FMM CMSC 878R/AMSC 698R

Lecture 7

Outline

- Norm of the translation operator
- Example of S|R-translation
- Summary of requirements for functions (potentials) that can be used in FMM
- Idea of a Single Level FMM (SLFMM)
- Space division and expansion domains
- SLFMM algorithm
- Asymptotic complexity of SLFMM
- Optimization of SLFMM

Example from previous lectures

$$\Phi(y, x_i) = \frac{1}{y - x_i}.$$

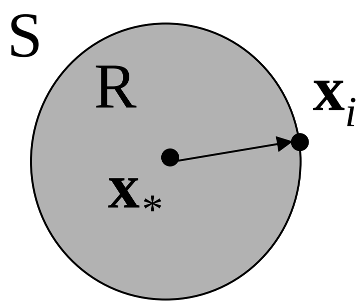
$|y - x_*| < |x_i - x_*|$:

R-expansion

$$\Phi(y, x_i) = \sum_{m=0}^{\infty} a_m(x_i, x_*) R_m(y - x_*),$$

$$a_m(x_i, x_*) = -(x_i - x_*)^{-m-1}, \quad m = 0, 1, \ldots,$$

$$R_m(y - x_*) = (y - x_*)^m, \quad m = 0, 1, \ldots$$



$|y - x_*| > |x_i - x_*|$:

S-expansion

$$\Phi(y, x_i) = \sum_{m=0}^{\infty} b_m(x_i, x_*) S_m(y - x_*),$$

$$b_m(x_i, x_*) = (x_i - x_*)^m, \quad m = 0, 1, \ldots,$$

$$S_m(y - x_*) = (y - x_*)^{-m-1}, \quad m = 0, 1, \ldots$$

In this case we have

$(|y - x_*| < |t|)$

$$S_n(y - x_* + t) = (t + y)^{-n-1} = \sum_{m=0}^{\infty} \frac{1}{m!} \frac{d^m S_n(t)}{dt^n} (y - x_*)^m$$

$$= \sum_{m=0}^{\infty} \frac{1}{m!} \frac{d^m S_n(t)}{dt^n} R_m(y - x_*) = \sum_{m=0}^{\infty} (S|R)_{mn}(t) R_m(y - x_*).$$

So

$$(S|R)_{mn}(t) = \frac{1}{m!} \frac{d^m S_n(t)}{dt^n} = \frac{(-1)^m (m+n)!}{m! n! t^{n+m+1}}.$$

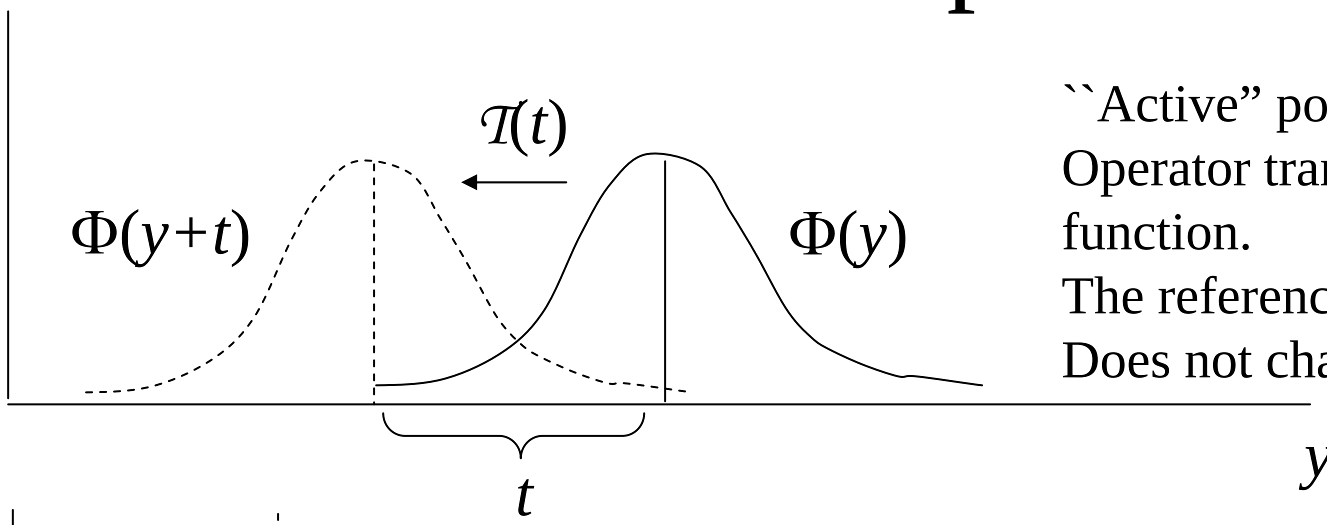
$$(\mathbf{S}|\mathbf{R})(t) = \begin{pmatrix} t^{-1} & t^{-2} & t^{-3} & \ldots \\ -t^{-2} & -2t^{-3} & -3t^{-4} & \ldots \\ t^{-3} & 3t^{-4} & 6t^{-5} & \ldots \\ \ldots & \ldots & \ldots & \ldots \end{pmatrix}$$

