FMM CMSC 878R/AMSC 698R

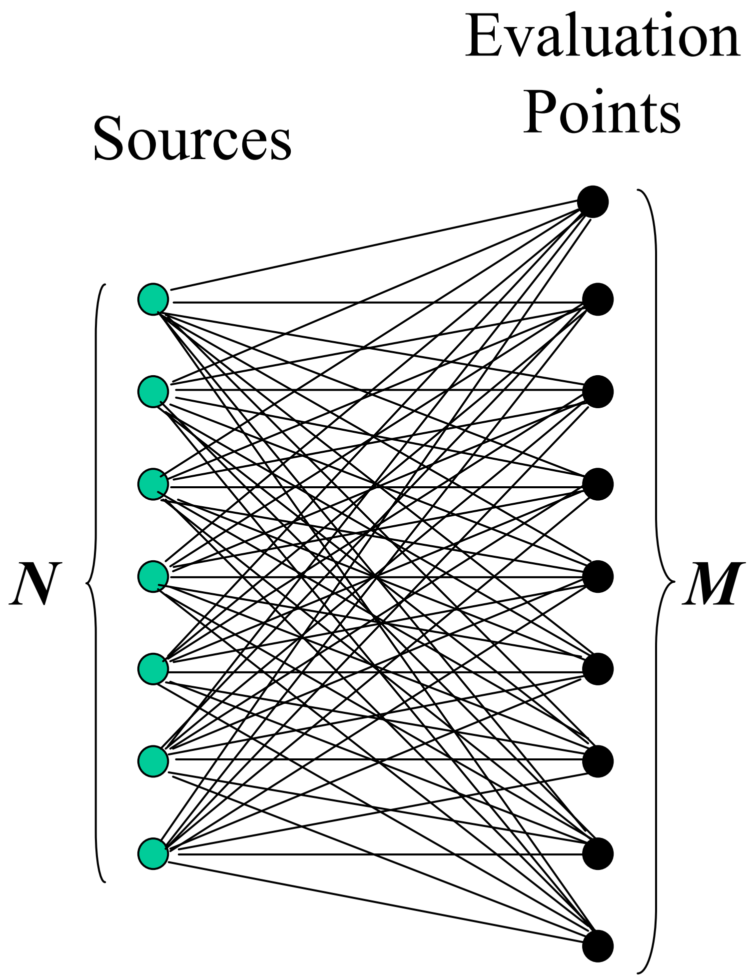
Lecture 10

Progression

- Direct ($O(N^2)$)
  - No data structures
- Middleman: Degenerate Factorizations ($O(N)$)
  - Factorization needed, no data-structure
  - Factorization not always available
- Pre-FMM: S-expansions and R-expansions ($O(N^{3/2})$)
  - Factorizations in terms of near or far-field functions
  - $O(N^{3/2})$ for optimal number of boxes
  - Need boxes to organize source or target sets, and manage those pairs that require direct summation
- SLFMM: S and R-expansions, S|R translation ($O(N^{4/3})$)
- MLFMM: S and R-expansions, S|S, S|R and R|R translations ($O(N \log N)$)

