M$$

$$u(\boldsymbol{x}) = \sum_{j=1}^{N} \log(\boldsymbol{x} - \boldsymbol{y}_j)\, q_j$$



Multipole expansion — truncated to $P+1$ terms — costs:

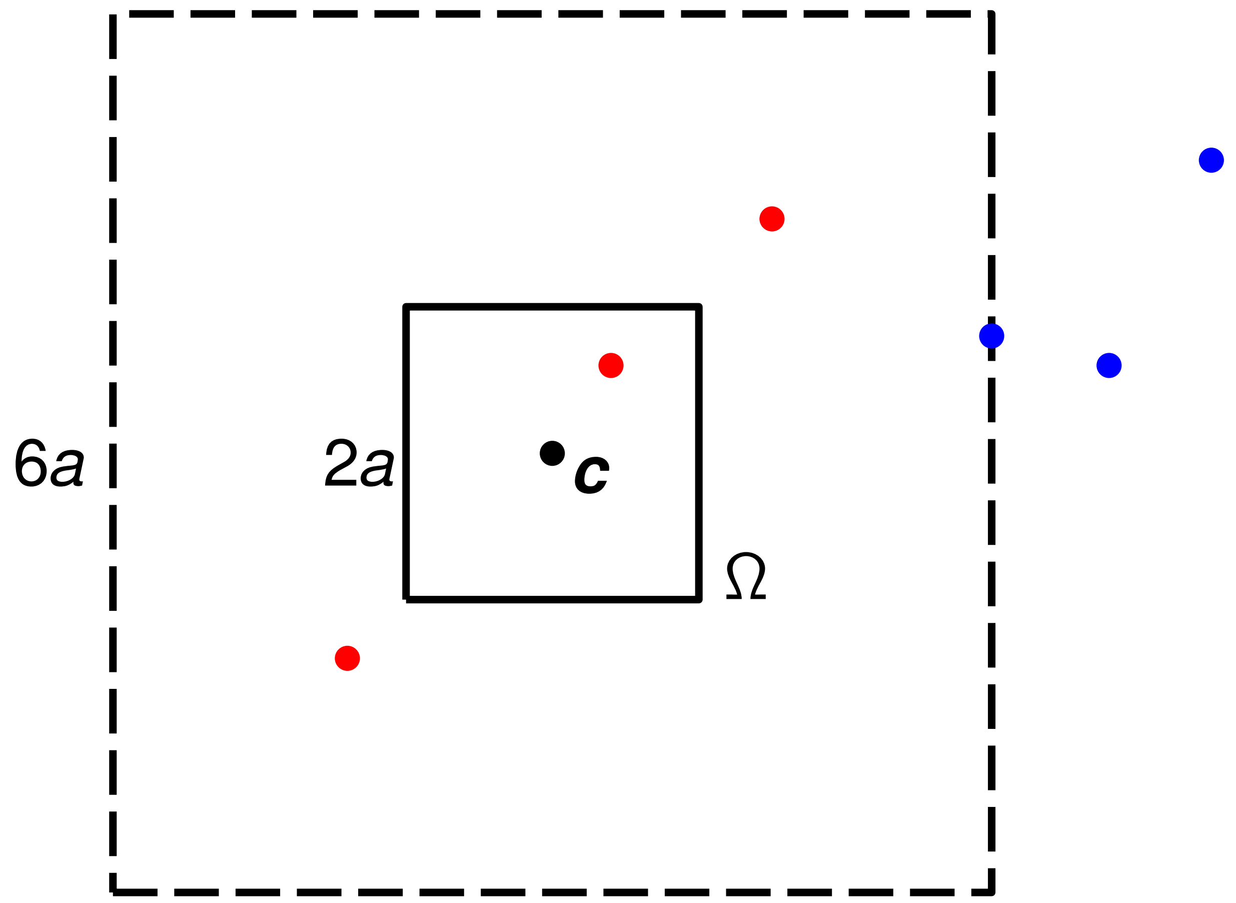
Evaluate $\hat{q}_p = -\frac{1}{p}\sum_{j=1}^{N}(\boldsymbol{y}_j - \boldsymbol{c})^p\, q_j$ for $p = 0,\, 1,\, \dots,\, P$ — *cost is $O(N\,P)$.*

Evaluate $u_i = \log(\boldsymbol{x}_i - \boldsymbol{c})\,\hat{q}_0 + \sum_{p=1}^{P}\frac{1}{(\boldsymbol{x}_i - \boldsymbol{c})^p}\,\hat{q}_p$ — *cost is $O(M\,P)$.*

The cost has been reduced from $O(MN)$ to $O(P(M+N))$.

Definition: Let Ω be a square with center $\boldsymbol{c} = (c_1, c_2)$ and side length $2a$. Then we say that a point $\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2$ is *well-separated* from Ω if

$$\max(|x_1 - c_1|, |x_2 - c_2|) \geq 3a.$$



Any point on or outside of the dashed square is well-separated from Ω.

Blue points are well-separated. Red points are not.

Definition:

Let Ω be a square with center $\boldsymbol{c}$ containing sources $\{q_j\}_{j=1}^n$ at locations $\{\boldsymbol{y}_j\}_{j=1}^n$.

The *outgoing expansion* of Ω (to order P) is the vector $\hat{\mathbf{q}} \in \mathbb{C}^{P+1}$ with numbers

$$\hat{q}_0 = \sum_{j=1}^{n} q_j,$$

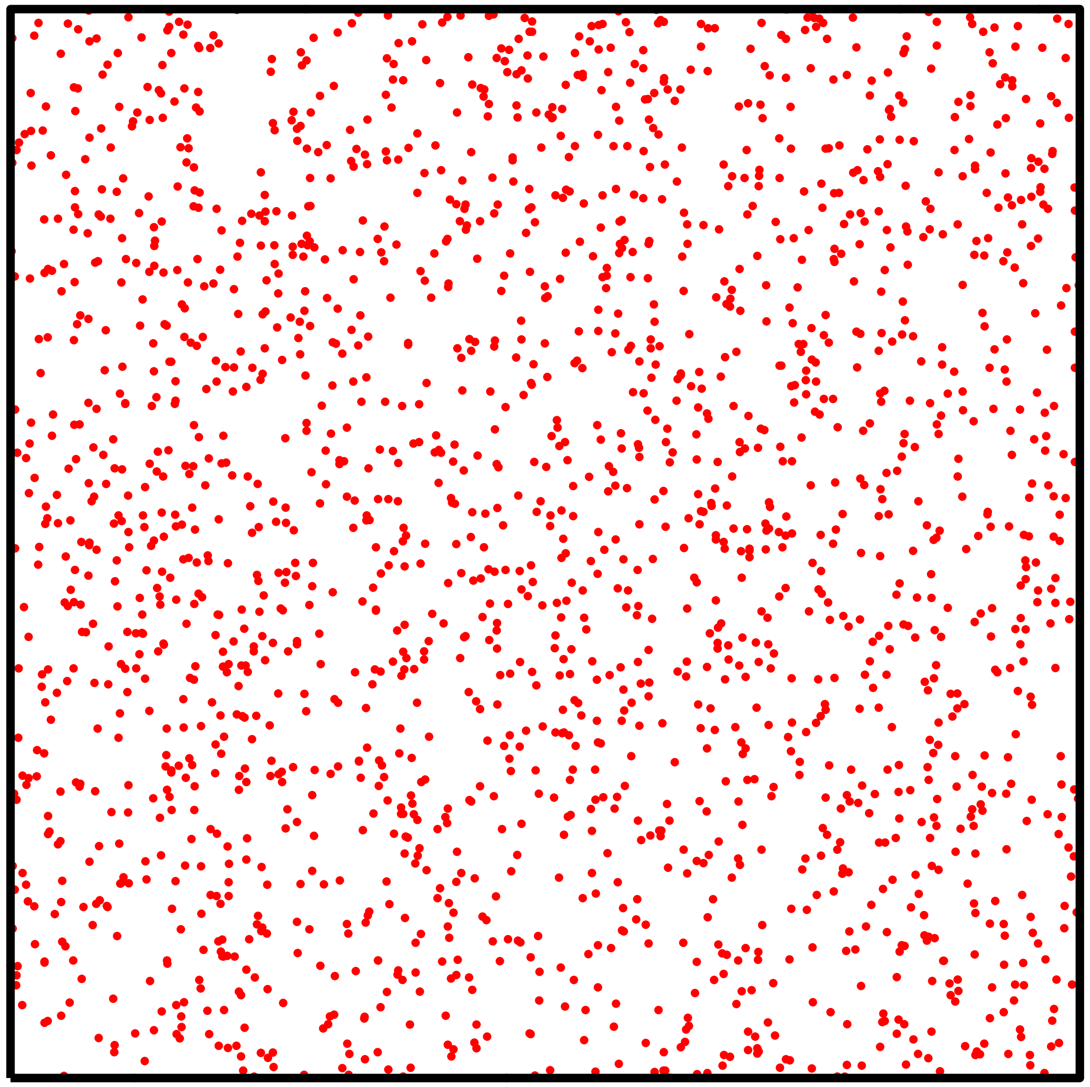
$$\hat{q}_p = -\frac{1}{p}\sum_{j=1}^{n} (\boldsymbol{y}_j - \boldsymbol{c})^j\, q_j, \qquad p = 1, 2, 3, \dots, P.$$

The outgoing expansion compactly encodes source strengths and source locations.

It allows you to evaluate the potential u caused by the sources in Ω (to precision that depends on P and the distance to the sources).

"outgoing expansion" = "multipole expansion"

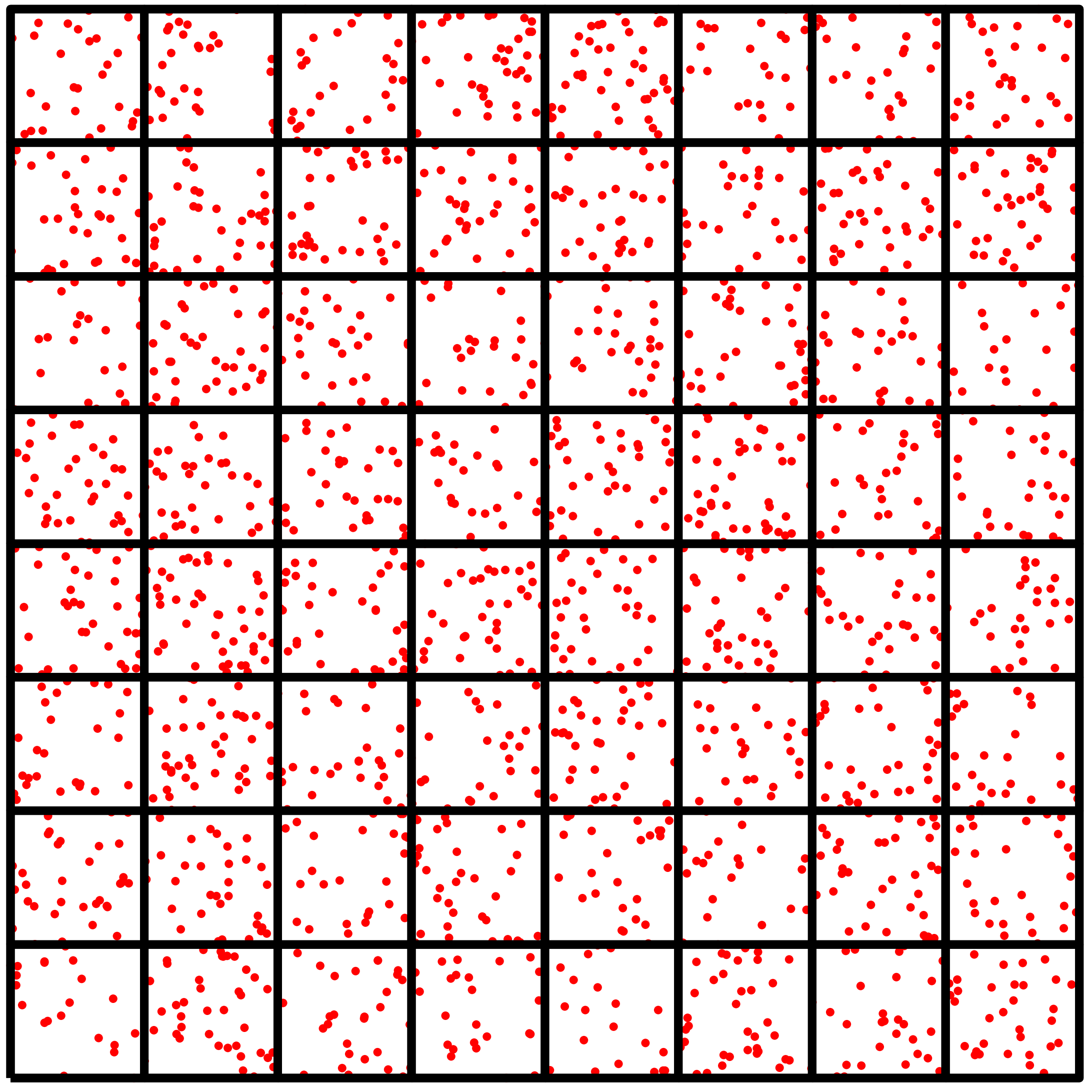
Single-level Barnes-Hut



We seek to evaluate all pairwise interactions between N particles in a box;

for now, assume that the particle locations $\{\boldsymbol{x}_i\}_{i=1}^N$ are fairly evenly distributed.

Single-level Barnes-Hut



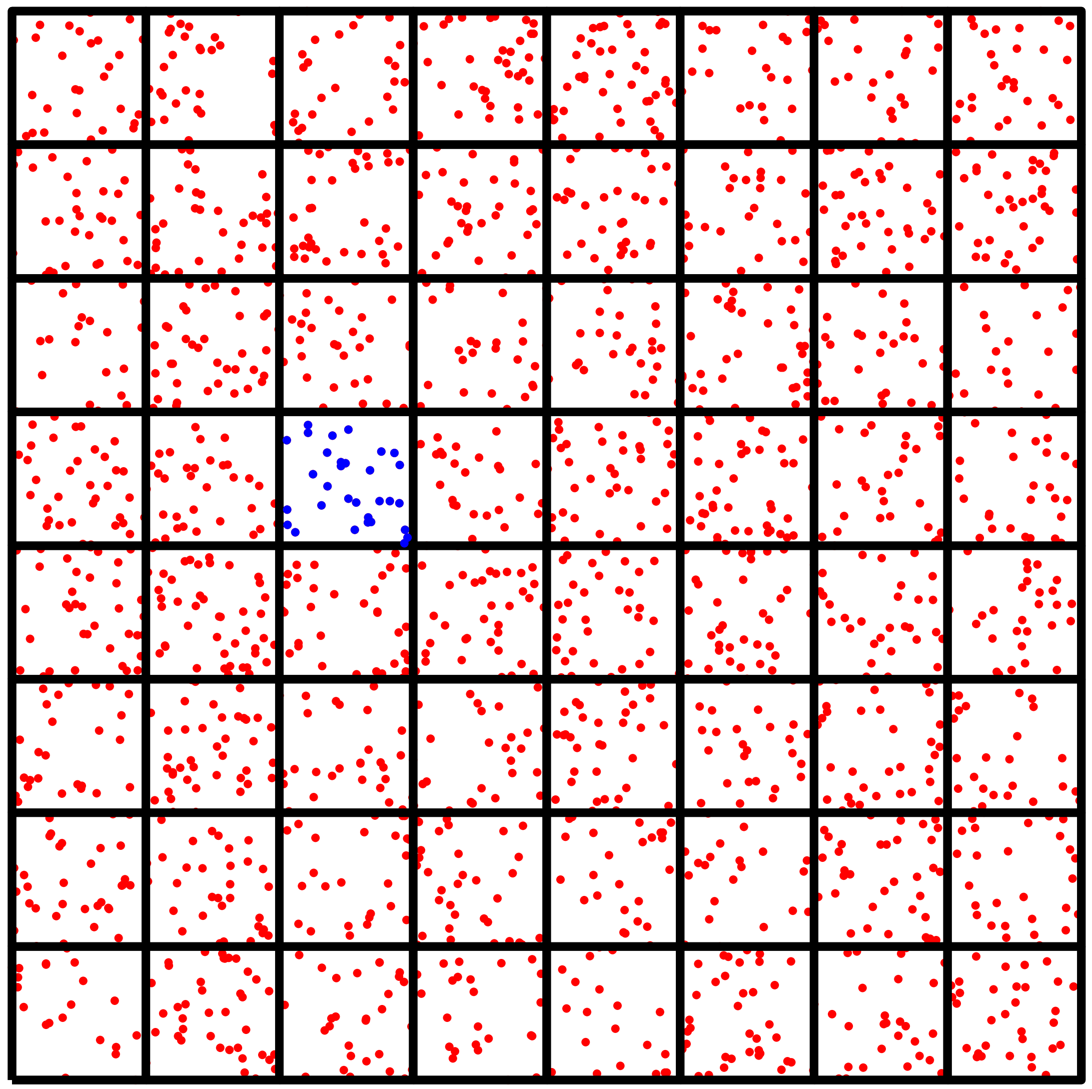
Place a square grid of boxes on top of the computational box.