Norm of the Translation Operator

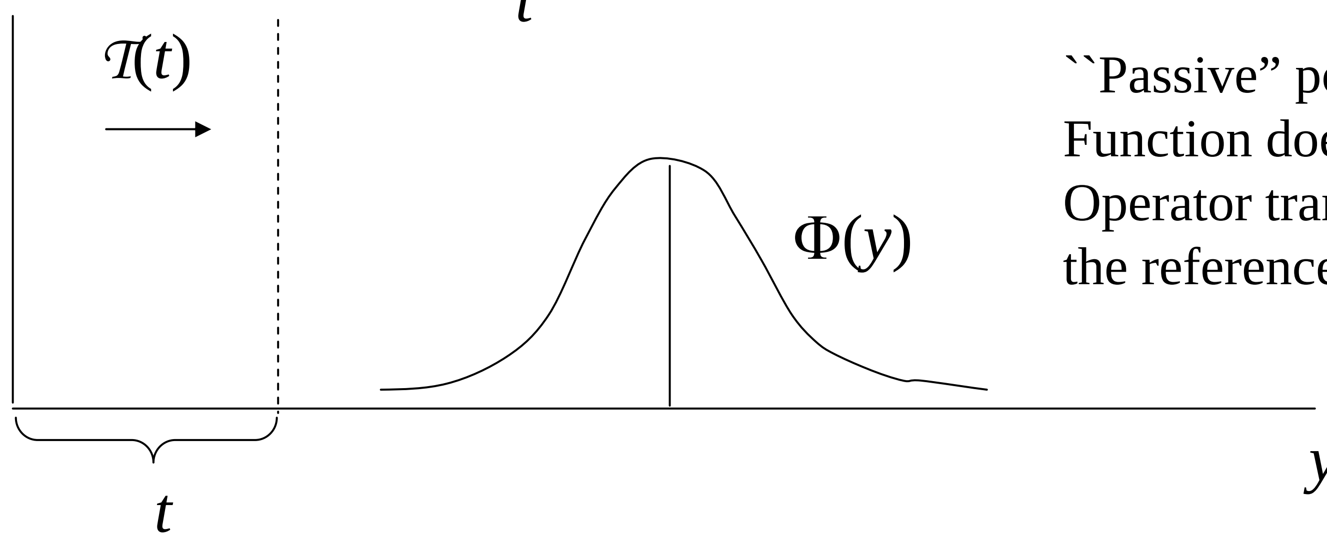
Theorem. Let $\mathbb{F}(\Omega)$ be a set of functions bounded in $\mathbb{R}^d$. Then $\| \mathcal{T}(\mathbf{t}) \| = 1$.
Proof.

$$\| \mathcal{T}(\mathbf{t}) \| = \frac{\| \mathcal{T}(\mathbf{t})\Phi(\mathbf{y}) \|}{\| \Phi(\mathbf{y}) \|} = \frac{\| \Phi(\mathbf{y}+\mathbf{t}) \|}{\| \Phi(\mathbf{y}) \|} = \frac{\sup_{\mathbf{y}\in\mathbb{R}^d} |\Phi(\mathbf{y}+\mathbf{t})|}{\sup_{\mathbf{y}\in\mathbb{R}^d} |\Phi(\mathbf{y})|} = 1.$$

