Outline

- Norm of the translation operator
- Example of S|R-translation
- Alibi and Alias points of view
- Elements of Group Theory
- 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
S|R-operator has almost the same properties as S|S and R|R

(t cannot be zero)

\[ \Phi(y) = B(x_*) \circ S(y - x_*) , \]

\[ \Phi(y + t) = \tilde{A}(x_*, t) \circ R(y - x_*) \]

\[ \Phi(y) = \tilde{A}(x_*, t) \circ R(y - x_* - t) . \]

\[ \tilde{A}(x_*, t) = (S|R)(t)B(x_*) . \]
Picture is different...

Original expansion
Is valid only here!

\[ |y - x_* - t| < r_1 = |t| - r \]

Since
\[ \Omega_{r_1}(x_* + t) \subset \Omega_r(t) ! \]

Also
\[ |x_i - x_*| < r \]
	singular point!
Example from previous lectures

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

\[ |y - x_*| < |x_i - x_*| : \]
\[ \Phi(y, x_i) = \sum_{m=0}^{\infty} a_m(x_i, x_*) R_m(y - x_*), \]
\[ a_m(x_i, x_*) = -(x_i - x_*)^{-m-1}, \quad m = 0, 1, \ldots, \]
\[ R_m(y - x_*) = (y - x_*)^m, \quad m = 0, 1, \ldots \]

\[ |y - x_*| > |x_i - x_*| : \]
\[ \Phi(y, x_i) = \sum_{m=0}^{\infty} b_m(x_i, x_*) S_m(y - x_*), \]
\[ b_m(x_i, x_*) = (x_i - x_*)^m, \quad m = 0, 1, \ldots, \]
\[ S_m(y - x_*) = (y - x_*)^{-m-1}, \quad m = 0, 1, \ldots \]
In this case we have

\[(|y - x_{*}| < |t|)\]

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

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

So

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

\[(S|R)(t) = \begin{pmatrix}
      t^{-1} & t^{-2} & t^{-3} & 
      -t^{-2} & -2t^{-3} & -3t^{-4} & 
      t^{-3} & 3t^{-4} & 6t^{-5} & 
      \vdots & \vdots & \vdots & \vdots
    \end{pmatrix}\]
Norm of the Translation Operator

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

**Proof.**

\[
\| \mathcal{T}(t) \| = \frac{\| \mathcal{T}(t) \Phi(y) \|}{\| \Phi(y) \|} = \frac{\| \Phi(y + t) \|}{\| \Phi(y) \|} = \frac{\sup_{y \in \mathbb{R}^d} |\Phi(y + t)|}{\sup_{y \in \mathbb{R}^d} |\Phi(y)|} = 1.
\]
Alibi and Alias points of view on translation operator

``Active” or “Alibi” point of view:
Operator transforms functions (vectors).
The reference frame does not change.

``Passive” or “Alias” point of view:
Functions (vectors) do not change.
Operator transforms the reference frame.
Norms of $R|R$, $S|S$, and $S|R$-operators (1)

$\Phi(y)$ is bounded in $\Omega$.

$\Omega' \subset \Omega$.

Therefore $\Phi(y)$ is bounded in $\Omega'$, and

$$\|\Phi(y)\|_{\Omega'} = \sup_{y \in \Omega'} |\Phi(y)| \leq \sup_{y \in \Omega} |\Phi(y)| = \|\Phi(y)\|_{\Omega}.$$
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 \textbf{PLUS} they shrink the domain, where the function is bounded, then their norms do not exceed 1.

\[ \Omega' \subset \Omega \]

\[
\| (R|R)(t) \| = \frac{\sup_{y \in \Omega'} |\Phi(y)|}{\sup_{y \in \Omega} |\Phi(y)|} \leq 1, \\
\| (S|S)(t) \| = \frac{\sup_{y \in \Omega'} |\Phi(y)|}{\sup_{y \in \Omega} |\Phi(y)|} \leq 1, \\
\| (S|R)(t) \| = \frac{\sup_{y \in \Omega'} |\Phi(y)|}{\sup_{y \in \Omega} |\Phi(y)|} \leq 1. 
\]

This is the difference between general translation operator and $R|R$, $S|S$, and $S|R$ operators.
Error of exact $R\mid R$, $S\mid S$, and $S\mid R$-translation

If

$$\|\Phi(y) - \Phi^p(y)\| < \varepsilon,$$

then

$$\|(R|R)(t)(\Phi(y) - \Phi^p(y))\| = \|(R|R)(t)\|\|\Phi(y) - \Phi^p(y)\| < \varepsilon,$$

$$\|(S|S)(t)(\Phi(y) - \Phi^p(y))\| = \|(S|S)(t)\|\|\Phi(y) - \Phi^p(y)\| < \varepsilon,$$

$$\|(S|R)(t)(\Phi(y) - \Phi^p(y))\| = \|(S|R)(t)\|\|\Phi(y) - \Phi^p(y)\| < \varepsilon.$$
Five Key Stones of FMM