Assume each box holds about m particles (so there are about N/m boxes).

Given a tolerance ε, pick P so that, roughly, $(\sqrt{2}/3)^P < \varepsilon$ (... details left out ...).

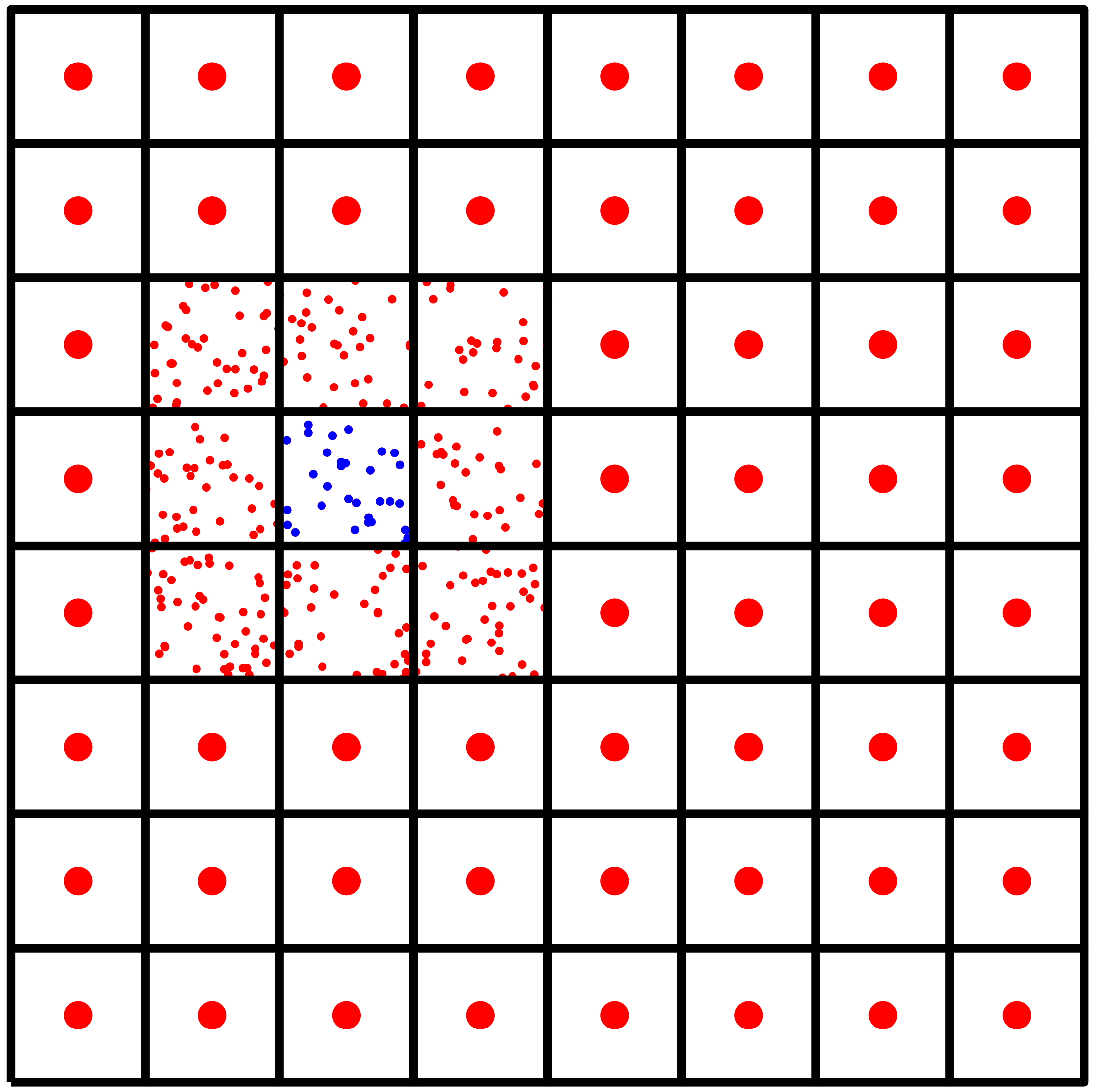
For each box, compute its outgoing expansion.

Single-level Barnes-Hut



How do you evaluate the potentials at the blue locations?

Single-level Barnes-Hut



How do you evaluate the potentials at the blue locations?

Directly evaluate interactions with the particles close by.

For all long-distance interactions, use the outgoing expansion!

Cost of the single-level Barnes-Hut algorithm:

Let m denote the number of particles in a box.

For each particle, we need to do the following:

Step 1:	Evaluate the P outgoing moments:	P
Step 2:	Evaluate potentials from outgoing expansions:	$((N/m) - 9)\,P$
Step 3:	Evaluate potentials from close particles:	$9\,m$

Using that P is a smallish constant we find

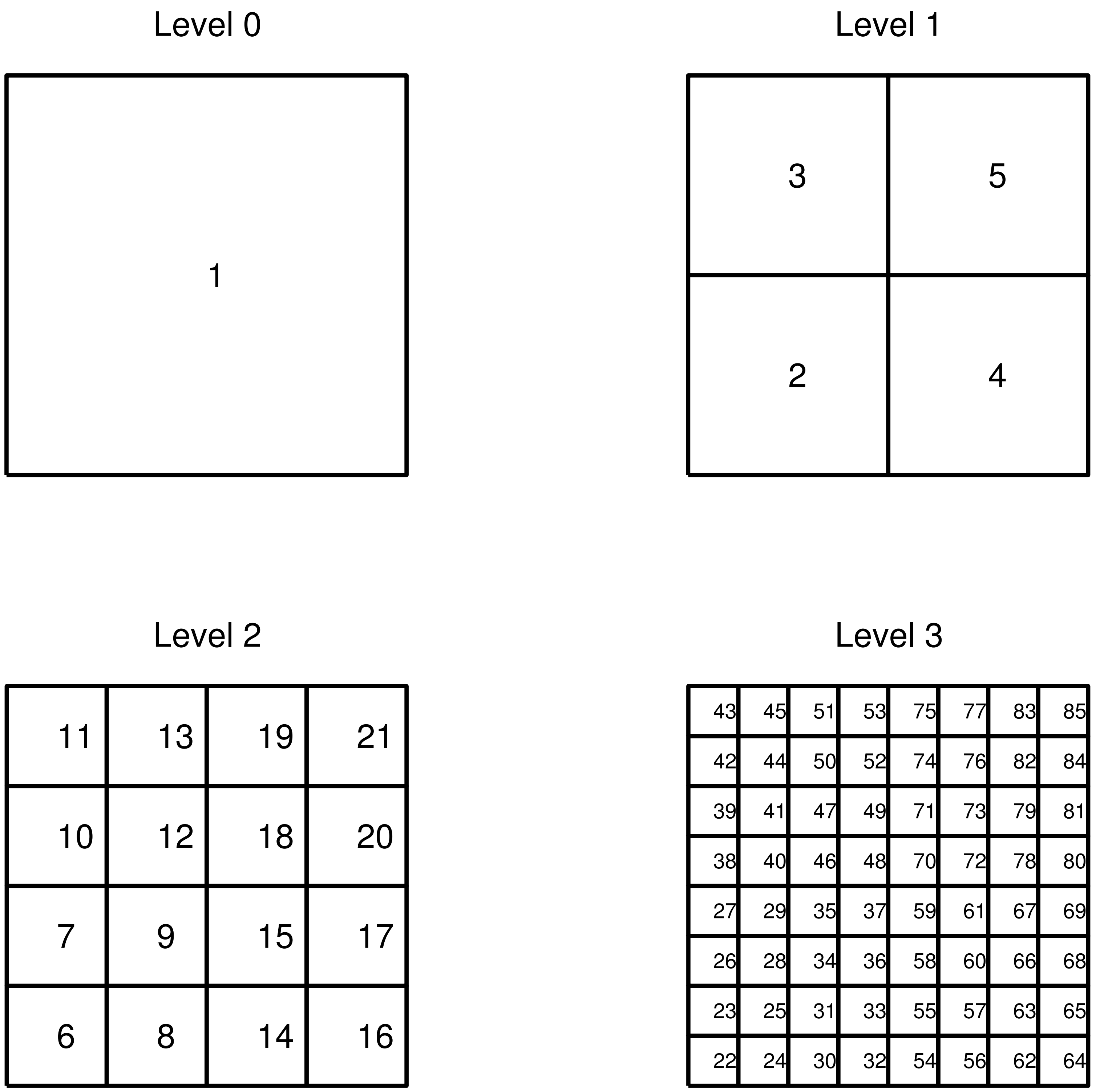
$$\text{cost} \sim \frac{N^2}{m} + N\,m.$$

Set $m \sim N^{1/2}$ to obtain:

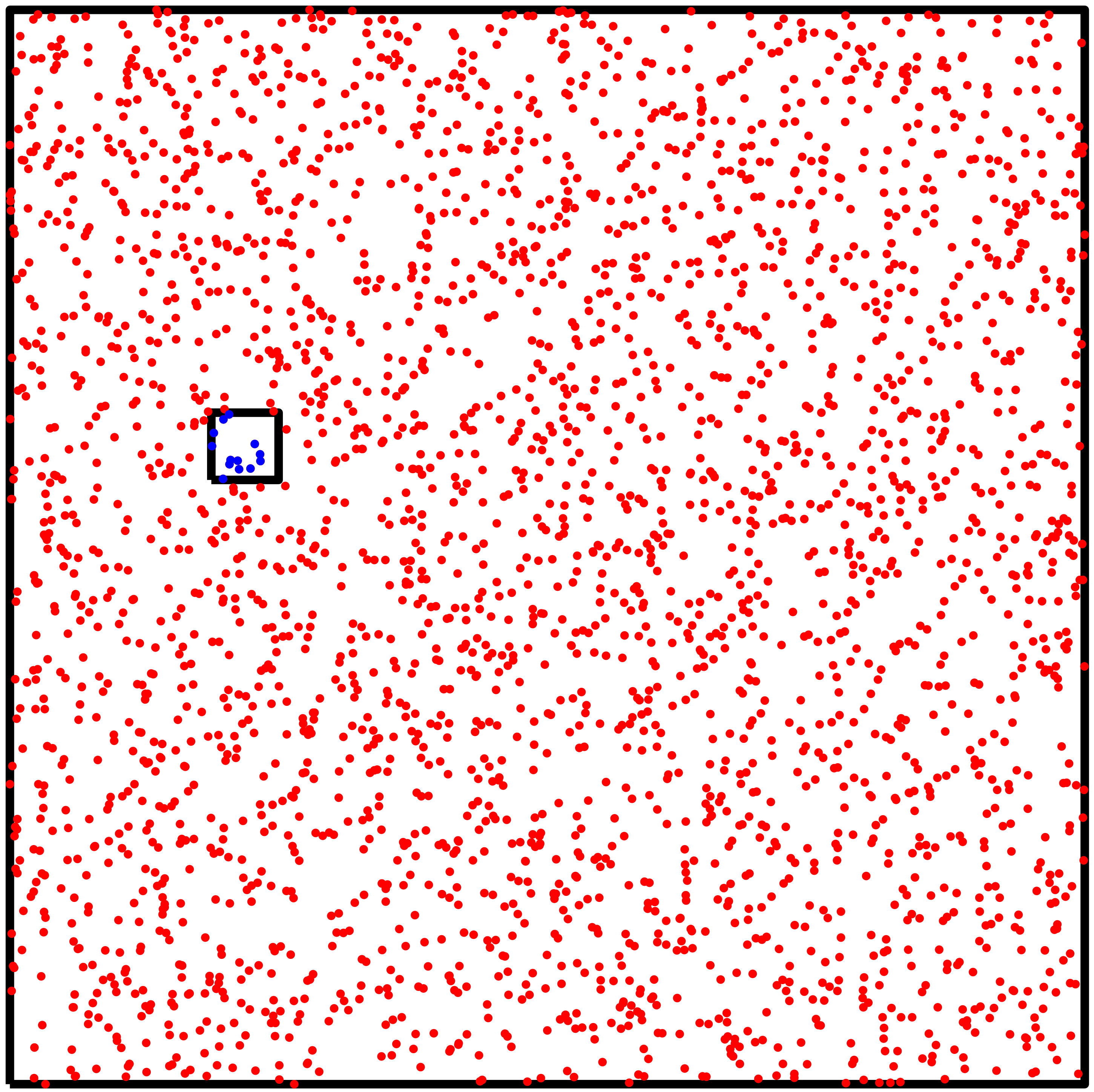
$$\text{cost} \sim N^{1.5}.$$

We're doing better than $O(N^2)$ but still not great.

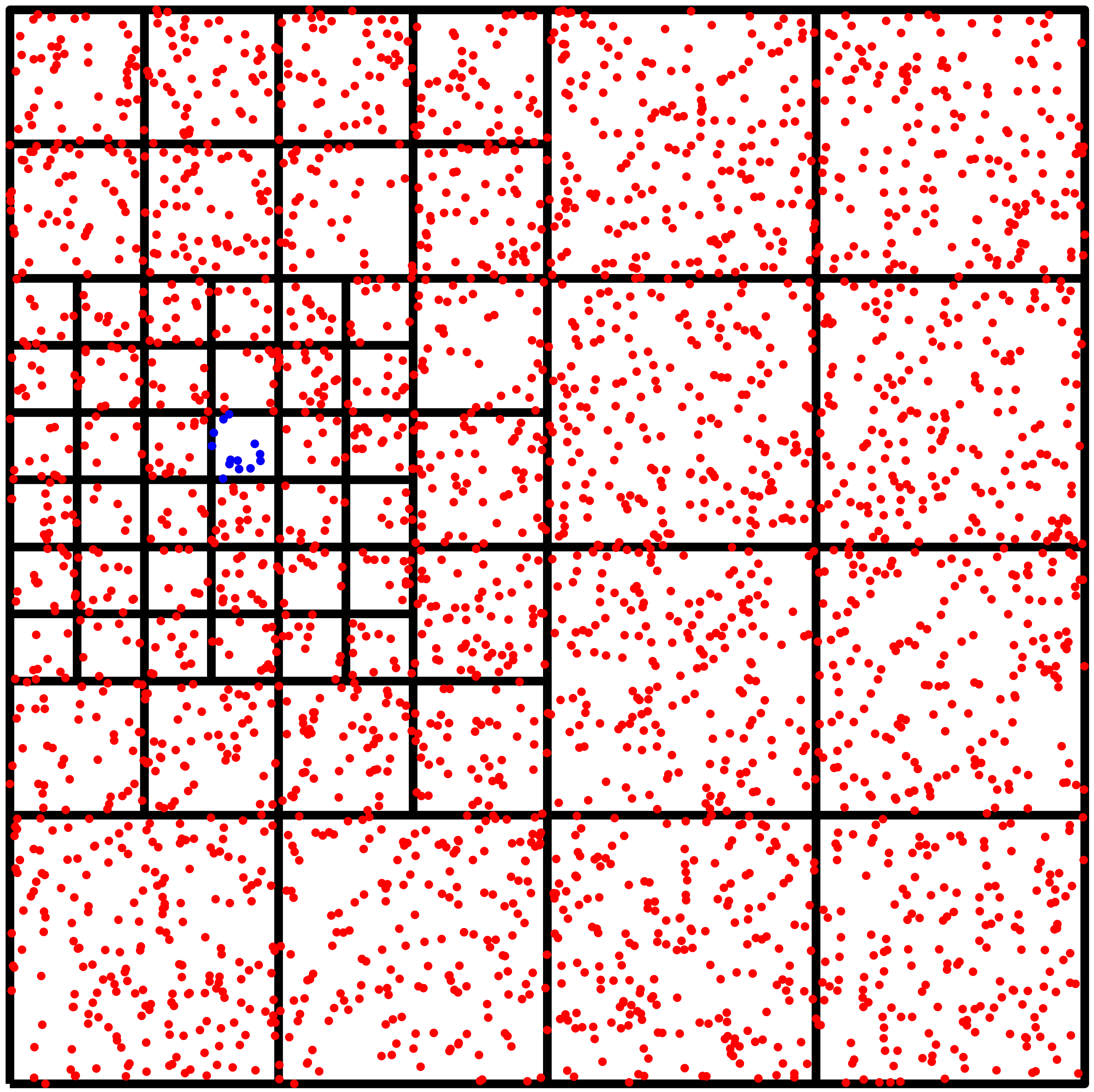
To get the asymptotic cost down further, we need a *hierarchy of boxes* (or a “tree of boxes”) on the computational domain:



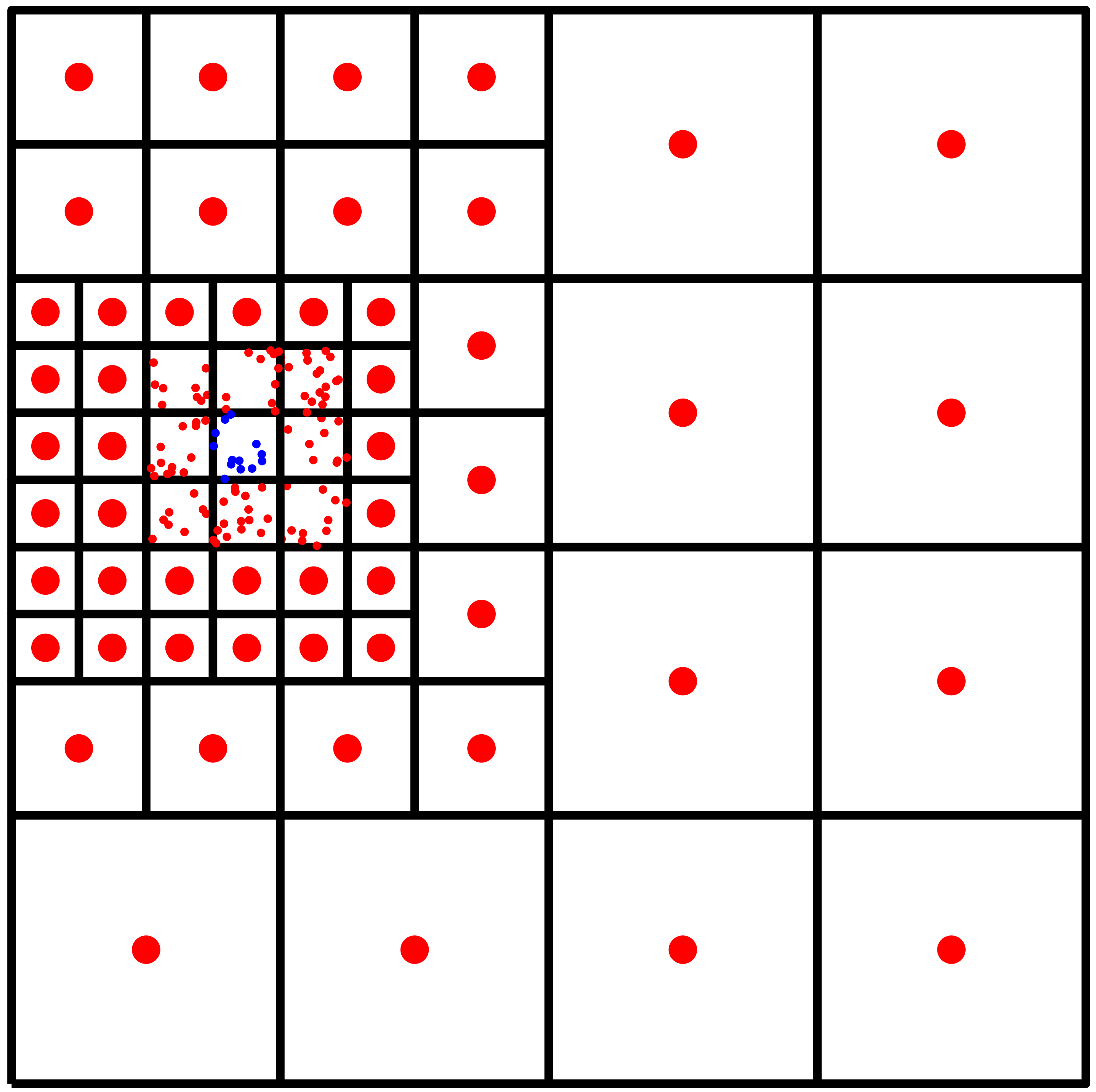
For each box in the tree, compute its outgoing expansion.



How do you find the potential at the locations marked in blue?



Tessellate the remainder of the domain using as large boxes as you can, with the constraint that *the target box has to be well-separated from every box that is used.*



Then replace the original sources in each well-separated box by the corresponding multipole expansion.

The work required to evaluate one potential is now $O(\log N)$.

Cost of the multi-level Barnes-Hut algorithm:

Suppose there are L levels in the tree, and that there are about m particles in each box so that $L \approx \log_4(N/m)$.

We let m be a fixed number (say $m \approx 200$) so $L \sim \log(N)$.

Observation: On each level, there are at most 27 well-separated boxes.

For each particle $\boldsymbol{x}_i$, we need to do the following:

Step 1:	Evaluate the P outgoing moments for the L boxes holding $\boldsymbol{x}_i$:	$L\,P$
Step 2:	Evaluate potentials from outgoing expansions:	$27\,L\,P$
Step 3:	Evaluate potentials from neighbors:	$9\,m$

Using that P is a smallish constant (say $P = 20$) we find

$$\text{cost} \sim N\,L \sim N\,\log(N).$$

This is not bad.

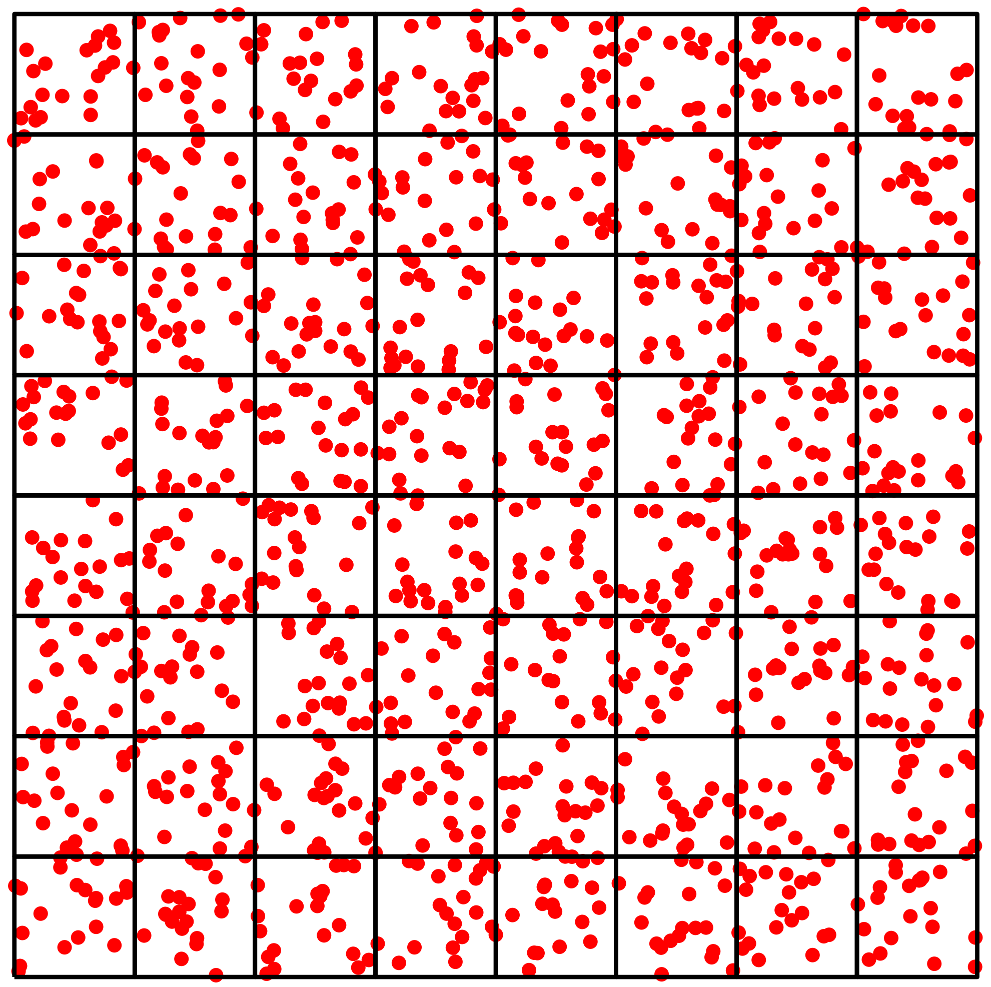
The Barnes-Hut algorithm has asymptotic cost $O(N \log N)$ (at constant precision).

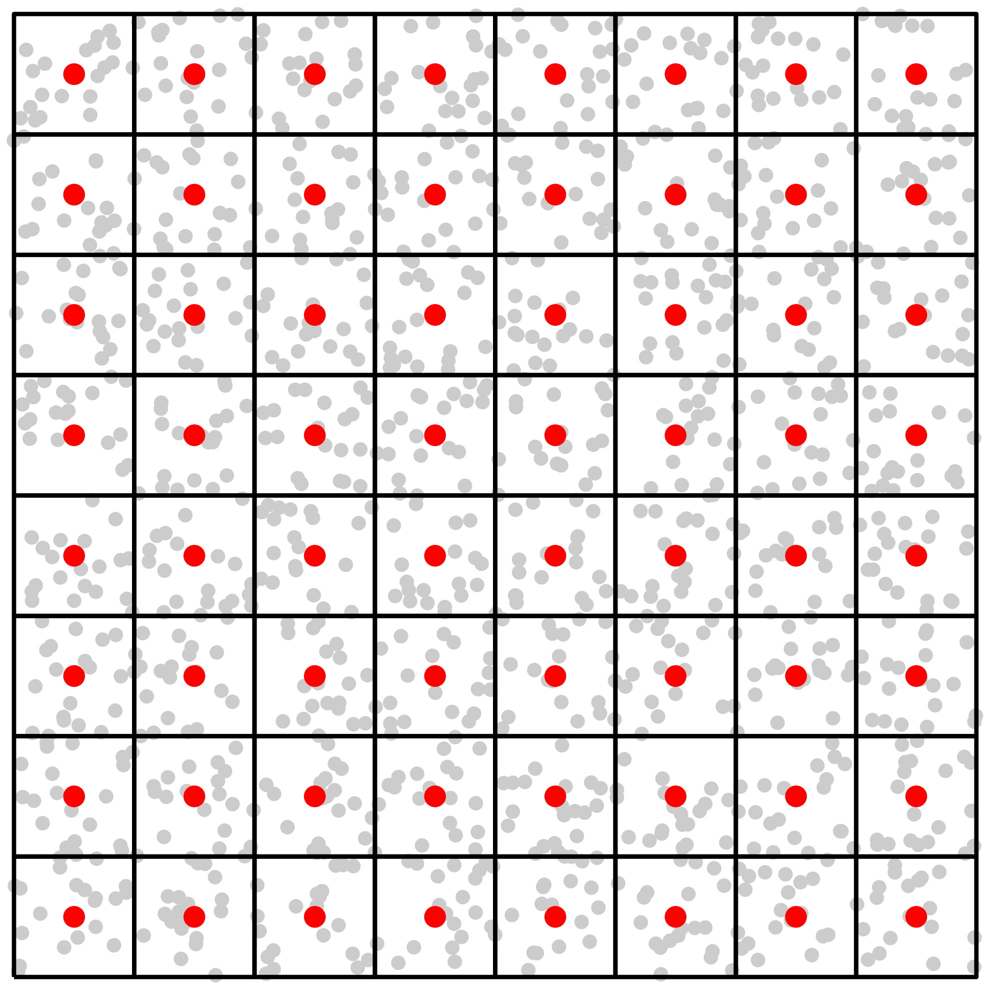
We will next improve the accuracy to the optimal $O(N)$.

Recall: We partition into $L \sim \log N$ levels. Then the costs are:

	Asymptotic cost	*How get to $O(N)$?*
Stage 1 — compute all outgoing expansions: Each particle communicates with all L boxes that contain it.	$O(N \log N)$	Easy!
Stage 2 — compute all far field potentials: Each particle gathers contributions from $\sim 27L$ outgoing expansions.	$O(N \log N)$	???
Stage 3 — compute all near field potentials: Each particle gathers contributions directly from all particles in the same box, and from all particles in the 8 immediate neighbors.	$O(N)$	No need to fix!

Reducing the cost of computing all outgoing expansions from $O(N \log N)$ to $O(N)$:



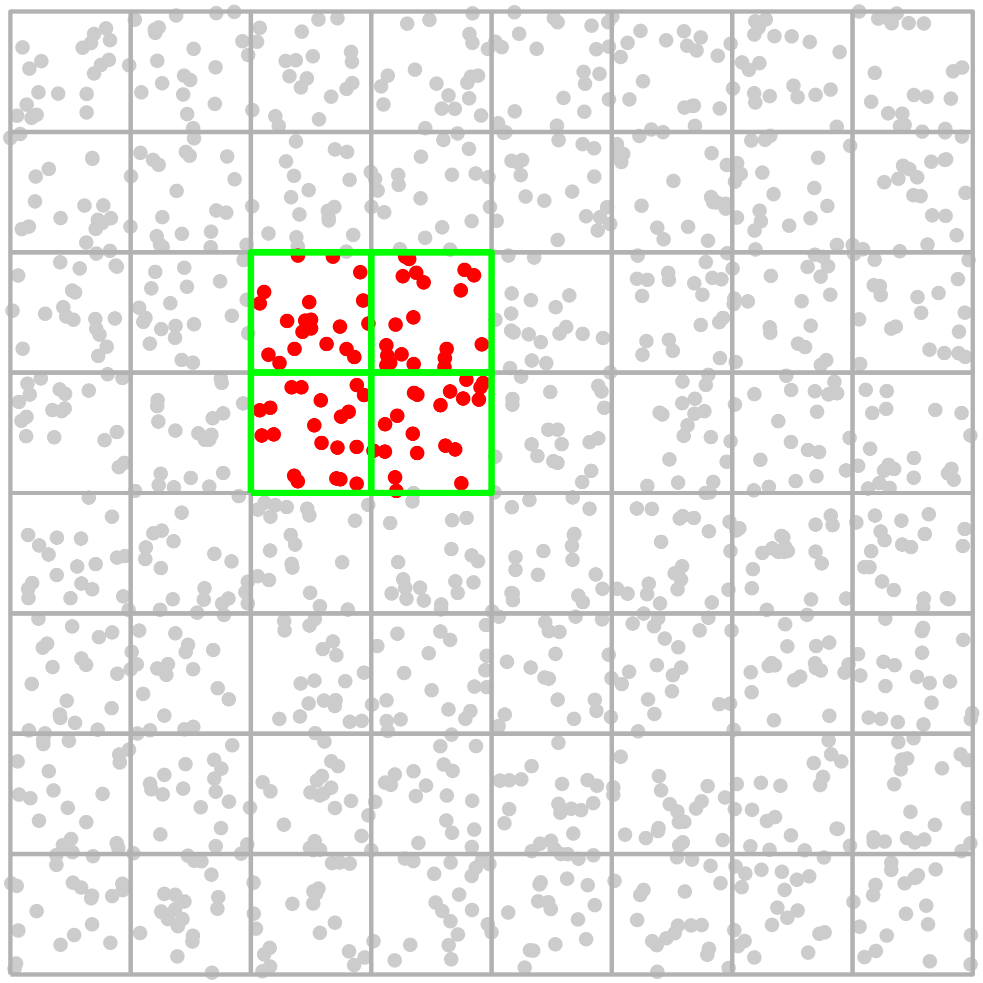
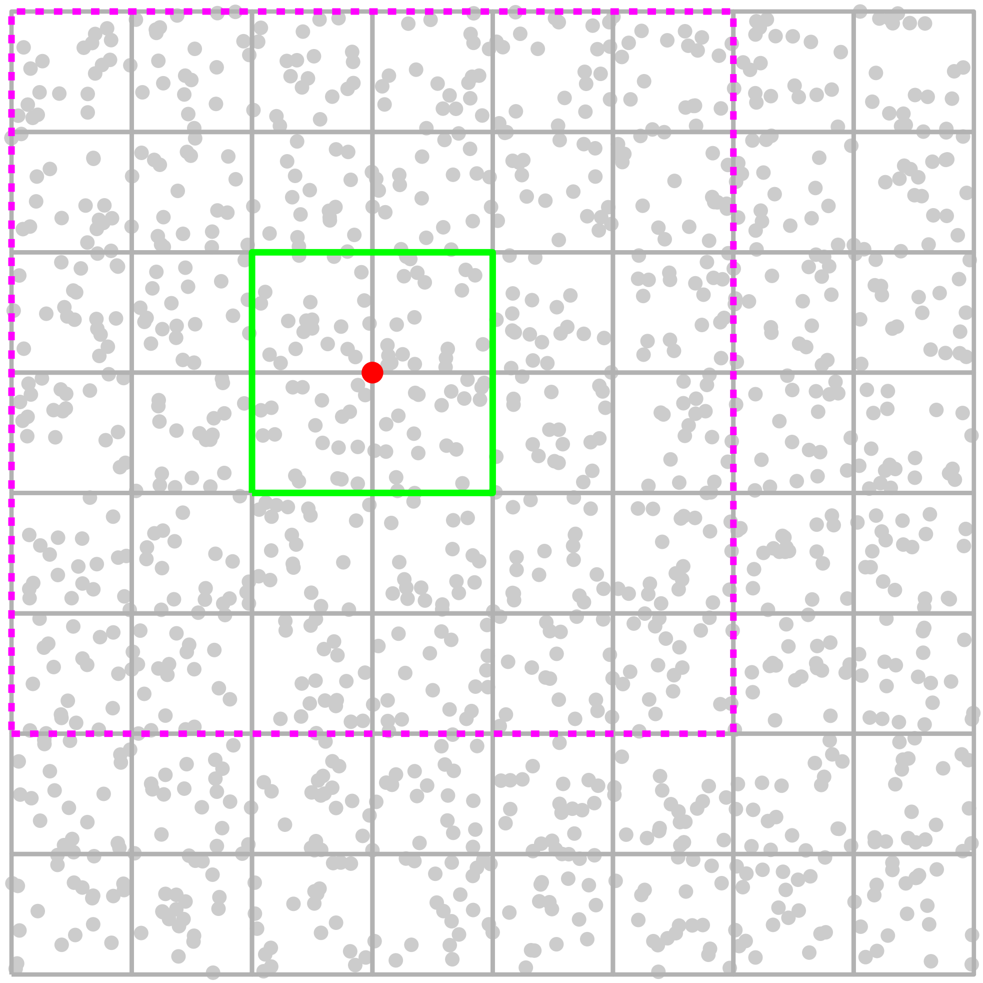


For every leaf box τ, we directly compute the outgoing expansion from the source vector

$$\hat{\mathbf{q}}_\tau = \mathbf{C}_\tau\, \mathbf{q}(J_\tau).$$

(Just as before.)