Active and Passive points of view on translation operator

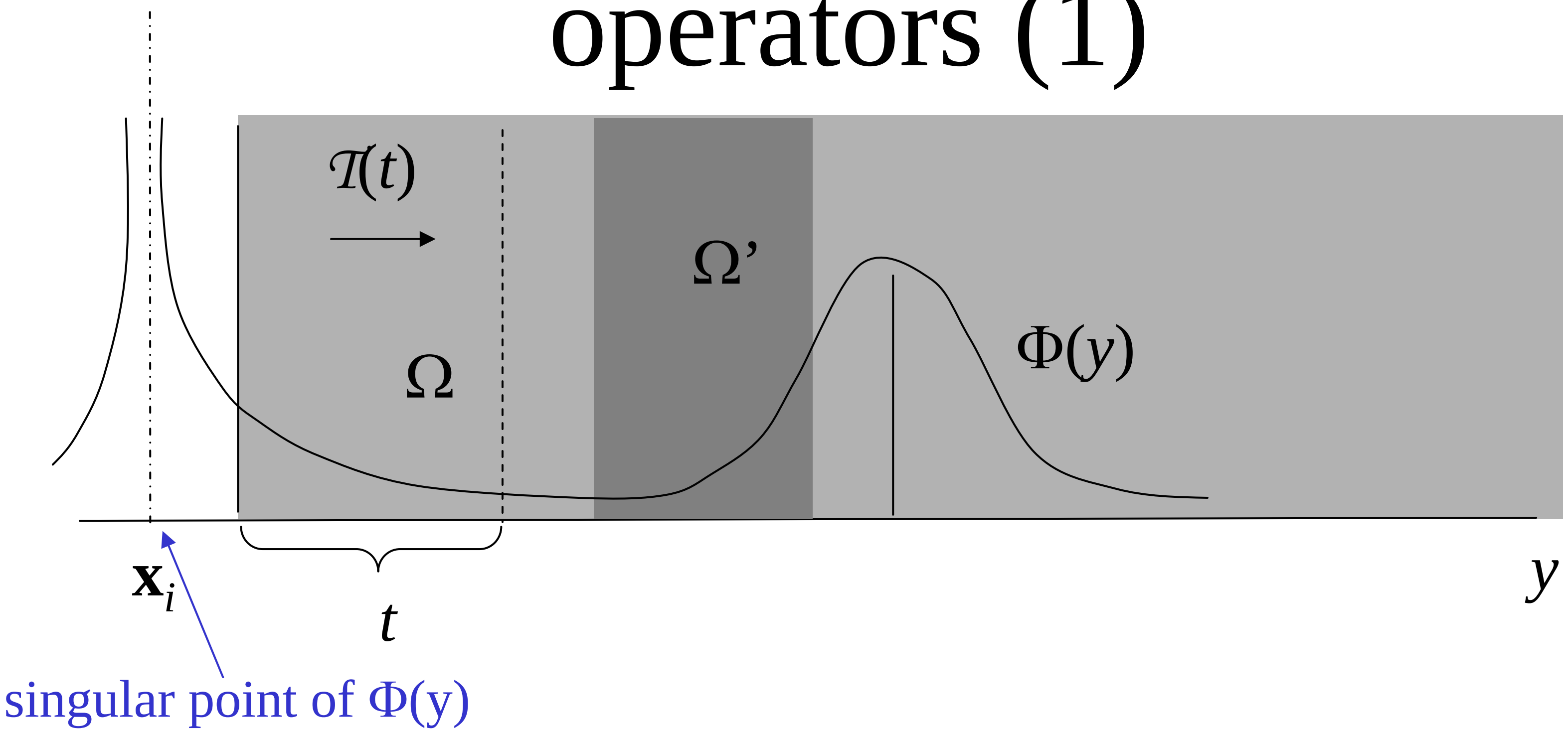


``Active" point of view:
Operator transforms
function.
The reference frame
Does not change.



``Passive" point of view:
Function does not change.
Operator transforms
the reference frame.

Norms of $R|R$, $S|S$, and $S|R$-operators (1)



$\Phi(\mathbf{y})$ is bounded in Ω.
$\Omega' \subset \Omega$.
Therefore $\Phi(\mathbf{y})$ is bounded in Ω', and

$$\|\Phi(\mathbf{y})\|_{\Omega'} = \sup_{\mathbf{y}\in\Omega'}|\Phi(\mathbf{y})| \leqslant \sup_{\mathbf{y}\in\Omega}|\Phi(\mathbf{y})| = \|\Phi(\mathbf{y})\|_{\Omega}.$$

Norms of *R*|*R*, *S*|*S*, and *S*|*R*-operators (2)

From the passive point of view, the translation operator does nothing, but just changes the reference frame. So if we consider that R|R, S|S, and S|R do just change of the reference frame **PLUS** *they shrink the domain, where the function is bounded, then their norms do not exceed* 1.

$$\Omega' \subset \Omega$$

$$\|(\mathcal{R}|\mathcal{R})(\mathbf{t})\| = \frac{\sup_{\mathbf{y}\in\Omega'}|\Phi(\mathbf{y})|}{\sup_{\mathbf{y}\in\Omega}|\Phi(\mathbf{y})|} \leqslant 1,$$

$$\|(\mathcal{S}|\mathcal{S})(\mathbf{t})\| = \frac{\sup_{\mathbf{y}\in\Omega'}|\Phi(\mathbf{y})|}{\sup_{\mathbf{y}\in\Omega}|\Phi(\mathbf{y})|} \leqslant 1,$$

$$\|(\mathcal{S}|\mathcal{R})(\mathbf{t})\| = \frac{\sup_{\mathbf{y}\in\Omega'}|\Phi(\mathbf{y})|}{\sup_{\mathbf{y}\in\Omega}|\Phi(\mathbf{y})|} \leqslant 1.$$

This is the difference between general translation operator and *R*|*R*, *S*|*S*, and *S*|*R* operators.

Error of exact *R*|*R*, *S*|*S*, and *S*|*R*-translation

If

$$\|\Phi(\mathbf{y}) - \Phi^p(\mathbf{y})\| < \epsilon,$$

then

$$\|(\mathcal{R}|\mathcal{R})(\mathbf{t})(\Phi(\mathbf{y}) - \Phi^p(\mathbf{y}))\| = \|(\mathcal{R}|\mathcal{R})(\mathbf{t})\| \|\Phi(\mathbf{y}) - \Phi^p(\mathbf{y})\| < \epsilon,$$

$$\|(\mathcal{S}|\mathcal{S})(\mathbf{t})(\Phi(\mathbf{y}) - \Phi^p(\mathbf{y}))\| = \|(\mathcal{S}|\mathcal{S})(\mathbf{t})\| \|\Phi(\mathbf{y}) - \Phi^p(\mathbf{y})\| < \epsilon,$$

$$\|(\mathcal{S}|\mathcal{R})(\mathbf{t})(\Phi(\mathbf{y}) - \Phi^p(\mathbf{y}))\| = \|(\mathcal{S}|\mathcal{R})(\mathbf{t})\| \|\Phi(\mathbf{y}) - \Phi^p(\mathbf{y})\| < \epsilon.$$

Four Key Stones of FMM

- Factorization
- Error
- Translation
- Grouping

Summary of formal requirements for functions that can be used in FMM

- We have two sets of points:

$$\mathsf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}, \quad \mathbf{x}_i \in \mathsf{R}^d, \quad i = 1, ..., N,$$
$$\mathsf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_M\}, \quad \mathbf{y}_j \in \mathsf{R}^d, \quad j = 1, ..., M.$$

- We have functions (potentials):

$$\Phi(\mathbf{x}_i, \mathbf{y}) : \mathsf{R}^d \to \mathsf{R}, \quad \mathbf{y} \in \mathsf{R}^d, \quad i = 1, ..., N.$$

- These functions can be factorized as (local expansion):

$$\Phi(\mathbf{x}_i, \mathbf{y}) = \mathbf{A}(\mathbf{x}_i, \mathbf{x}_*) \circ \mathbf{R}(\mathbf{y} - \mathbf{x}_*), \quad |\mathbf{y} - \mathbf{x}_*| < r < |\mathbf{x}_i - \mathbf{x}_*|, \quad i = 1, ..., N$$

- These functions can be factorized as (far field expansion):

$$\Phi(\mathbf{x}_i, \mathbf{y}) = \mathbf{B}(\mathbf{x}_i, \mathbf{x}_*) \circ \mathbf{S}(\mathbf{x} - \mathbf{x}_*), \quad |\mathbf{y} - \mathbf{x}_*| > R > |\mathbf{x}_i - \mathbf{x}_*|, \quad i = 1, ..., N$$