• Factorization
• Error
• Translation
• Grouping
• Data Structure
Elements of Group Theory

A set \( G \) of elements (objects) \( a, b, c, \ldots \) is called group if there defined some binary operation \( \odot \) called "group operation", which for any pair of elements \( a, b \in G \) correspond some object \( a \odot b \), such that for any \( a, b, c \in G \) satisfies the following properties:

1). \( a \odot b \in G \), \ (G \text{ is closed with respect to } \odot),

2). \( a \odot (b \odot c) = (a \odot b) \odot c \), \ (associativity),

3). \( \exists e \in G, \ e \odot a = a \), \ (G \text{ contains the unity, } e ),

4). \( \exists a^{-1} \in G, \ a^{-1} \odot a = e \), \ (G \text{ contains the inverse element for each } a \in G \).
Examples of Groups

- Group of Euclidean Motions in 3D: $E(3)$;
- Group of Orthogonal Transforms in 3D: $O(3)$;
- Special Group of Orthogonal Transforms in 3D (includes only "pure rotations"): $SO(3)$;
- Special Group of Euclidean Motions in 3D: $SE(3)$;
- Group of Translations in 3D: $T(3)$. 
Subgroups

A subgroup is a subset of the group, which is also a group with respect to the group operation.

\[ E(3) = O(3) \cup T(3), \]
\[ SE(3) = SO(3) \cup T(3), \]
\[ SO(3) \subset O(3) \subset E(3), \]
\[ SO(3) \subset SE(3) \subset E(3), \]
\[ T(3) \subset SE(3) \subset E(3). \]
Homomorphism

A homomorphism is mapping from one group into another $F : G \to F$, such that

$$F(f_1 \circ f_2) = F(f_1) \circ F(f_2), \quad f_1, f_2 \in G, \quad F(f_1), F(f_2) \in F.$$ 

Example:

In the space of functions we have the Translation Operator, which is due to translations in the Euclidean Space.
**Summary of formal requirements for functions that can be used in FMM**

- We have two sets of points:
  \[ X = \{x_1, x_2, \ldots, x_N\}, \quad x_i \in \mathbb{R}^d, \quad i = 1, \ldots, N, \]
  \[ Y = \{y_1, y_2, \ldots, y_M\}, \quad y_j \in \mathbb{R}^d, \quad j = 1, \ldots, M. \]

- We have functions (potentials):
  \[ \Phi(x_i, y) : \mathbb{R}^d \to \mathbb{R}, \quad y \in \mathbb{R}^d, \quad i = 1, \ldots, N. \]

- These functions can be factorized as (local expansion):
  \[ \Phi(x_i, y) = A(x_i, x_*) \circ R(y - x_*), \quad |y - x_*| < r < |x_i - x_*|, \quad i = 1, \ldots, N. \]

- These functions can be factorized as (far field expansion):
  \[ \Phi(x_i, y) = B(x_i, x_*) \circ S(x - x_*), \quad |y - x_*| > R > |x_i - x_*|, \quad i = 1, \ldots, N. \]

- The product is distributive operation with respect to addition
  \[ (u_1A_1 + u_2A_2) \circ F = u_1A_1 \circ F + u_2A_2 \circ F, \quad F = S, R. \]
Summary of formal requirements for functions that can be used in FMM (2)

- R-expansion coefficients can be $R|R$-translated:
  \[ |x - x_{*2}| < |x_i - x_{*1} - x_{*1} - x_{*2}| : \]
  \[ A(x_i, x_{*2}) = (R|R)(x_{*2} - x_{*1})A(x_i, x_{*1}) \]

- S-expansion coefficients can be $S|S$-translated:
  \[ |x - x_{*2}| > |x_{*1} - x_{*2}| + |x_i - x_{*1}|, \]
  \[ B(x_i, x_{*2}) = (S|S)(x_{*2} - x_{*1})B(x_i, x_{*1}) \]

- $S$-expansion coefficients can be $S|R$-translated (converted to $R$-expansion coefficients)
  \[ |x - x_{*2}| < |x_{*1} - x_{*2}| + |x_i - x_{*1}|, \]
  \[ A(x_i, x_{*2}) = (S|R)(x_{*2} - x_{*1})B(x_i, x_{*1}) \]

- And we are looking for sums:
  \[ v_j = \sum_{i=1}^{N} u_i \Phi(y_j, x_i), \quad j = 1, ..., M. \]