Reducing the cost of computing all out-going expansions from $O(N \log N)$ to $O(N)$:

Now consider a box Ω_τ made up of four leaves: $\Omega_\tau = \Omega_{\sigma_1} \cup \Omega_{\sigma_2} \cup \Omega_{\sigma_3} \cup \Omega_{\sigma_4}$

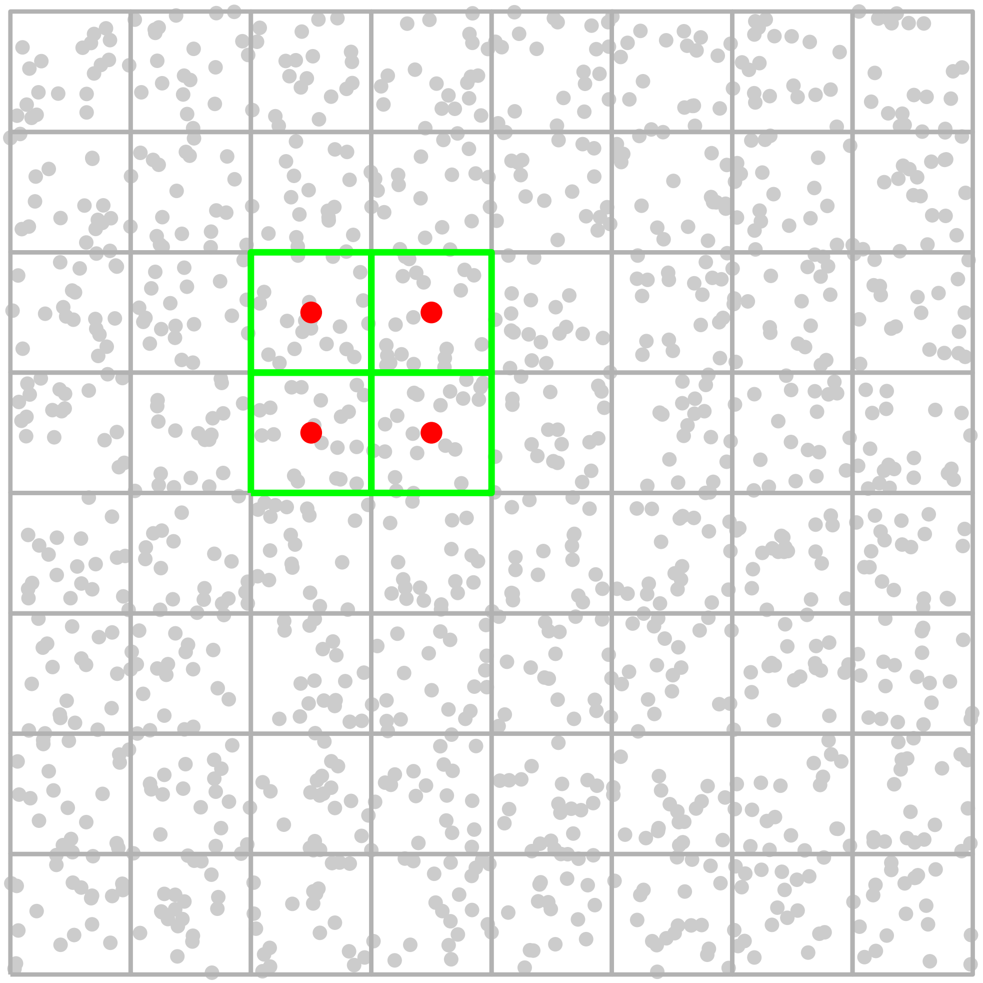
We seek an outgoing expansion that is valid outside the dotted magenta line.

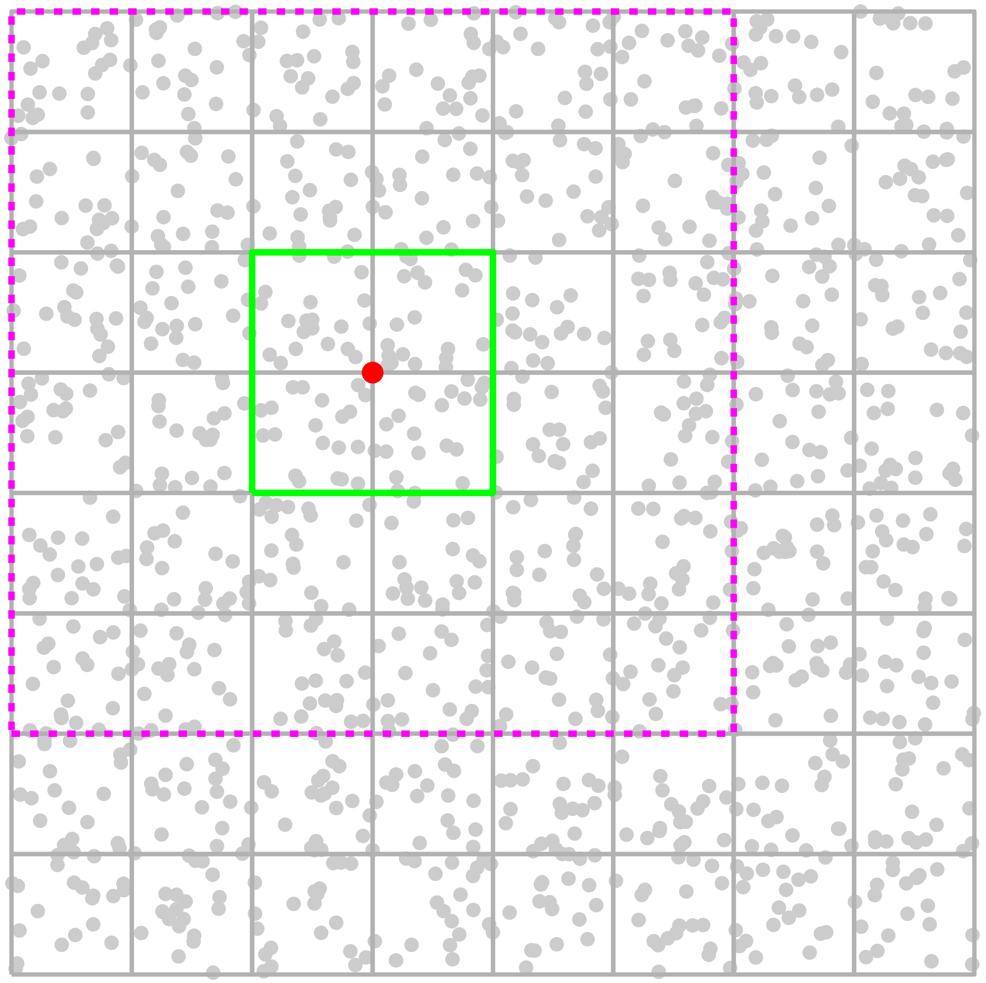
In this region, the outgoing expansions of the children $\{\sigma_1,\, \sigma_2,\, \sigma_3,\, \sigma_4\}$ are valid.

"Move" these expansions via a so called *outgoing-from-outgoing translation operator:*

$$\hat{\mathbf{q}}_\tau = \sum_{j=1}^{4} \mathbf{T}^{(\mathrm{ofo})}_{\tau,\sigma_j}\, \hat{\mathbf{q}}_{\sigma_j}.$$

Reducing the cost of computing all out-going expansions from $O(N \log N)$ to $O(N)$:





Now consider a box Ω_τ made up of four leaves: $\Omega_\tau = \Omega_{\sigma_1} \cup \Omega_{\sigma_2} \cup \Omega_{\sigma_3} \cup \Omega_{\sigma_4}$

We seek an outgoing expansion that is valid outside the dotted magenta line.

In this region, the outgoing expansions of the children $\{\sigma_1,\, \sigma_2,\, \sigma_3,\, \sigma_4\}$ are valid.

"Move" these expansions via a so called *outgoing-from-outgoing translation operator:*

$$\hat{\mathbf{q}}_\tau = \sum_{j=1}^{4} \mathbf{T}^{(\mathrm{ofo})}_{\tau,\sigma_j}\, \hat{\mathbf{q}}_{\sigma_j}.$$

Cost of constructing all outgoing expansions:

(Recall: $L \sim \log N$ is the number of levels. P is the length of the expansion.)

Level L (the leaves)	"outgoing from sources"	NP
Level $L-1$ (next coarser)	"outgoing from outgoing"	$(N_{\rm boxes})\,P^2$
Level $L-2$	"outgoing from outgoing"	$(N_{\rm boxes}/4)\,P^2$
Level $L-3$	"outgoing from outgoing"	$(N_{\rm boxes}/16)\,P^2$
Level $L-4$	"outgoing from outgoing"	$(N_{\rm boxes}/64)\,P^2$
$\vdots$	$\vdots$	$\vdots$
Total:		$NP + N_{\rm boxes}\,P^2$

We succeeded in attaining $O(N)$ cost for computing all outgoing expansions.

Next we will device an $O(N)$ scheme for computing so called *incoming expansions.*

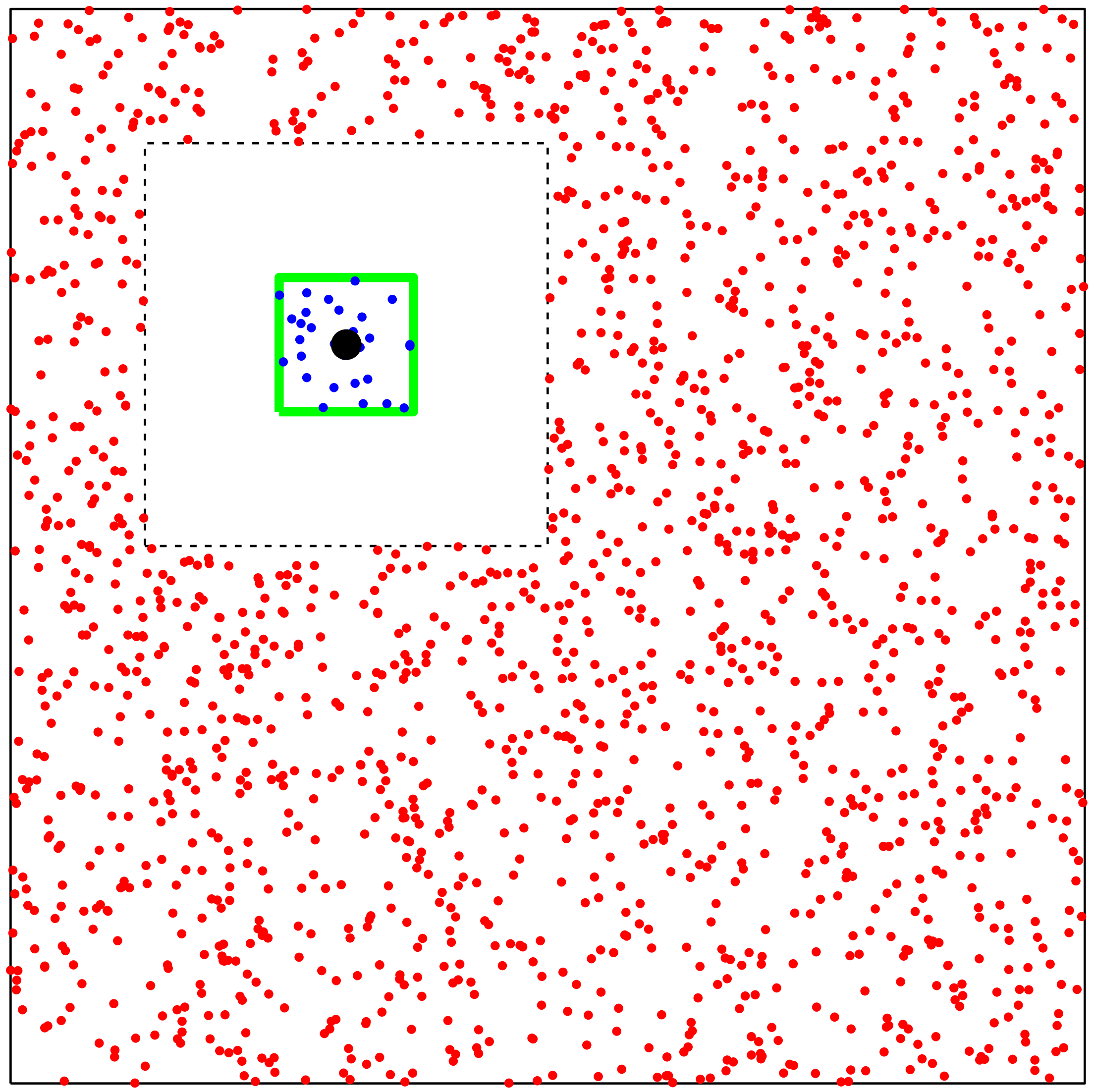
The incoming expansion

Let Ω_τ be a box (green).

Let $\boldsymbol{c}_\tau$ be the center of τ (black).

Let $I_\tau^{(\mathrm{far})}$ denote a list of all sources well-separated from τ (red), and let ϕ denote the potential

$$\phi(\boldsymbol{x}) = \sum_{j \in I_\tau^{(\mathrm{far})}} \log(\boldsymbol{x}_i - \boldsymbol{y}_j)\, q_j, \qquad \boldsymbol{x} \in \Omega_\tau.$$



The *incoming expansion* of τ is a vector $\hat{\mathbf{u}} = [\hat{u}_p]_{p=0}^P$ of complex numbers such that

$$\phi(\boldsymbol{x}) \approx \sum_{p=0}^{P} \hat{u}_p (\boldsymbol{x} - \boldsymbol{c}_\tau)^p, \qquad \boldsymbol{x} \in \Omega_\tau. \tag{3}$$

The incoming expansion is a compact representation of the field generated by sources that are well-separated from Ω_τ (it encodes both the source locations and magnitudes).

Recall:

$$\phi(\boldsymbol{x}) \approx \sum_{p=0}^{P} \hat{u}_p(\boldsymbol{x} - \boldsymbol{c}_\tau)^p, \qquad \boldsymbol{x} \in \Omega_\tau.$$

Recall that each $\hat{u}_p = \hat{u}_p^{\mathrm{r}} + i\,\hat{u}_p^{\mathrm{i}}$ is a complex number.