- The product is distributive operation with respect to addition

$$(u_1\mathbf{A}_1 + u_2\mathbf{A}_2) \circ \mathbf{F} = u_1\mathbf{A}_1 \circ \mathbf{F} + u_2\mathbf{A}_2 \circ \mathbf{F}, \quad \mathbf{F} = \mathbf{S}, \mathbf{R}$$

Summary of formal requirements for functions that can be used in FMM (2)

- R-expansion coefficients can be $R|R$-translated:

$$|\mathbf{x} - \mathbf{x}_{*2}| < |\mathbf{x}_i - \mathbf{x}_{*1}| - |\mathbf{x}_{*1} - \mathbf{x}_{*2}| :$$

$$\mathbf{A}(\mathbf{x}_i, \mathbf{x}_{*2}) = (\mathbf{R}|\mathbf{R})(\mathbf{x}_{*2} - \mathbf{x}_{*1})\mathbf{A}(\mathbf{x}_i, \mathbf{x}_{*1})$$

- S-expansion coefficients can be $S|S$-translated:

$$|\mathbf{x} - \mathbf{x}_{*2}| > |\mathbf{x}_{*1} - \mathbf{x}_{*2}| + |\mathbf{x}_i - \mathbf{x}_{*1}|,$$

$$\mathbf{B}(\mathbf{x}_i, \mathbf{x}_{*2}) = (\mathbf{S}|\mathbf{S})(\mathbf{x}_{*2} - \mathbf{x}_{*1})\mathbf{B}(\mathbf{x}_i, \mathbf{x}_{*1})$$

- S-expansion coefficients can be $S|R$-translated (converted to R-expansion coefficients)

$$|\mathbf{x} - \mathbf{x}_{*2}| < |\mathbf{x}_{*1} - \mathbf{x}_{*2}| + |\mathbf{x}_i - \mathbf{x}_{*1}|,$$

$$\mathbf{A}(\mathbf{x}_i, \mathbf{x}_{*2}) = (\mathbf{S}|\mathbf{R})(\mathbf{x}_{*2} - \mathbf{x}_{*1})\mathbf{B}(\mathbf{x}_i, \mathbf{x}_{*1})$$

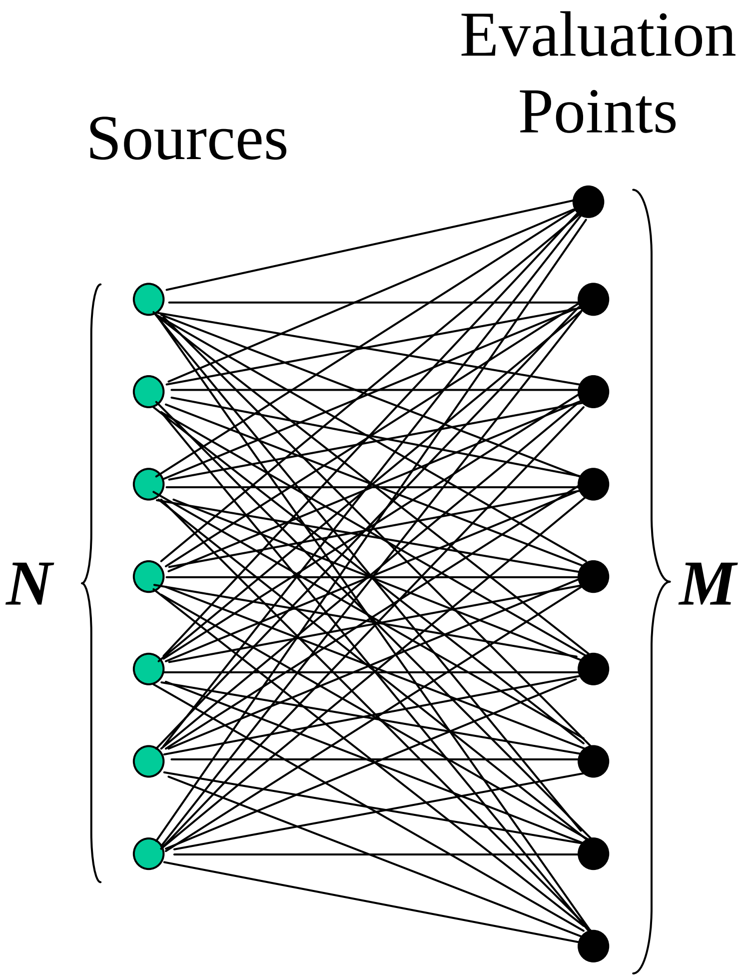
- And we are looking for sums:

$$v_j = \sum_{i=1}^{N} u_i \Phi(\mathbf{y}_j, \mathbf{x}_i), \quad j = 1, \ldots, M.$$

- Some generalization are possible, say instead of $\Phi(\mathbf{y}_j, \mathbf{x}_i)$ we can consider $\Phi_i(\mathbf{y}_j)$, etc.