- Some generalization are possible, say instead of $\Phi(y_j, x_i)$ we can consider $\Phi_i(y_j)$, etc.
Middleman Algorithm

Standard algorithm

Sources

Evaluation Points

\[ N \]

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

Middleman algorithm

Sources

Evaluation Points

\[ N \]

Total number of operations: \( O(N+M) \)
Idea of a Single Level FMM

Standard algorithm

Evaluation Points

Total number of operations: $O(NM)$

SLFMM

Evaluation Points

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

Sources $L$ groups

Sources $K$ groups
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

\[ \Phi_1^{(n)}(y) \quad \Phi_2^{(n)}(y) \quad \Phi_3^{(n)}(y) \]

\[ E_1 \quad E_2 \quad E_3 \]

\[ I_1(n) = n \quad I_2(n) = \{ \text{Neighbors}(n) \} \cup n \]

\[ I_3(n) = \{ \text{All boxes} \} \setminus I_2(n) \]

Boxes with these numbers belong to these spatial domains
Definition of potentials

\[
\Phi_1^{(n)}(y) = \sum_{x_i \in E_1(n)} u_i \Phi(y, x_i),
\]

\[
\Phi_2^{(n)}(y) = \sum_{x_i \in E_2(n)} u_i \Phi(y, x_i),
\]

\[
\Phi_3^{(n)}(y) = \sum_{x_i \in E_3(n)} u_i \Phi(y, x_i),
\]

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

\[
\Phi(y) = \sum_{i=1}^{N} u_i \Phi(y, x_i) = \sum_{x_i \in E_2(n) \cup E_3(n)} u_i \Phi(y, x_i) = \Phi_2^{(n)}(y) + \Phi_3^{(n)}(y),
\]

for arbitrary \( n \).
SLFMM Algorithm

Step 1. Generate S-expansion coefficients for each box

\[ \Phi^{(n)}_{i}(x) = \mathbf{C}^{(n)} \odot \mathbf{S}(x - x_{c}^{(n)}) , \]
\[ \mathbf{C}^{(n)} = \sum_{x_{i} \in E_{1}(n,L)} u_{i} \mathbf{B}(x_{i}, x_{c}^{(n)}) . \]

For \( n \in \text{NonEmptySource} \)

Get \( x_{c}^{(n)} \), the center of the box;
\[ \mathbf{C}^{(n)} = \mathbf{0} ; \]

For \( x_{i} \in E_{1}(n) \)

Get \( \mathbf{B}(x_{i}, x_{c}^{(n)}) \), the S-expansion coefficients near the center of the box;
\[ \mathbf{C}^{(n)} = \mathbf{C}^{(n)} + u_{i} \mathbf{B}(x_{i}, x_{c}^{(n)}) ; \]

End;

End;

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

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

\[
\Phi_3^{(n)}(y) = D^{(n)} \circ R(y - x_c^{(n)}),
\]
\[
D^{(n)} = \sum_{m \in I_3(n)} (S|R)(x_c^{(n)} - x_c^{(m)}) C^{(m)}.
\]

For \( n \in \text{NonEmptyEvaluation} \)

Get \( x_c^{(n)} \), the center of the box;
\[
D^{(n)} = 0;
\]

For \( m \in I_3(n) \)

Get \( x_c^{(m)} \), the center of the box;
\[
D^{(n)} = D^{(n)} + (S|R)(x_c^{(n)} - x_c^{(m)}) C^{(m)};
\]

End;

End;

Implementation can be different!
All we need is to get \( D^{(n)} \).
S|R-translation
SLFMM Algorithm

Step 3. Final Summation

\[ v_j = \Phi(y_j) = \sum_{x_i \in E_2(n)} \Phi(y_j, x_i) + D^{(n)} \circ R(y_j - x_c^{(n)}), \quad y_j \in E_1(n). \]

For \( n \in \text{NonEmptyEvaluation} \)

Get \( x_c^{(n)} \), the center of the box;

For \( y_j \in E_1(n) \)

\[ v_j = D^{(n)} \circ R(y_j - x_c^{(n)}); \]

For \( x_i \in E_2(n) \)

\[ v_j = v_j + \Phi(y_j, x_i); \]

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 $P \times P$ 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:

\[ P = 10, \quad N = 10^5 \]

Straightforward \(O(N^2)\): Complexity \(\sim 10^{10}\)

SLFMM \(O((PN)^{4/3})\): Complexity \(\sim 10^8\)

100 Times CPU savings!

\[ P = 10, \quad N = 10^8 \]

Straightforward \(O(N^2)\): Complexity \(\sim 10^{16}\)

SLFMM \(O((PN)^{4/3})\): Complexity \(\sim 10^{12}\)

100000 Times CPU savings!