Taking real parts, and using polar coordinates,

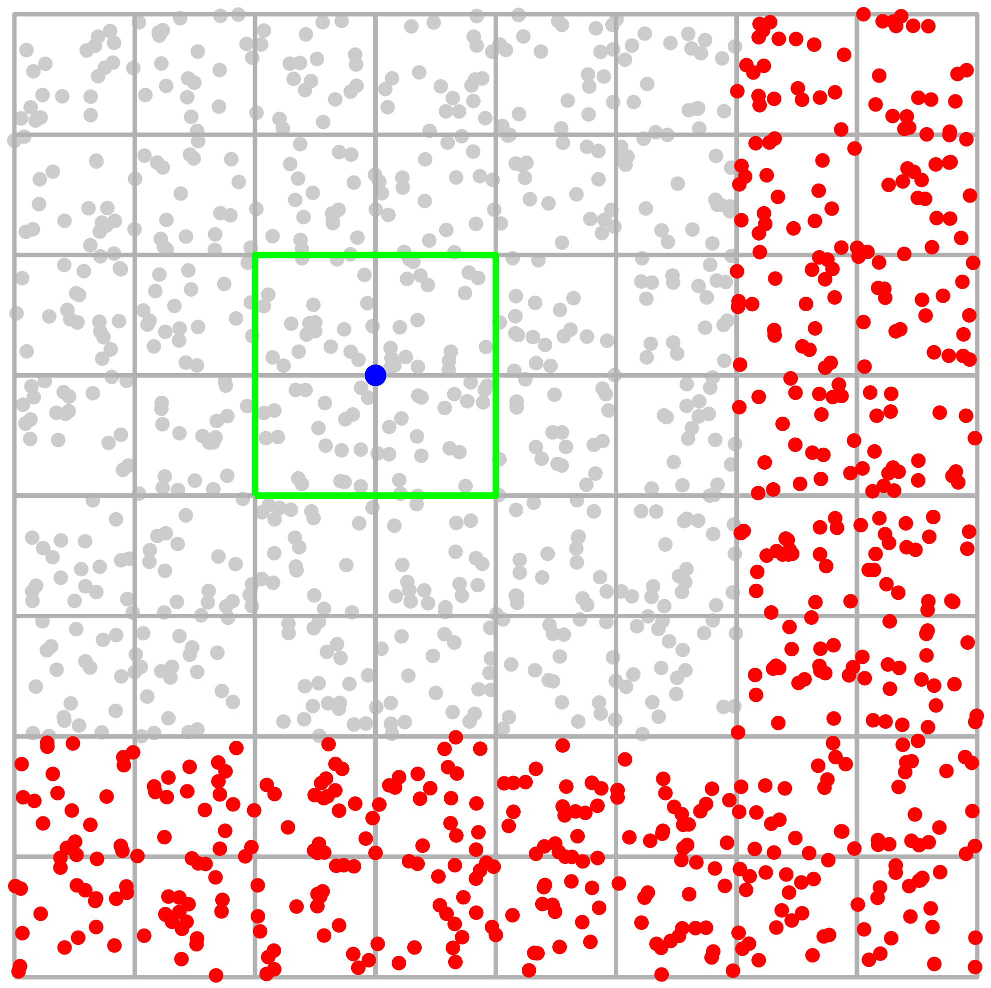
$$\boldsymbol{x} - \boldsymbol{c}_\tau = r\,e^{i\theta},$$

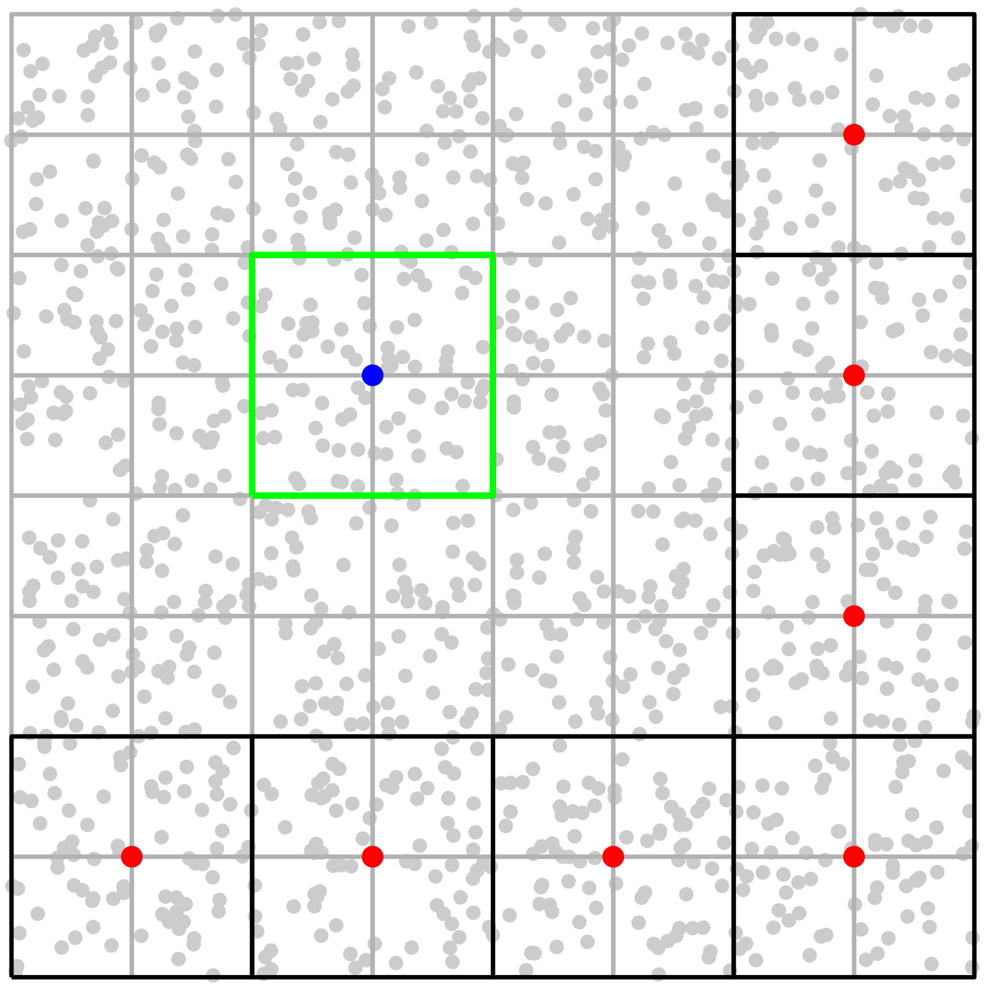
we get

$$\begin{aligned}\text{Real}(\phi(\boldsymbol{x})) =& \hat{u}_0^{\mathrm{r}} + \\ & \hat{u}_1^{\mathrm{r}}\, r^1 \cos(1\theta) - \hat{u}_1^{\mathrm{i}}\, r^1 \sin(1\theta) + \cdots \\ & \hat{u}_2^{\mathrm{r}}\, r^2 \cos(2\theta) - \hat{u}_2^{\mathrm{i}}\, r^2 \sin(2\theta) + \cdots \\ & \hat{u}_3^{\mathrm{r}}\, r^3 \cos(3\theta) - \hat{u}_3^{\mathrm{i}}\, r^3 \sin(3\theta) + \cdots\end{aligned}$$

Classical expansion in harmonics.

Computing the incoming expansions for all boxes in $O(N)$ operations





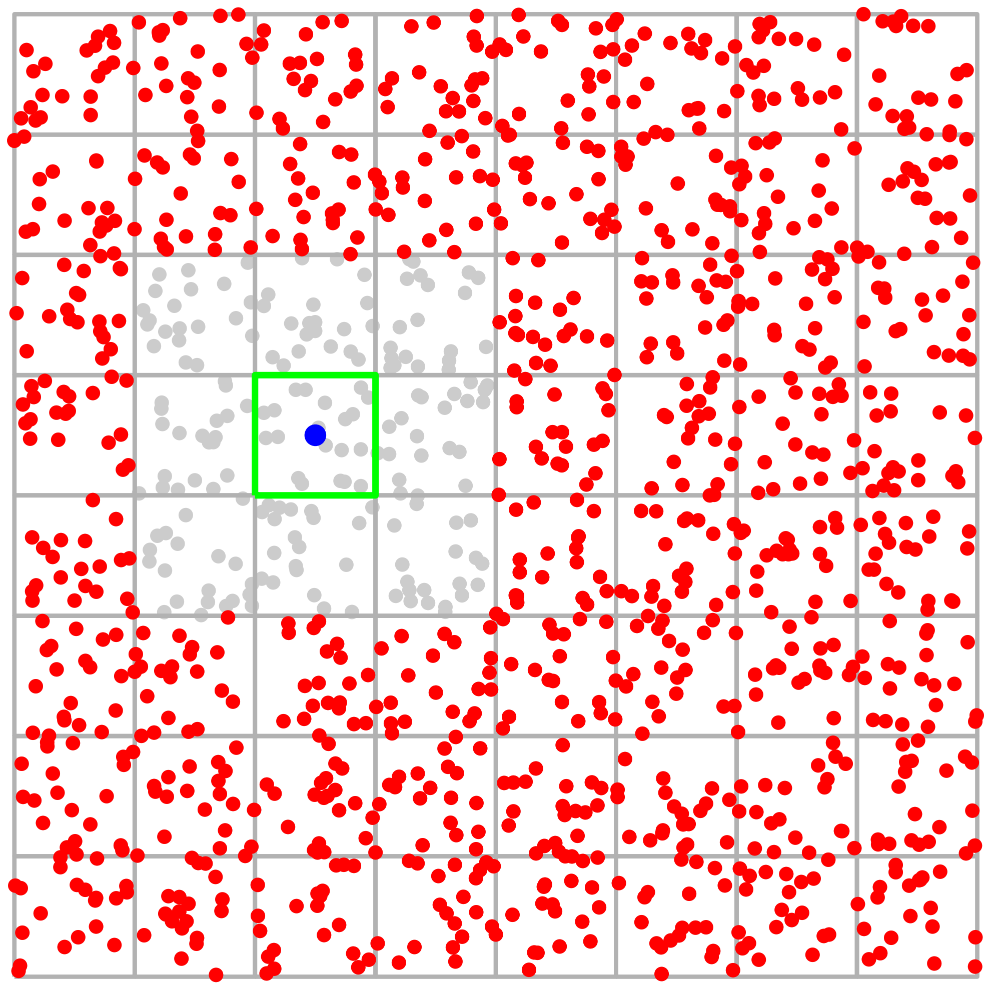
We seek to construct the incoming expansion for box τ (marked in green).

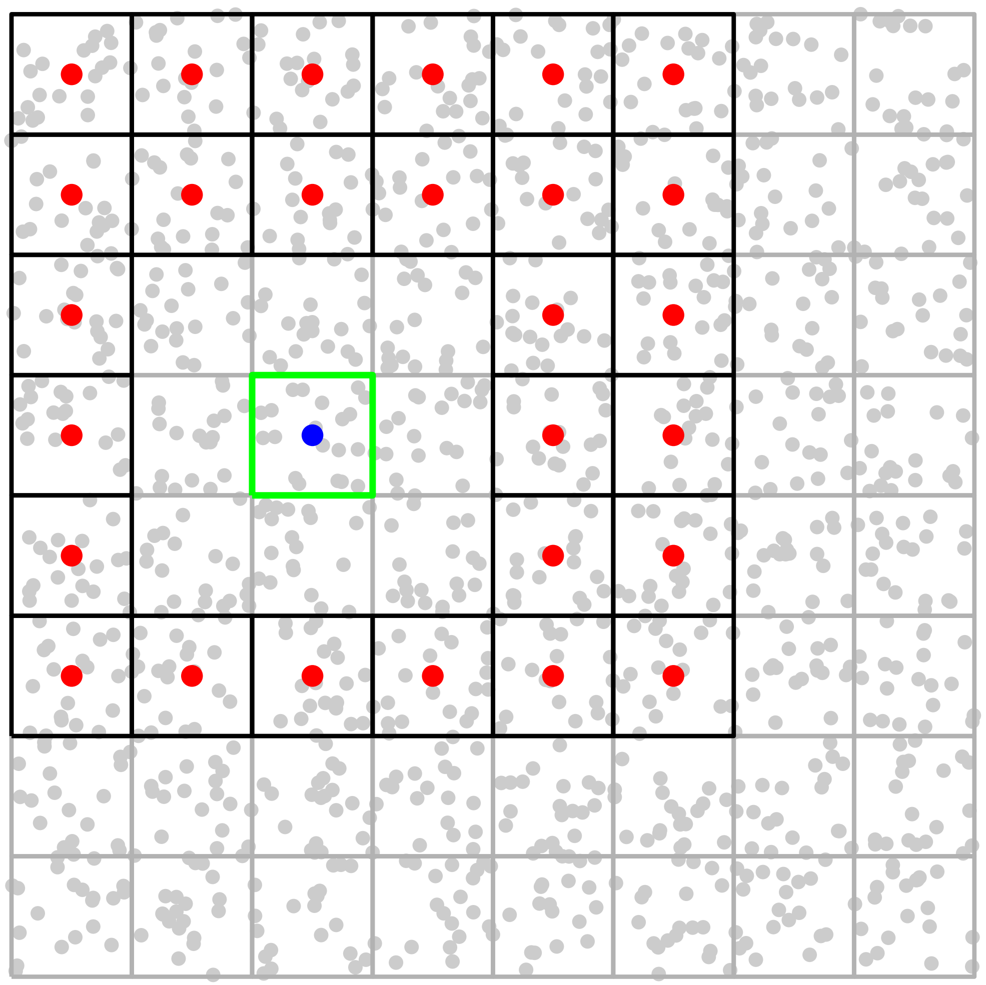
We use the outgoing expansions for all well-separated boxes:

$$\hat{\mathbf{u}}_\tau = \sum_{\sigma \in \mathcal{L}_\tau^{(\mathrm{int})}} \mathbf{T}_{\tau,\sigma}^{(\mathrm{ifo})}\,\hat{\mathbf{q}}_\sigma$$

where $\mathbf{T}_{\tau,\sigma}$ is the *incoming-from-outgoing* translation operator, and $\mathcal{L}_\tau^{(\mathrm{int})}$ is the *interaction list* of box τ.