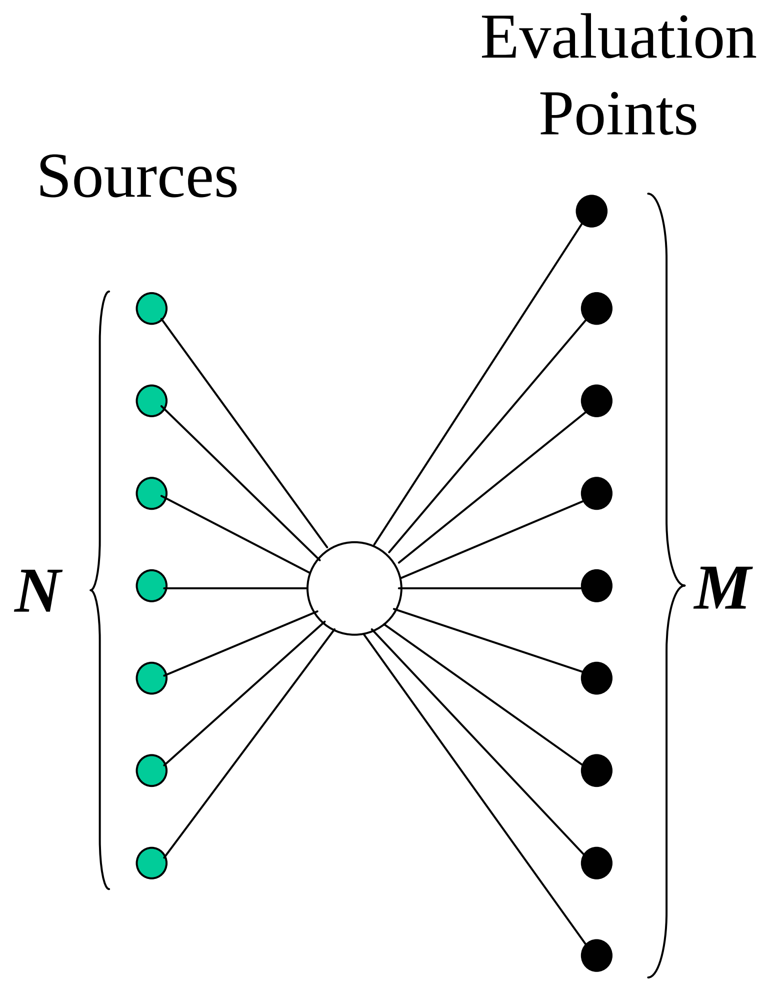
Middleman Algorithm

Standard algorithm



Total number of operations: ***O(NM)***

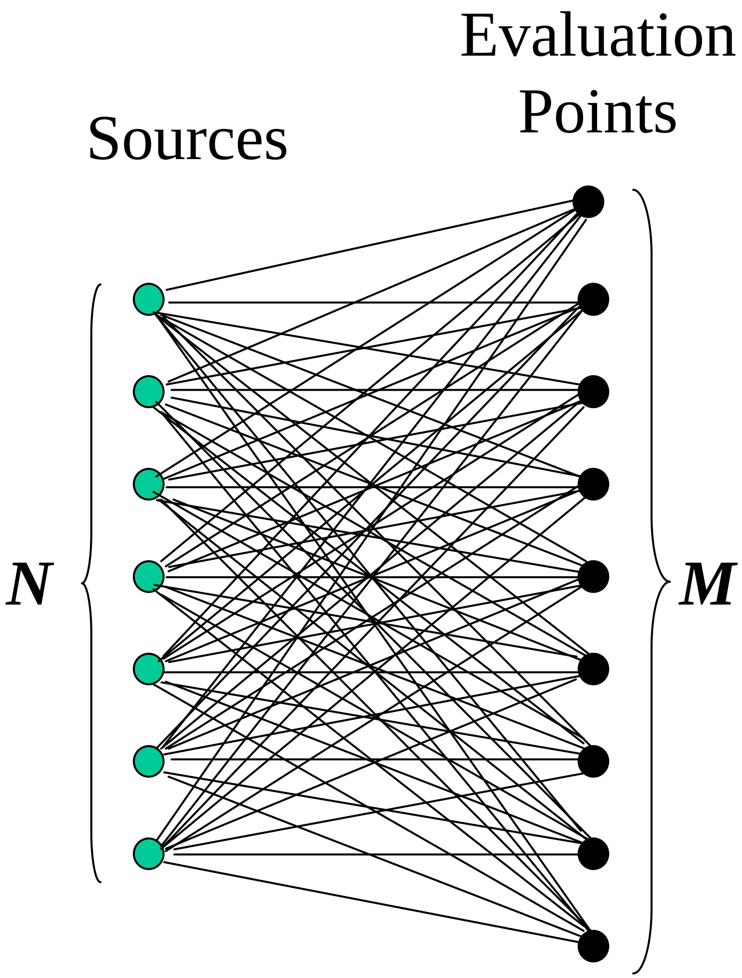
Middleman algorithm



Total number of operations: ***O(N+M*)**

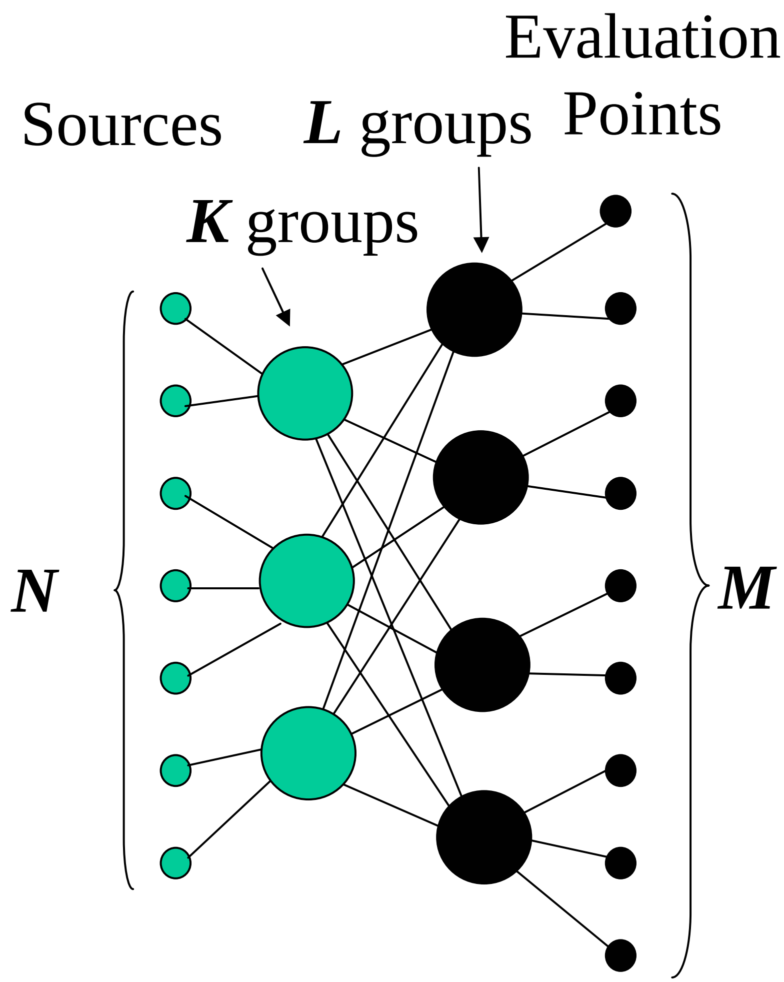
Idea of a Single Level FMM

Standard algorithm



Total number of operations: ***O(NM)***

SLFMM



Total number of operations: ***O(N+M+KL)***

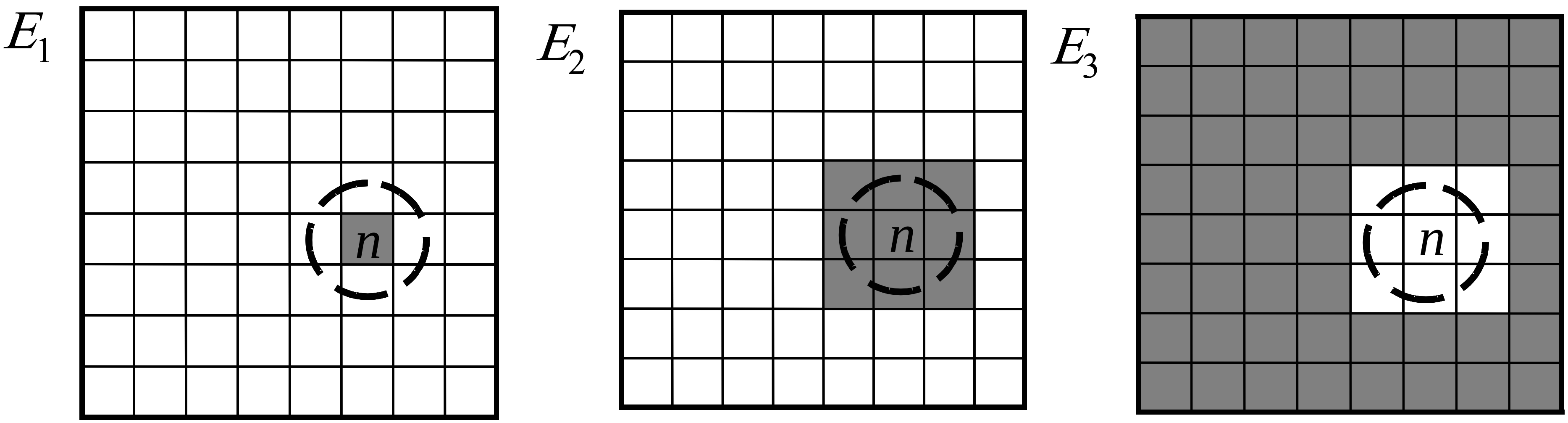
Why do we need SLFMM if Middleman has smaller complexity?

- Expansions can be valid in domains smaller than the computational domain.
- Even though expansion can be valid everywhere, the truncation number can be huge for large domains to provide accuracy.
- Sources and evaluation points can be spatially close, and there is a problem to evaluate singular potentials.
- Important theoretical question: determining optimal number of groups automatically