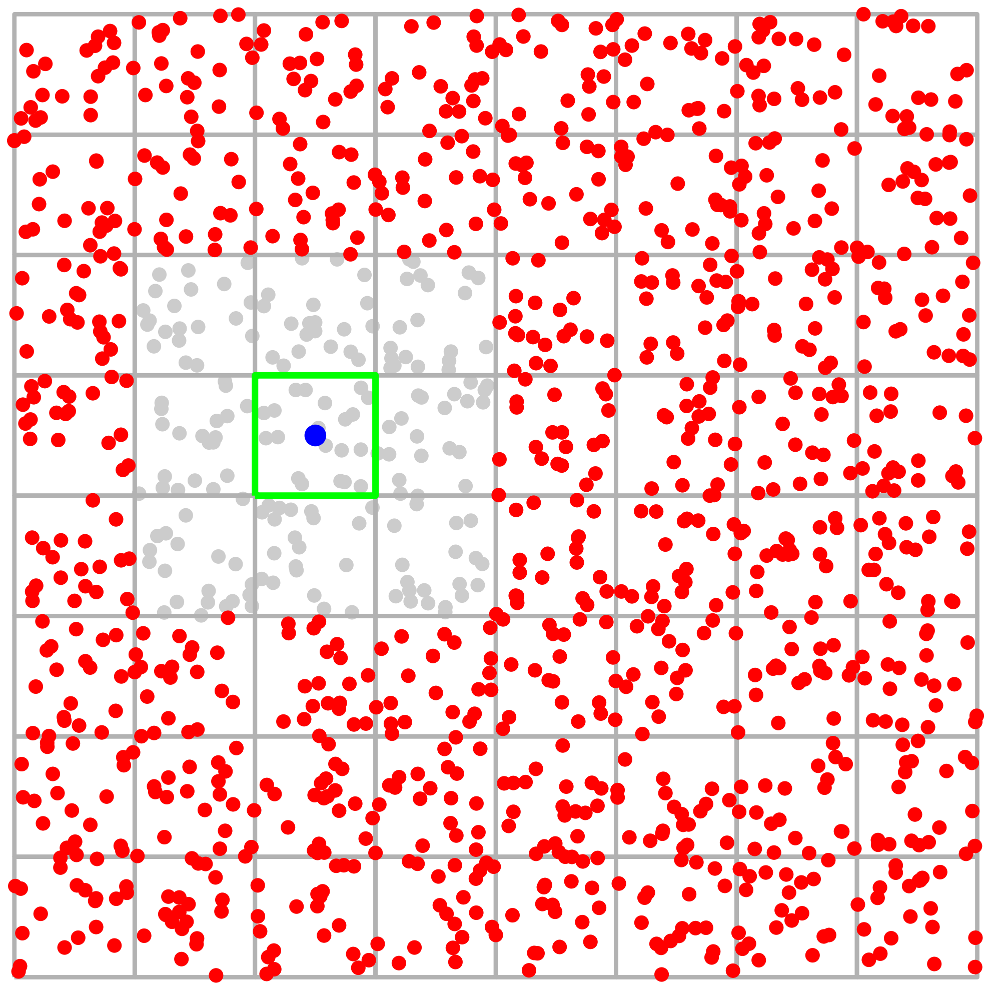
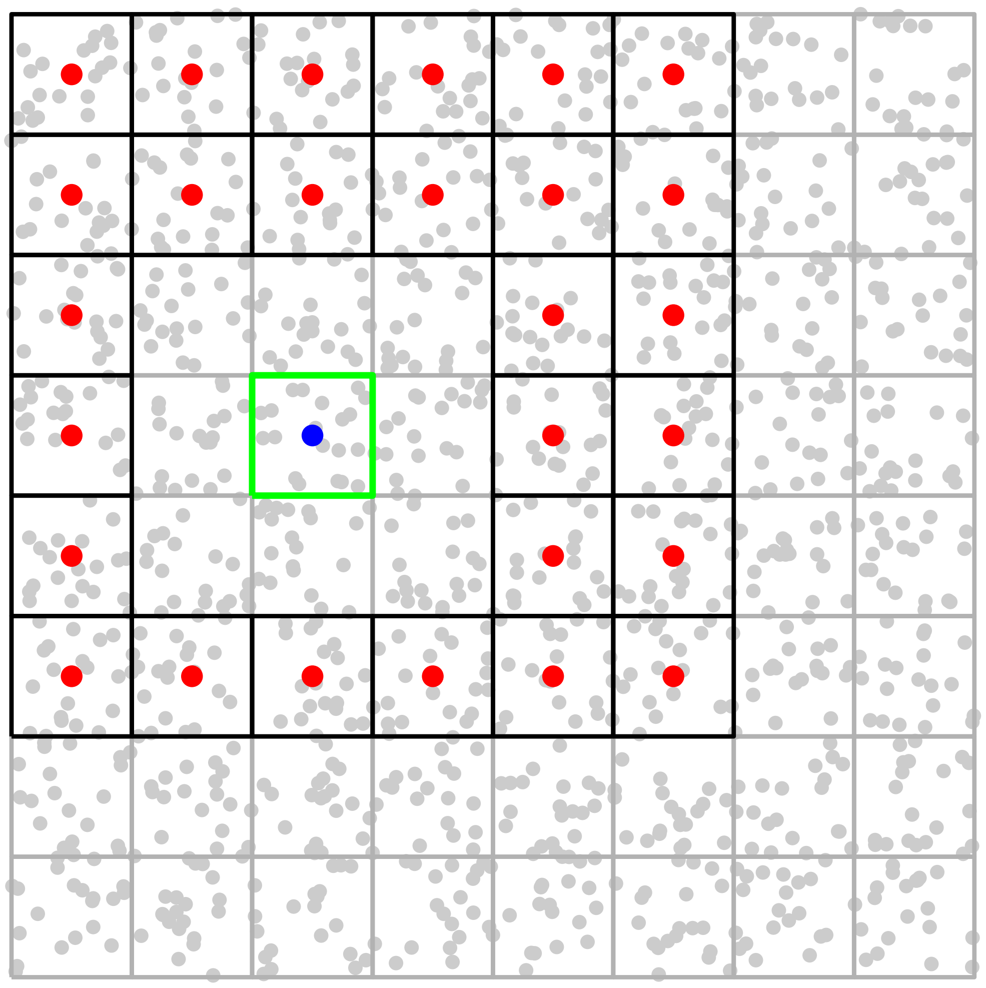
Computing the incoming expansions for all boxes in $O(N)$ operations





We seek to construct the incoming expansion for box τ (marked in green):

Computing the incoming expansions for all boxes in $O(N)$ operations

We seek to construct the incoming expansion for box τ (marked in green):

Transfer the incoming expansion from the parent box, ν, and then add all contributions from boxes in the interaction list:

$$\hat{\mathbf{u}}_\tau = \mathbf{T}^{(\text{ifi})}_{\tau,\nu}\,\hat{\mathbf{u}}_\nu + \sum_{\sigma\in\mathcal{L}^{(\text{int})}_\tau} \mathbf{T}^{(\text{ifo})}_{\tau,\sigma}\,\hat{\mathbf{q}}_\sigma.$$

Summary of the Fast Multipole Method

As an initialization step, build a hierarchical tree of boxes to organize the points.

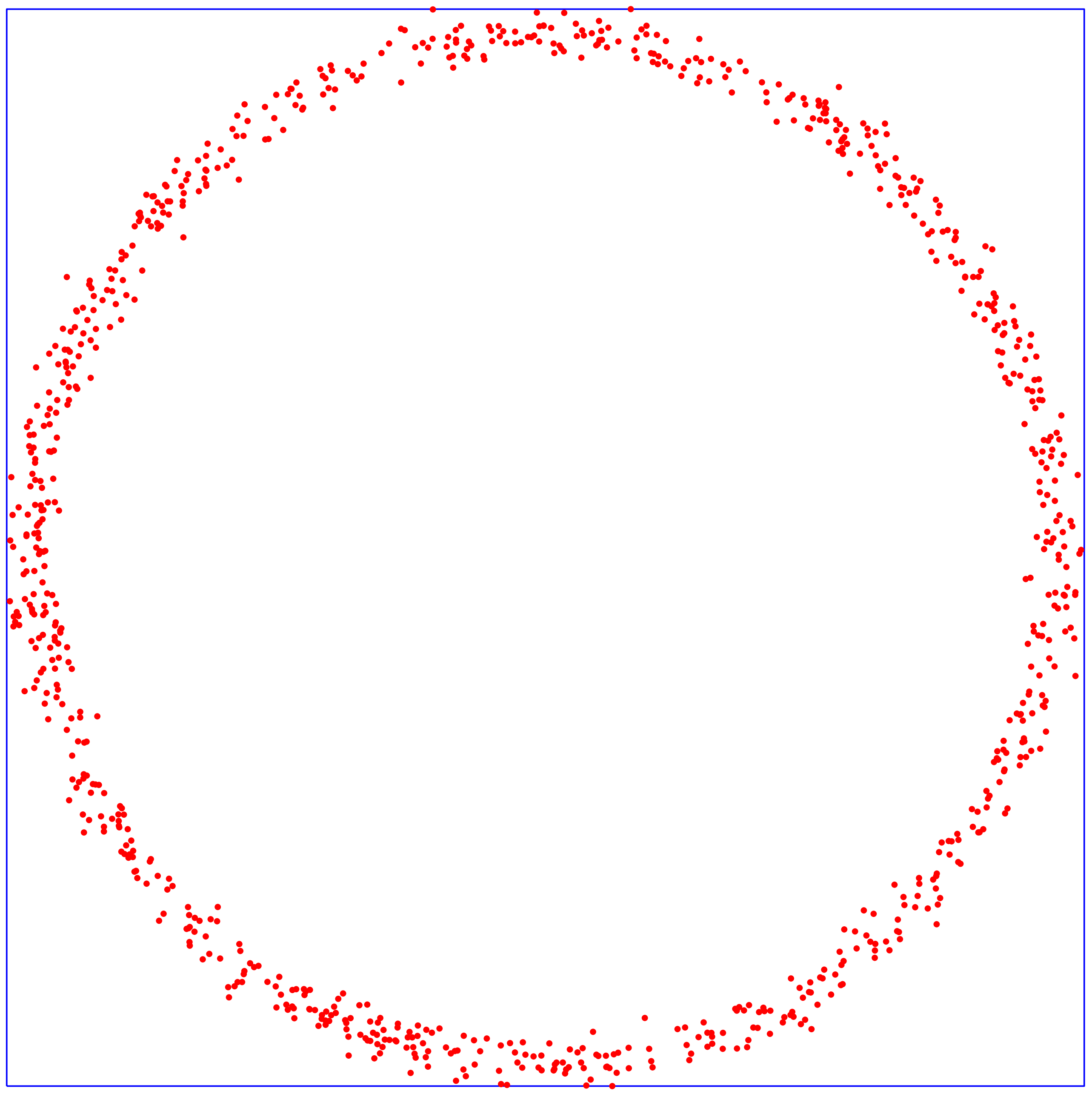
1. *Loop over all leaf boxes in the tree.*
   For each box, compute its outgoing expansion directly from the sources it contains.
2. *Loop over all parent boxes, going from smaller to bigger boxes.*
   For each parent box, compute its outgoing expansion from the expansions of its children.
3. *Loop over all parent boxes, going from bigger to smaller boxes.*
   For each parent box, compute its incoming expansion by combining a contribution from the incoming expansion of its parent, with the contributions from outgoing boxes in its “interaction list”.
4. *Loop over all leaf boxes.*
   For each leaf box, evaluate the contribution from the far-field using the incoming expansion, and add contributions from the near-field via direct evaluation.

Non-uniform particle distributions:

The ability to handle non-uniform distributions is a key advantage of the FMM.

Everything generalizes nicely (some tweaks required).

As an illustration, consider the following set of source locations:

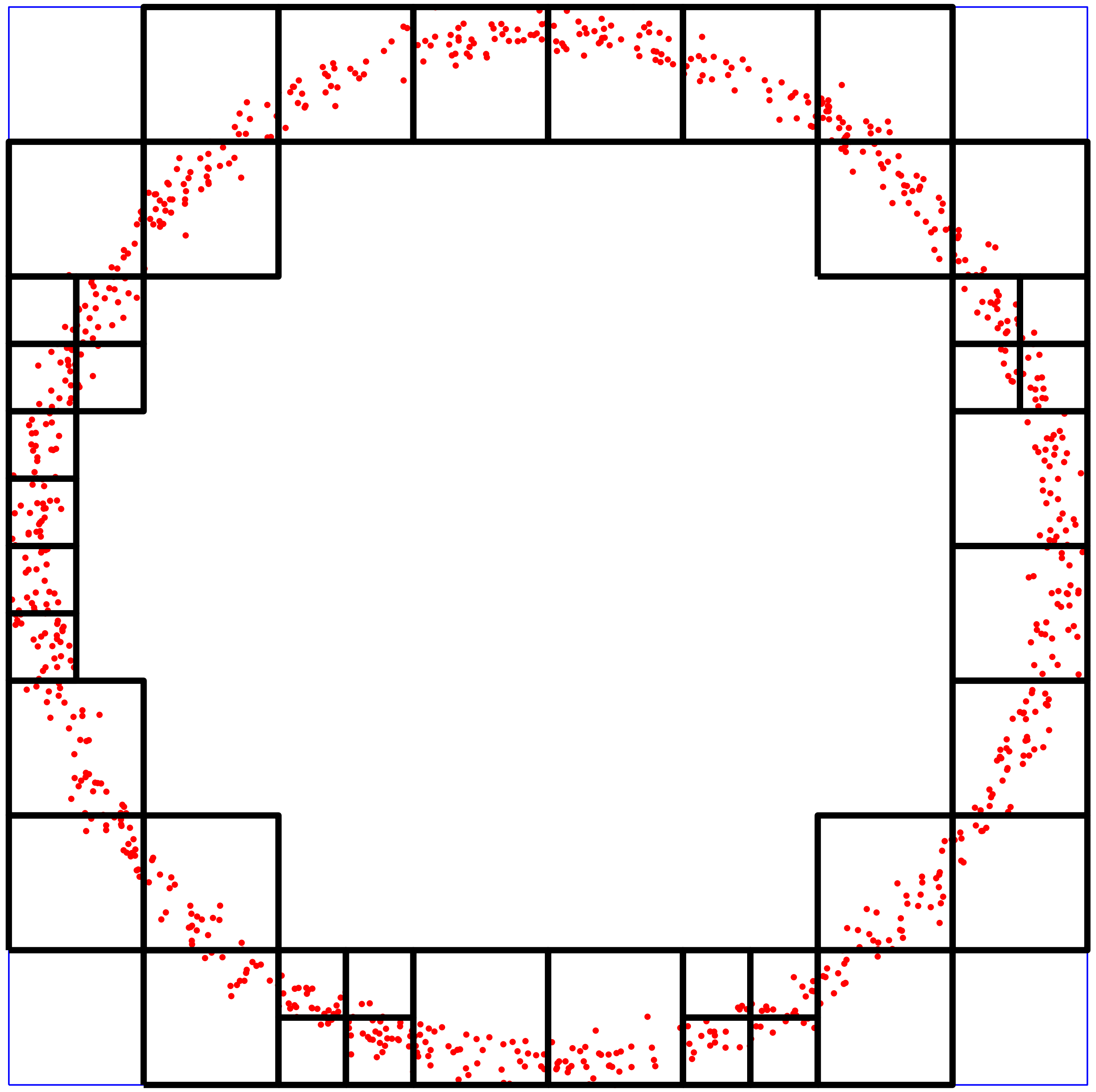


Non-uniform particle distributions:

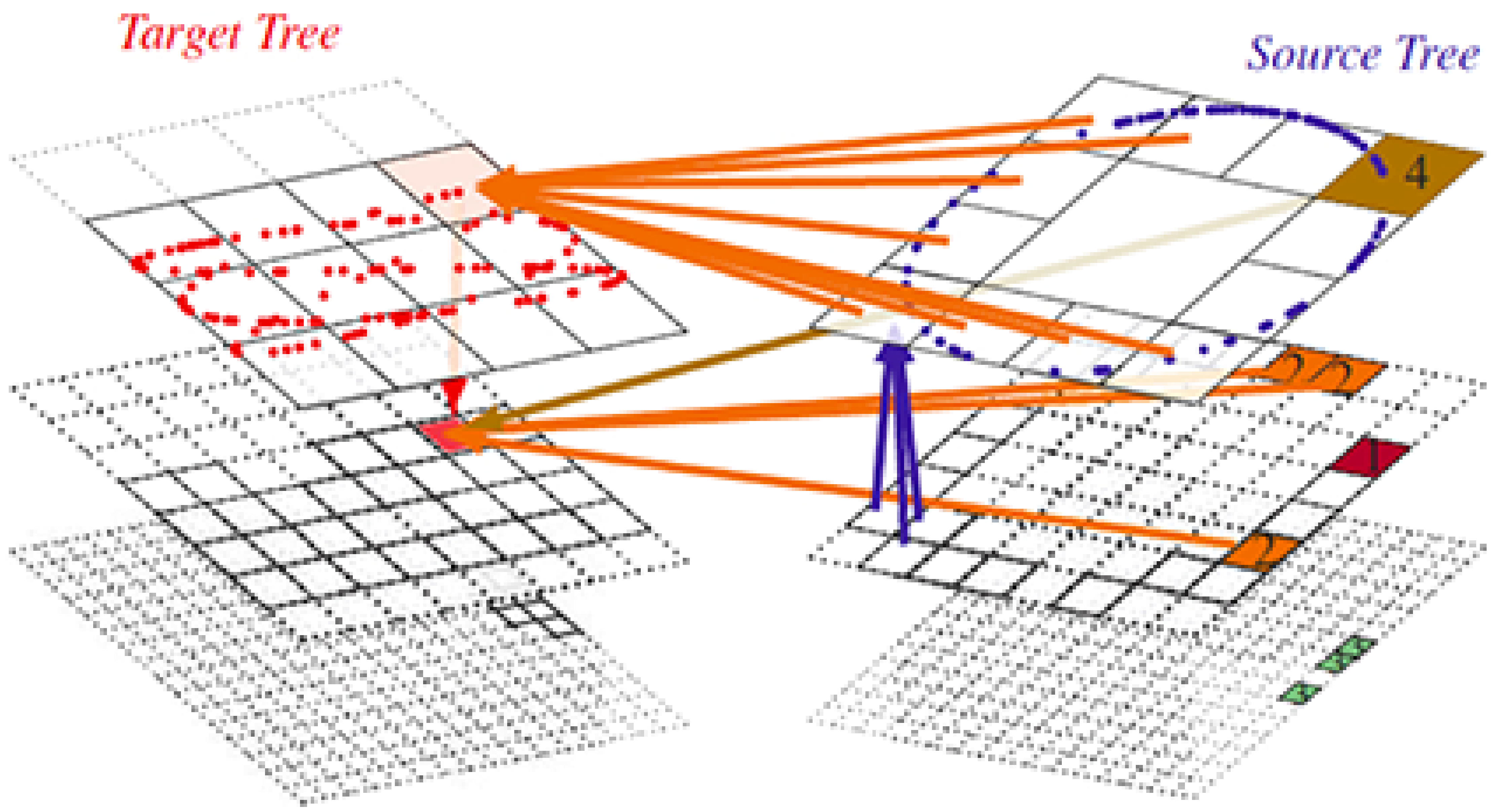
The ability to handle non-uniform distributions is a key advantage of the FMM.

Everything generalizes nicely (some tweaks required).

As an illustration, consider the following set of source locations:



Create boxes by adaptively refining the tree, keeping only non-empty boxes.



Picture from workshop homepage at `https://icerm.brown.edu/programs/sp-s18/w2/`

Important: This introduction to the FMM was brief and *highly* incomplete!

Extension to 3D: In 2D, the "interaction list" has at most 27 elements ($27 = 6^2 - 3^2$). In 3D, it typically has *189 elements* ($189 = 6^3 - 3^3$). Moreover, multipole expansions in 3D converge much more slowly. Instead of $\varepsilon \approx (\sqrt{3}/2)^{p/2}$, we have $\varepsilon \approx (1/\sqrt{3})^{\sqrt{p}}$. This makes the cost of evaluating all "incoming-from-outgoing" translation operators expensive. This cost can be greatly reduced, however, by using more sophisticated translation operators (so called "diagonal forms"). ☞ *Biros talk: Very high dim!*

Helmholtz equation: If the whole computational domain is small in terms of wave-lengths, then the FMM for Helmholtz equation looks very similar to what we just showed. However, if the box is large in terms of wave-lengths, then very different machinery is required. The high-frequency FMM is much more subtle! It does not rely on rank-considerations alone.

Coding the FMM well is not that easy: If you can, it is recommended to use a packaged version. The FMM for Laplace's equation in two dimensions is OK, but 3D is harder, and wideband Helmholtz is substantially harder.

Elliptic equations on lattices.

Joint work with Adrianna Gillman.

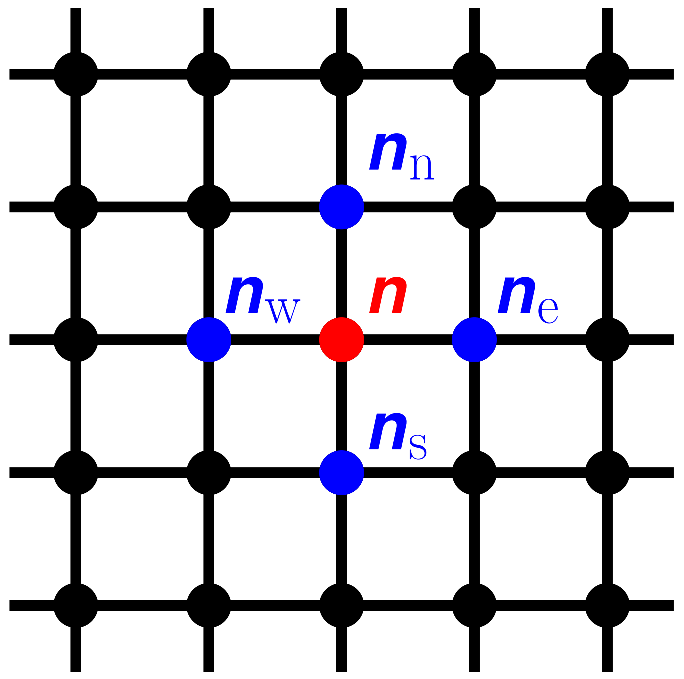
Consider the discrete Poisson equation

$$[Au](\boldsymbol{n}) = f(\boldsymbol{n}), \qquad \boldsymbol{n} \in \mathbb{Z}^2,$$

where A is the discrete Laplace operator

$$[Au](\boldsymbol{n}) = 4\,u(\boldsymbol{n}) - u(\boldsymbol{n}_{\rm s}) - u(\boldsymbol{n}_{\rm e}) - u(\boldsymbol{n}_{\rm n}) - u(\boldsymbol{n}_{\rm w}),$$

where $\boldsymbol{n} \in \mathbb{Z}^2$ is a node in the lattice, and where $\{\boldsymbol{n}_{\rm s},\,\boldsymbol{n}_{\rm e},\,\boldsymbol{n}_{\rm n},\,\boldsymbol{n}_{\rm w}\}$ are the nodes that are immediate neighbors of n to the south, east, north, and west of n, respectively.



We are interested in the discrete Poisson equation as a mathematical object in its own right; *not* as an approximation to the continuum problem $-\Delta u = f$.

Equations of this type come up when studying random walks on lattices, in quantum mechanical models, in modeling mechanical trusses and frames, and many more.

Long-range interactions are very similar to those for the analogous continuum equations, but short range interactions are fundamentally different.

Recall: We are interested in the discrete Poisson equation

$$[Au](\boldsymbol{n}) = f(\boldsymbol{n}), \qquad \boldsymbol{n} \in \mathbb{Z}^2, \tag{4}$$

where A is the discrete Laplace operator $[Au](\boldsymbol{n}) = 4\,u(\boldsymbol{n}) - u(\boldsymbol{n}_{\rm s}) - u(\boldsymbol{n}_{\rm e}) - u(\boldsymbol{n}_{\rm n}) - u(\boldsymbol{n}_{\rm w})$.

We introduce the Fourier transform $\tilde{u}(\boldsymbol{t}) = [Fu](\boldsymbol{t}) = \frac{1}{2\pi}\sum_{\boldsymbol{n}\in\mathbb{Z}^2} u(\boldsymbol{n})\, e^{-i\boldsymbol{n}\cdot\boldsymbol{t}}, \qquad \boldsymbol{t} \in [-\pi,\pi]^2.$

Then $F[Au](\boldsymbol{t}) = \sigma(\boldsymbol{t})\tilde{u}(\boldsymbol{t})$, where

$$\sigma(\boldsymbol{t}) = 4 - e^{it_1} - e^{-it_1} - e^{it_2} - e^{-it_2} = 2\left(\sin\frac{t_1}{2}\right)^2 + 2\left(\sin\frac{t_2}{2}\right)^2 = |\boldsymbol{t}|^2 + O(|\boldsymbol{t}|^4).$$

The equation (4) gets diagonal in Fourier space, $\sigma(\boldsymbol{t})\tilde{u}(\boldsymbol{t}) = \tilde{f}(\boldsymbol{t})$, so its solution is

$$u(\boldsymbol{n}) = F^{-1}\left[\frac{1}{\sigma(\boldsymbol{t})}\tilde{f}(\boldsymbol{t})\right](\boldsymbol{n}), \qquad \boldsymbol{n} \in \mathbb{Z}^2.$$

Multiplication in Fourier space corresponds to convolution in physical space, so in fact

$$u(\boldsymbol{n}) = [\phi * f](\boldsymbol{n}) = \sum_{\boldsymbol{m}\in\mathbb{Z}^2} \phi(\boldsymbol{n}-\boldsymbol{m})\, f(\boldsymbol{m}), \qquad \boldsymbol{n} \in \mathbb{Z}^2,$$

where ϕ is the fundamental solution

$$\phi(\boldsymbol{n}) = F^{-1}\left[\frac{1}{\sigma(\boldsymbol{t})}\right](\boldsymbol{n}) = \frac{1}{2\pi}\int_{[-\pi,\pi]^2} e^{i\boldsymbol{t}\cdot\boldsymbol{n}}\frac{1}{\sigma(\boldsymbol{t})}\,d\boldsymbol{t}, \qquad \boldsymbol{n} \in \mathbb{Z}^2.$$

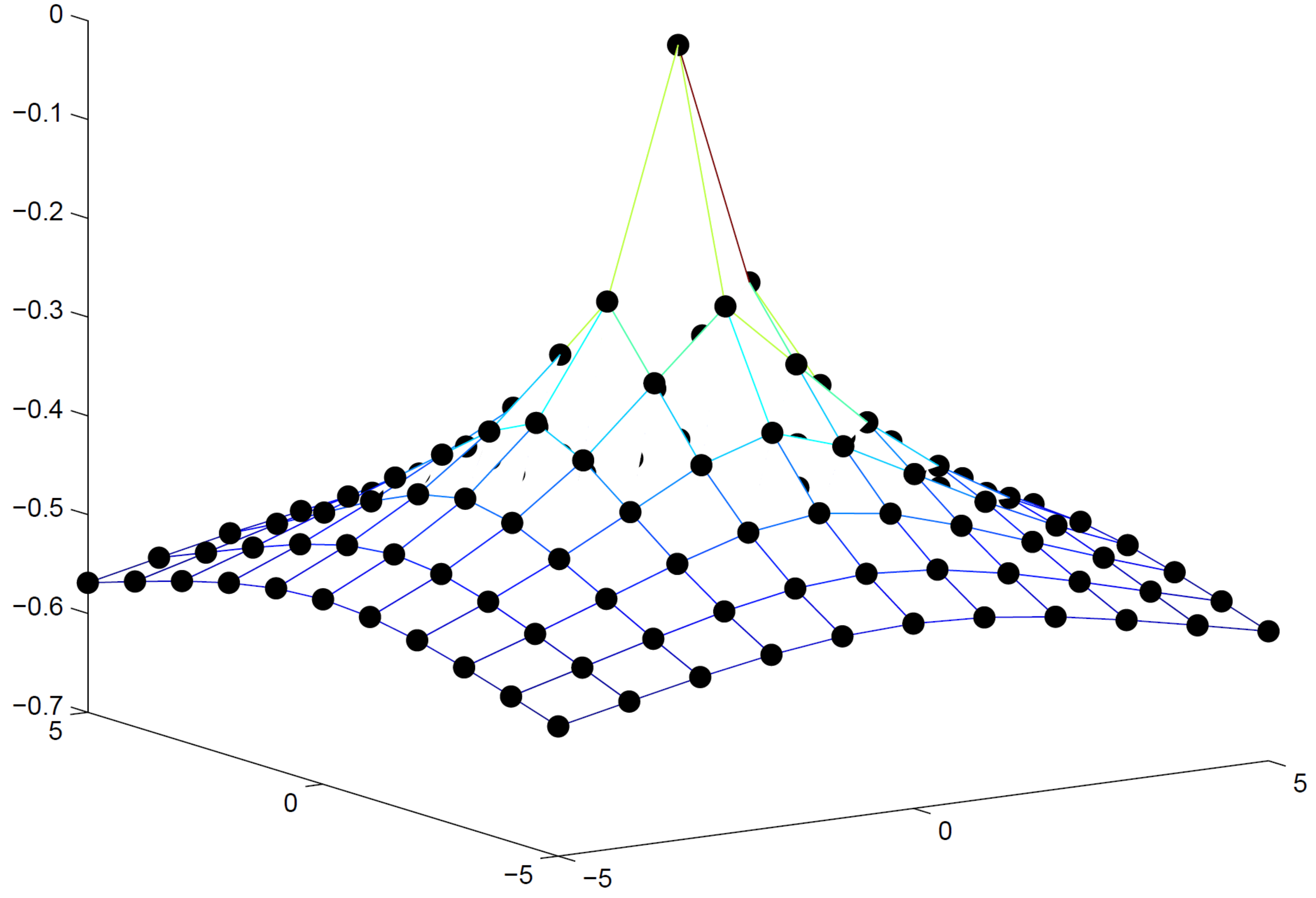
Summary: The solution of the discrete Laplace equation

$$(5)\qquad [Au](\boldsymbol{n}) = f(\boldsymbol{n}), \qquad \boldsymbol{n} \in \mathbb{Z}^2,$$

is given by a convolution against the lattice fundamental solution ϕ so that

$$u(\boldsymbol{n}) = \sum_{\boldsymbol{m}\in\mathbb{Z}^2} \phi(\boldsymbol{n}-\boldsymbol{m})\, f(\boldsymbol{m}), \qquad \boldsymbol{n} \in \mathbb{Z}^2.$$

Plot of $\phi = F^{-1}\left[\frac{1}{\sigma(\boldsymbol{t})}\right]$:



Note: There is a technical issue in that $\frac{1}{\sigma(\boldsymbol{t})} \sim \frac{1}{|\boldsymbol{t}|^2}$, *so the Fourier integral is not well-defined. What is shown is "regularized" version* $\phi(\boldsymbol{n}) = \frac{1}{2\pi}\int_{(-\pi,\pi)^2} \frac{e^{-i\boldsymbol{n}\cdot\boldsymbol{t}} - 1}{\sigma(\boldsymbol{t})}\, d\boldsymbol{t}$.

Summary: The solution of the discrete Laplace equation

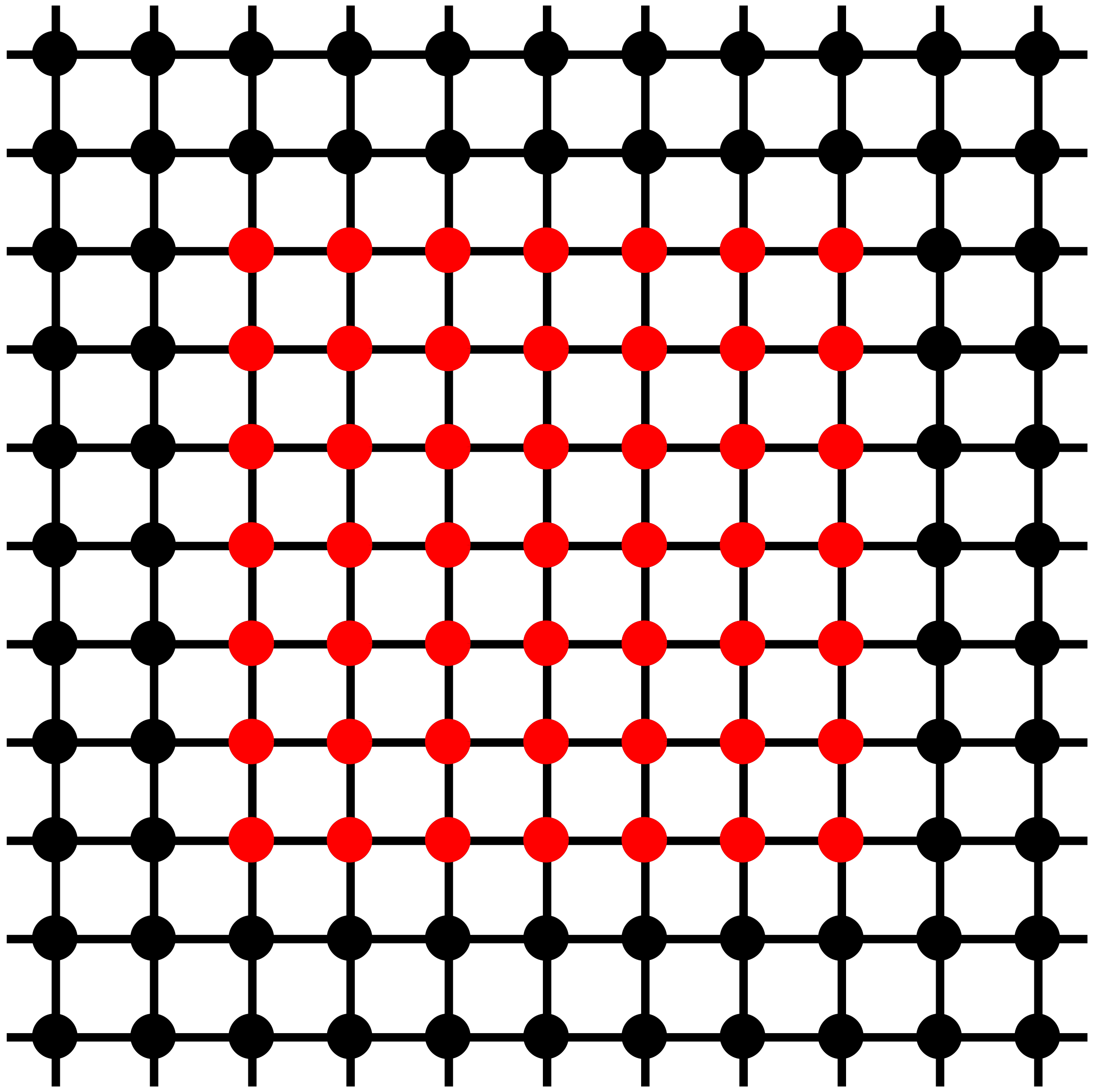
$$[Au](\boldsymbol{n}) = f(\boldsymbol{n}), \qquad \boldsymbol{n} \in \mathbb{Z}^2, \tag{6}$$

is given by a convolution against the lattice fundamental solution ϕ so that

$$u(\boldsymbol{n}) = \sum_{\boldsymbol{m}\in\mathbb{Z}^2} \phi(\boldsymbol{n}-\boldsymbol{m})\, f(\boldsymbol{m}), \qquad \boldsymbol{n} \in \mathbb{Z}^2.$$

Claim: The sum typeset in red is amenable to fast summation techniques analogous to the FMM. Everything works very well.

Special case: A uniform load contained in a box.