Spatial Domains

Potentials due to sources in these spatial domains

$\Phi_1^{(n)}(\mathbf{y})$ $\Phi_2^{(n)}(\mathbf{y})$ $\Phi_3^{(n)}(\mathbf{y})$



$I_1(n) = n$

$I_2(n) = \{Neighbors(n)\} \cup n$

$I_3(n) = \{All\ boxes\} \setminus I_2(n)$

Boxes with these numbers belong to these spatial domains

Definition of potentials

$$\Phi_1^{(n)}(\mathbf{y}) = \sum_{\mathbf{x}_i \in E_1(n)} u_i \Phi(\mathbf{y}, \mathbf{x}_i),$$

$$\Phi_2^{(n)}(\mathbf{y}) = \sum_{\mathbf{x}_i \in E_2(n)} u_i \Phi(\mathbf{y}, \mathbf{x}_i),$$

$$\Phi_3^{(n)}(\mathbf{y}) = \sum_{\mathbf{x}_i \in E_3(n)} u_i \Phi(\mathbf{y}, \mathbf{x}_i),$$

Since domains $E_2(n)$ and $E_3(n)$ are complimentary:

$$\Phi(\mathbf{y}) = \sum_{i=1}^{N} u_i \Phi(\mathbf{y}, \mathbf{x}_i) = \sum_{\mathbf{x}_i \in E_2(n) \cup E_3(n)} u_i \Phi(\mathbf{y}, \mathbf{x}_i) = \Phi_2^{(n)}(\mathbf{y}) + \Phi_3^{(n)}(\mathbf{y})$$

for arbitrary n.

SLFMM Algorithm

Step 1. Generate S-expansion coefficients for each box

$$\Phi_1^{(n)}(\mathbf{x}) = \mathbf{C}^{(n)} \circ \mathbf{S}(\mathbf{x} - \mathbf{x}_c^{(n)}),$$
$$\mathbf{C}^{(n)} = \sum_{\mathbf{x}_i \in E_1(n,L)} u_i \mathrm{B}\left(\mathbf{x}_i, \mathbf{x}_c^{(n)}\right).$$

loop over all non-empty source boxes

$For\ \ n \in NonEmptySource$

Get $\mathbf{x}_c^{(n)}$, the center of the box;

$\mathbf{C}^{(n)} = \mathbf{0};$

loop over all sources in the box

$For\ \mathbf{x}_i \in E_1(n)$

Get $\mathbf{B}\ (\mathbf{x}_i, \mathbf{x}_c^{(n)})$, the S-expansion coefficients near the center of the box;

$\mathbf{C}^{(n)} = \mathbf{C}^{(n)} + u_i\mathbf{B}\ (\mathbf{x}_i, \mathbf{x}_c^{(n)});$

End;

End;

Implementation can be different!
All we need is to get $\mathbf{C}^{(n)}$.

SLFMM Algorithm

Step 2. (S|R)-translate expansion coefficients

$$\Phi_3^{(n)}(\mathbf{y}) = \mathbf{D}^{(n)} \circ \mathrm{R}(\mathbf{y} - \mathbf{x}_c^{(n)}),$$

$$\mathbf{D}^{(n)} = \sum_{m \in I_3(n)} (\mathbf{S}|\mathrm{R})\left(\mathbf{x}_c^{(n)} - \mathbf{x}_c^{(m)}\right)\mathbf{C}^{(m)}.$$

loop over all non-empty evaluation boxes

For $n \in NonEmptyEvaluation$

Get $\mathbf{x}_c^{(n)}$, the center of the box;

$\mathbf{D}^{(n)} = \mathbf{0}$;

loop over all non-empty source boxes outside the neighborhood of the *n*-th box

For $m \in I_3(n)$

Get $\mathbf{x}_c^{(m)}$, the center of the box;

$\mathbf{D}^{(n)} = \mathbf{D}^{(n)} + (\mathbf{S}|\mathbf{R})(\mathbf{x}_c^{(n)} - \mathbf{x}_c^{(m)})\ \mathbf{C}^{(m)}$;

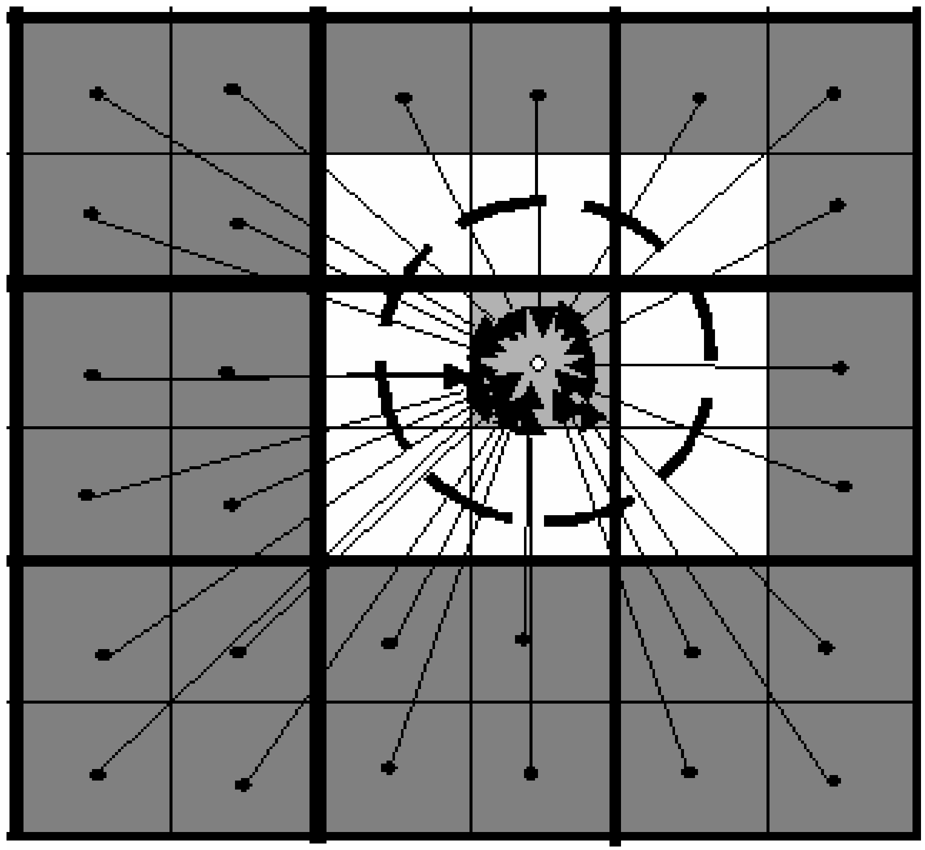
End;

End;

Implementation can be different!

All we need is to get $\mathbf{D}^{(n)}$.