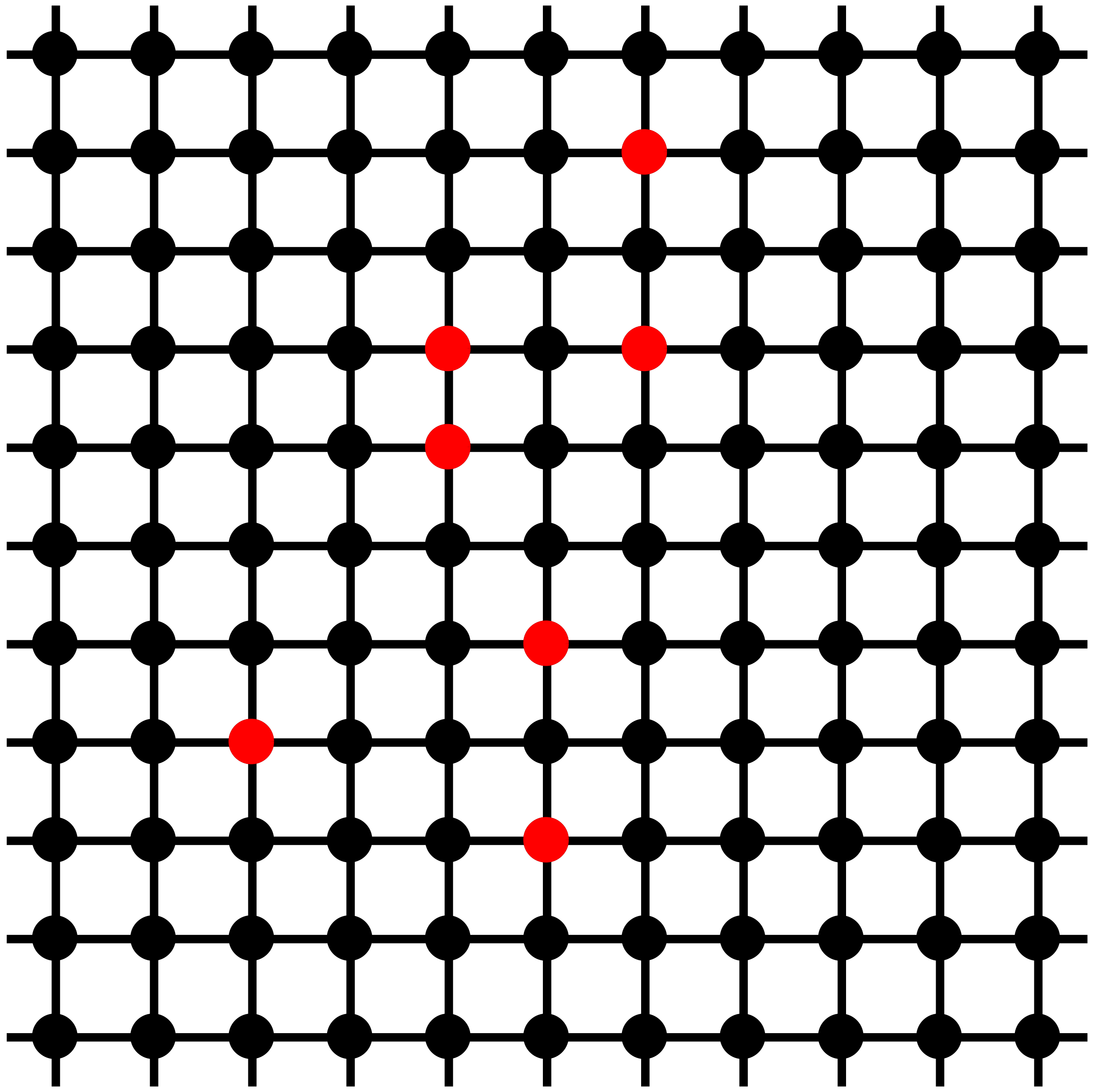


Evaluate $u(\boldsymbol{n}) = \sum_{\boldsymbol{m}\in\mathbb{Z}^2} \phi(\boldsymbol{n}-\boldsymbol{m})\,f(\boldsymbol{m})$ where f is supported on the red nodes.

This works fine. Complexity $O(N)$ where N is number of support nodes.

Hard to compete with the FFT, despite the incorrect boundary conditions.

Special case: A sparse load.

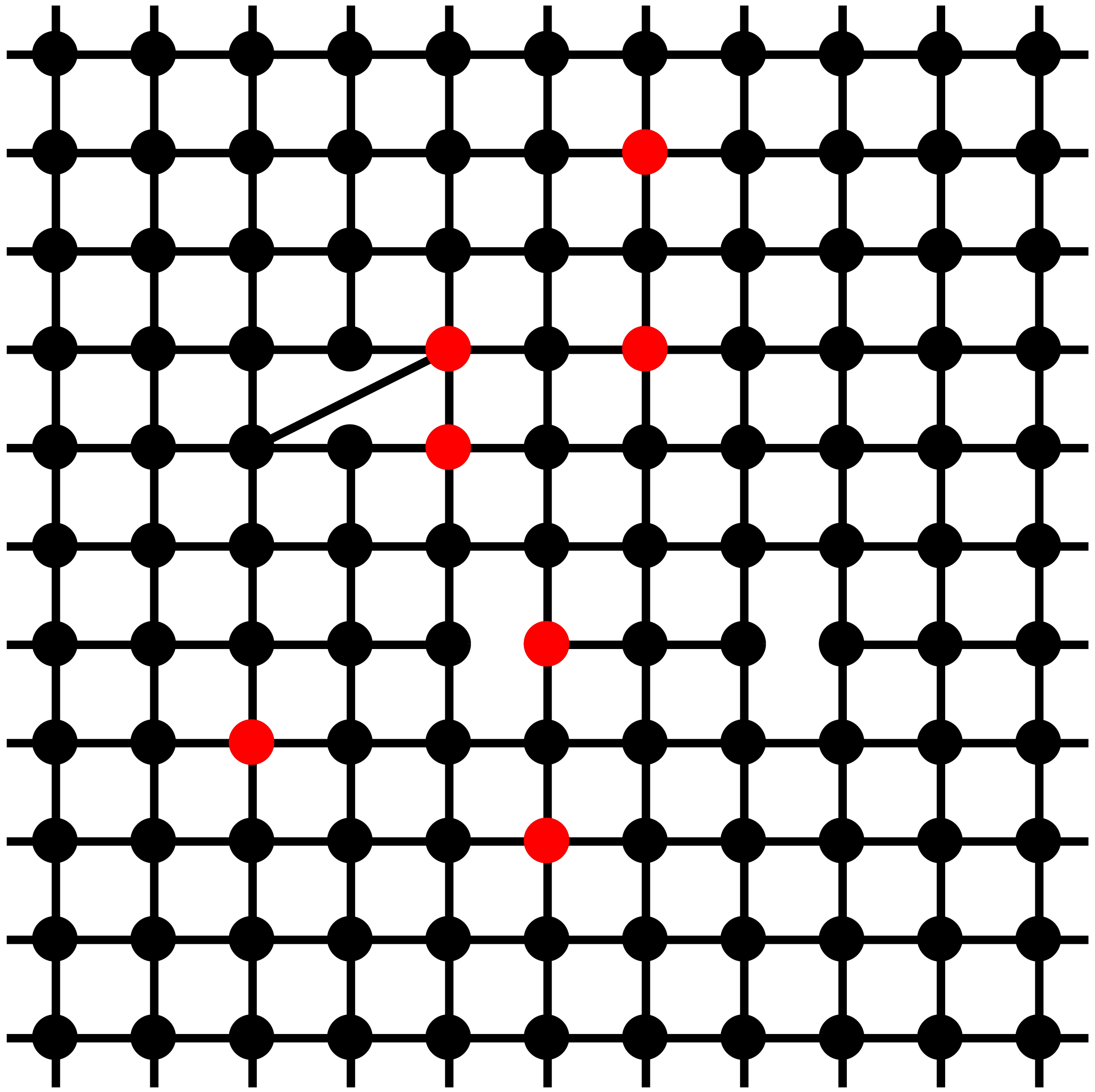


Evaluate $u(\boldsymbol{n}) = \sum_{\boldsymbol{m}\in\mathbb{Z}^2} \phi(\boldsymbol{n}-\boldsymbol{m})\,f(\boldsymbol{m})$ where f is supported on the red nodes.

Works great! Complexity $O(N_{\rm load})$, where $N_{\rm load}$ denotes the number of support nodes.

There is very little competition in this environment. (I think . . . ???)

Special case: A lattice with defects.

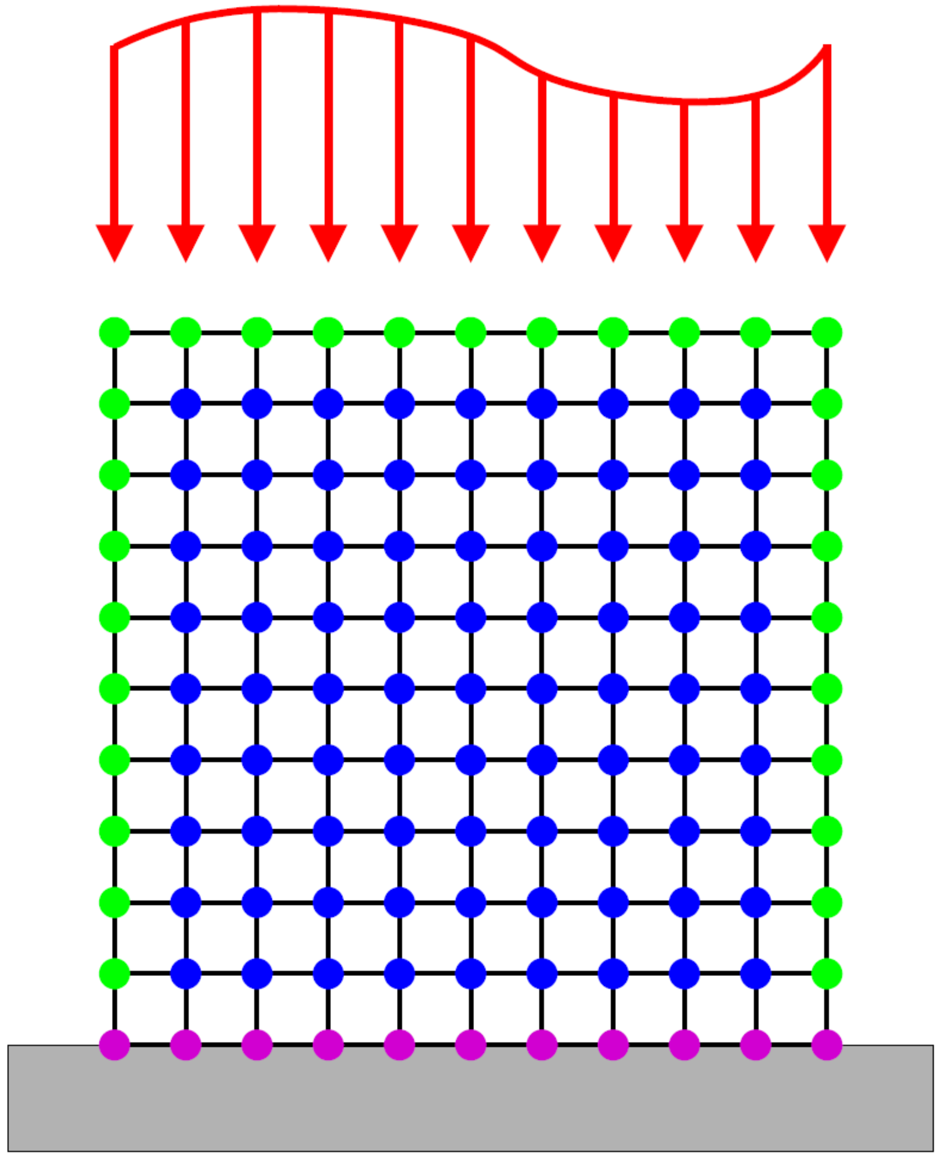


Suppose the lattice has a few deviations from perfect periodicity.

Works great! Complexity is $O(N_{\rm load} + N_{\rm defect}^3)$, where $N_{\rm load}$ is the number of supported nodes, and $N_{\rm defect}$ is the number of deviations from periodicity.

Can actually be reduced to $O(N_{\rm load} + N_{\rm defect})$ in many cases.

Special case: A finite lattice.



Consider a discrete boundary value problem

$$\begin{cases} Au = f, & \text{on } \Omega_{\rm i}, \\ u = g, & \text{on } \Gamma_{\rm D}, \\ \partial_\nu u = h, & \text{on } \Gamma_{\rm N}, \end{cases}$$

where A is the discrete Laplace operator, and where

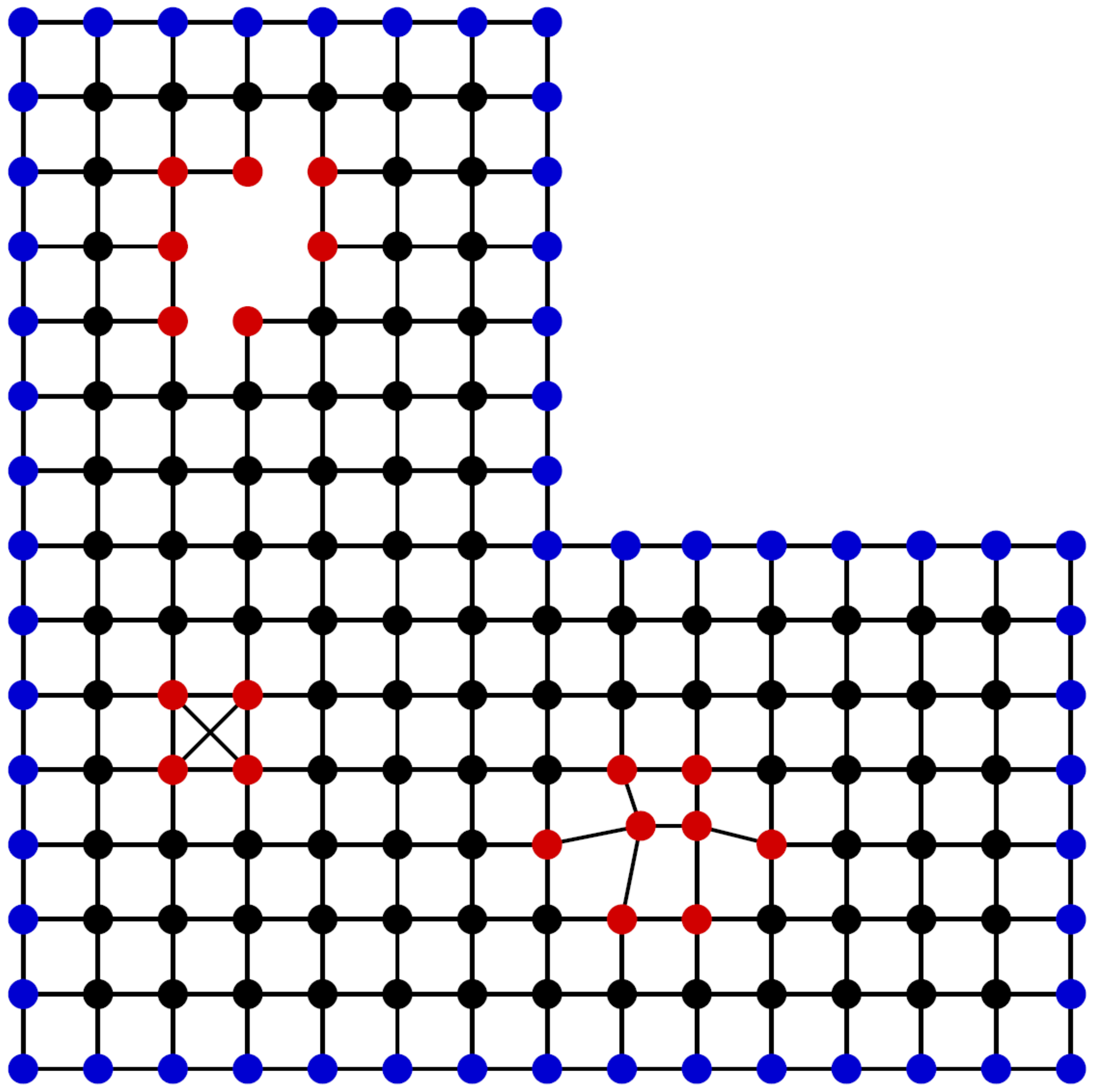
$\Omega_{\rm i}$ are the interior nodes,

$\Gamma_{\rm D}$ is the Dirichlet boundary,

$\Gamma_{\rm N}$ is the Neumann boundary.

Using the discrete fundamental solution and "fast" algorithms, this problem can be solved using $O(N_{\rm boundary})$ operations, where, of course, $N_{\rm boundary}$ refers to the number of boundary nodes.

Special case: A finite lattice — now with defects.



And yes, this problem can also be solved in complexity

$$O(N_{\rm load} + N_{\rm boundary} + N_{\rm defect}^{p}),$$

where $p \in [1,3]$, depending on circumstances.