S|R-translation



SLFMM Algorithm

Step 3. Final Summation

$$v_j = \Phi(\mathbf{y}_j) = \sum_{\mathbf{x}_i \in E_2(n)} \Phi(\mathbf{y}_j, \mathbf{x}_i) + \mathbf{D}^{(n)} \circ \mathbf{R}(\mathbf{y}_j - \mathbf{x}_c^{(n)}), \quad \mathbf{y}_j \in E_1(n).$$

For $n \in NonEmptyEvaluation$ ← loop over all boxes containing evaluation points

Get $\mathbf{x}_c^{(n)}$, the center of the box;

For $\mathbf{y}_j \in E_1(n)$ ← loop over all evaluation points in the box

$v_j = \mathbf{D}^{(n)} \circ \mathbf{R}(\mathbf{y}_j - \mathbf{x}_c^{(n)})$;

For $\mathbf{x}_i \in E_2(n)$ ← loop over all sources in the neighborhood of the *n*-th box

$v_j = v_j + \Phi(\mathbf{y}_j , \mathbf{x}_i)$;

End;

End;

End;

Implementation can be different!
All we need is to get v_j

Asymptotic Complexity of SLFMM

Assume that:

- By some magic we can easily find neighbors, and lists of points in each box.
- Translation is performed by straightforward PxP matrix-vector multiplication, where $P(p)$ is the total length of the translation vector. So the complexity of a single translation is $O(P^2)$.
- The source and evaluation points are distributed uniformly, and there are K boxes, with s source points in each box ($s=N/K$). We call s the *grouping* (or *clustering*) parameter.
- The number of neighbors for each box is $O(1)$.

Then Complexity is:

- For Step 1: $O(PN)$
- For Step 2: $O(P^2K^2)$
- For Step 3: $O(PM+Ms)$
- Total: $O(PN+ P^2K^2 +PM+Ms) =$
  $O(PN+ P^2K^2 +PM+MN/K)$

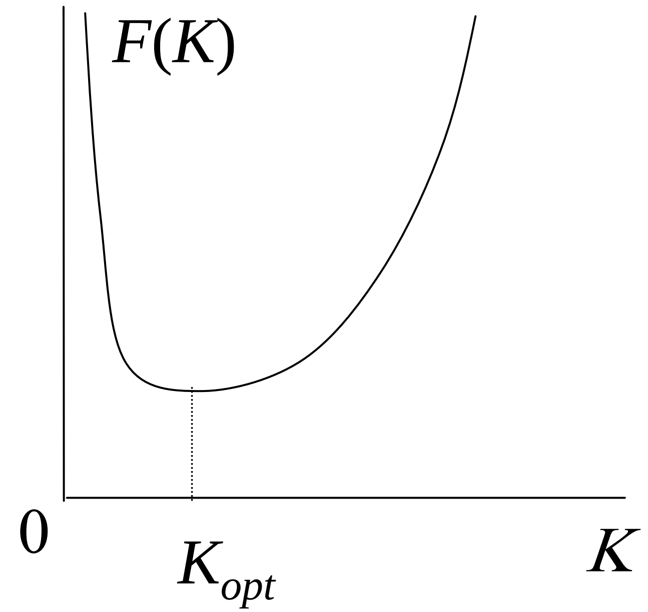
Selection of Optimal *K* (or *s*)

$$F(K) = PN + P^2K^2 + PM + PMN/K.$$

$$F'(K) = 2P^2K - PMN/K^2 = 0.$$

$$K_{opt} = \left(\frac{MN}{2P}\right)^{1/3} = O\left(\left(\frac{MN}{P}\right)^{1/3}\right).$$

$$s_{opt} = \frac{N}{K_{opt}} = \left(\frac{2PN^2}{M}\right)^{1/3} = O\left(\frac{PN^2}{M}\right)^{1/3}.$$



Complexity of Optimized SLFMM

$$F(K_{opt}) = PN + P^2\left(\frac{MN}{2P}\right)^{2/3} + PM + PMN\left(\frac{MN}{2P}\right)^{-1/3}$$
$$= P(M+N) + (MN)^{2/3}O(P^{4/3}).$$

At $K = K_{opt}$, and $M = O(N)$, the complexity of SLFMM is:

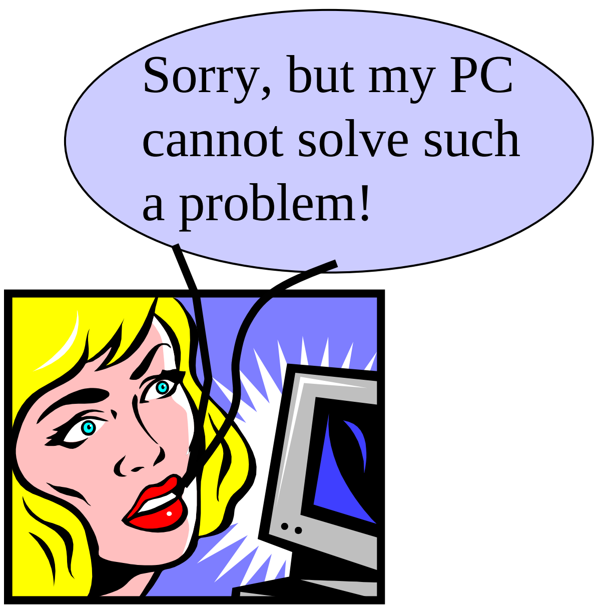
$$O(PN + P^{4/3}N^{4/3}) = O(P^{4/3}N^{4/3}).$$

Example of Complexity:

$P = 10,\ N = 10^5$

Straightforward $O(N^2)$: Complexity ~ 10^{10}
SLFMM $O((PN)^{4/3})$: Complexity ~ 10^8

100 Times CPU savings !



$P = 10,\ N = 10^8$

Straightforward $O(N^2)$: Complexity ~ 10^{16}
SLFMM $O((PN)^{4/3})$: Complexity ~ 10^{12}

10000 Times CPU savings !