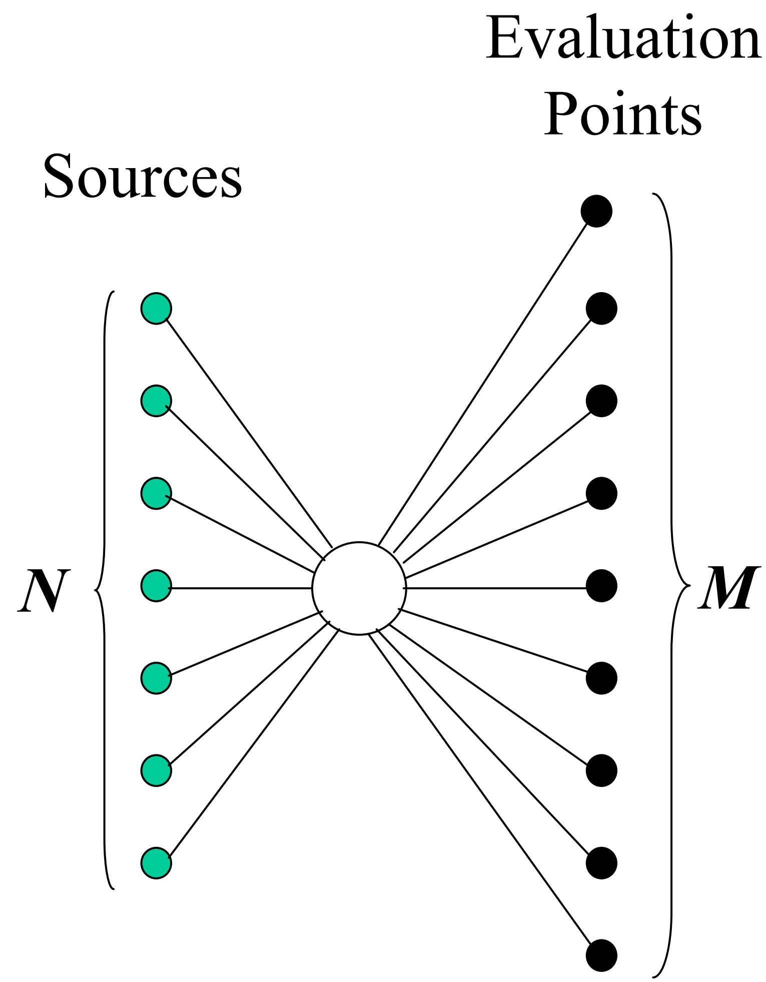
Middleman Algorithm

Standard algorithm



Total number of operations: $O(NM)$

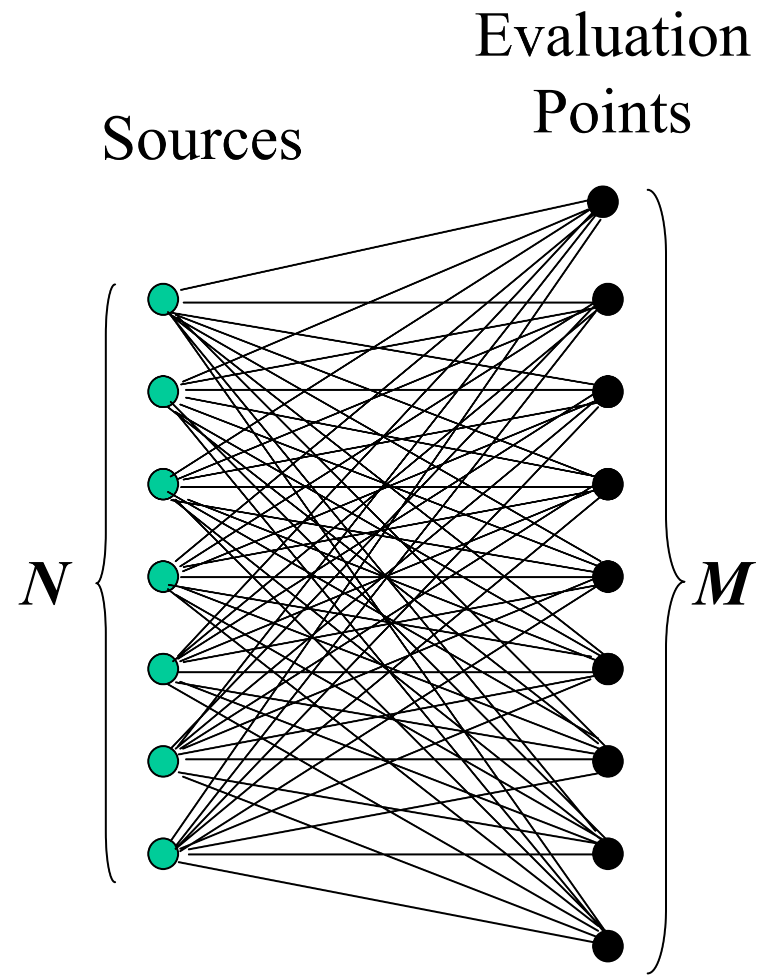
Middleman algorithm



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

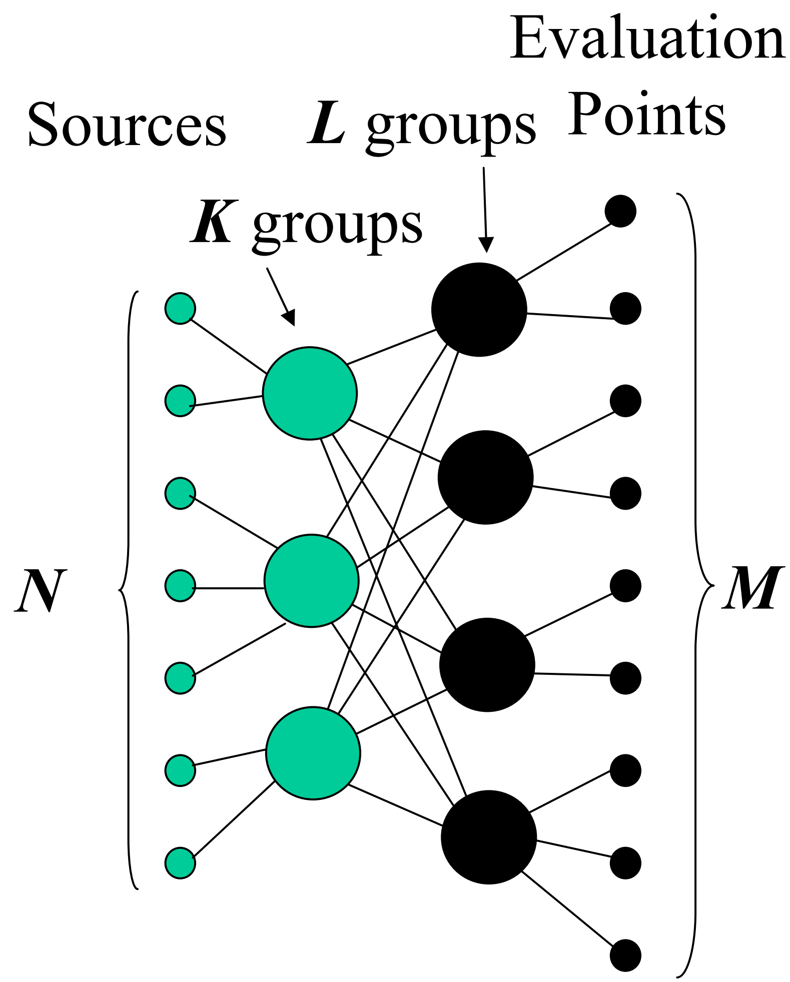
Idea of a Single Level FMM

Standard algorithm



Total number of operations: $O(NM)$

SLFMM



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

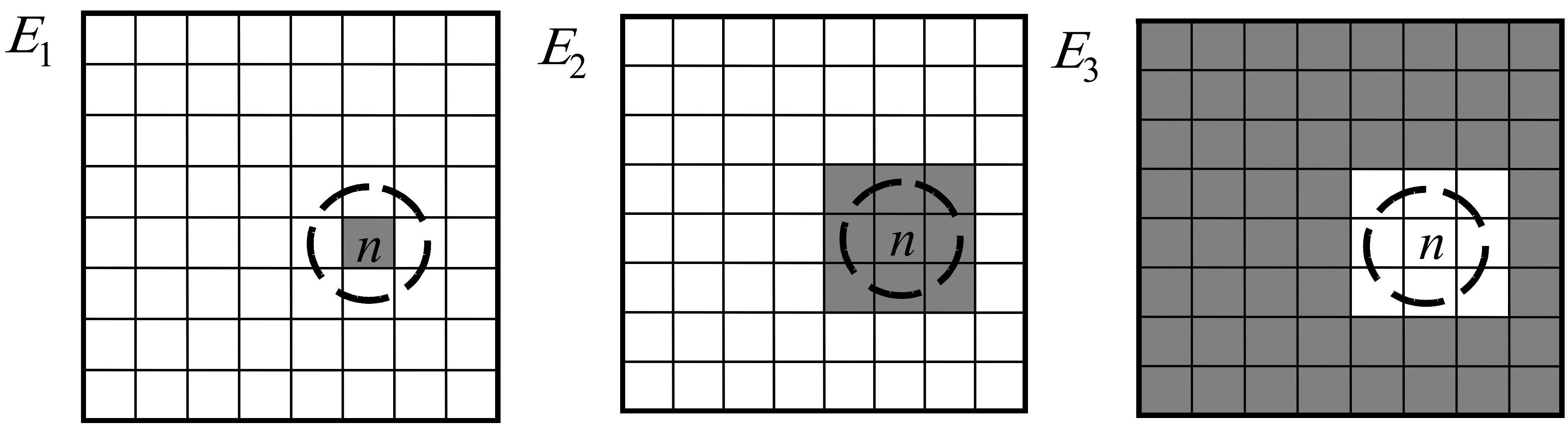
Why do we need SLFMM if Middleman has smaller complexity?

- 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

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

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

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

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

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;

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

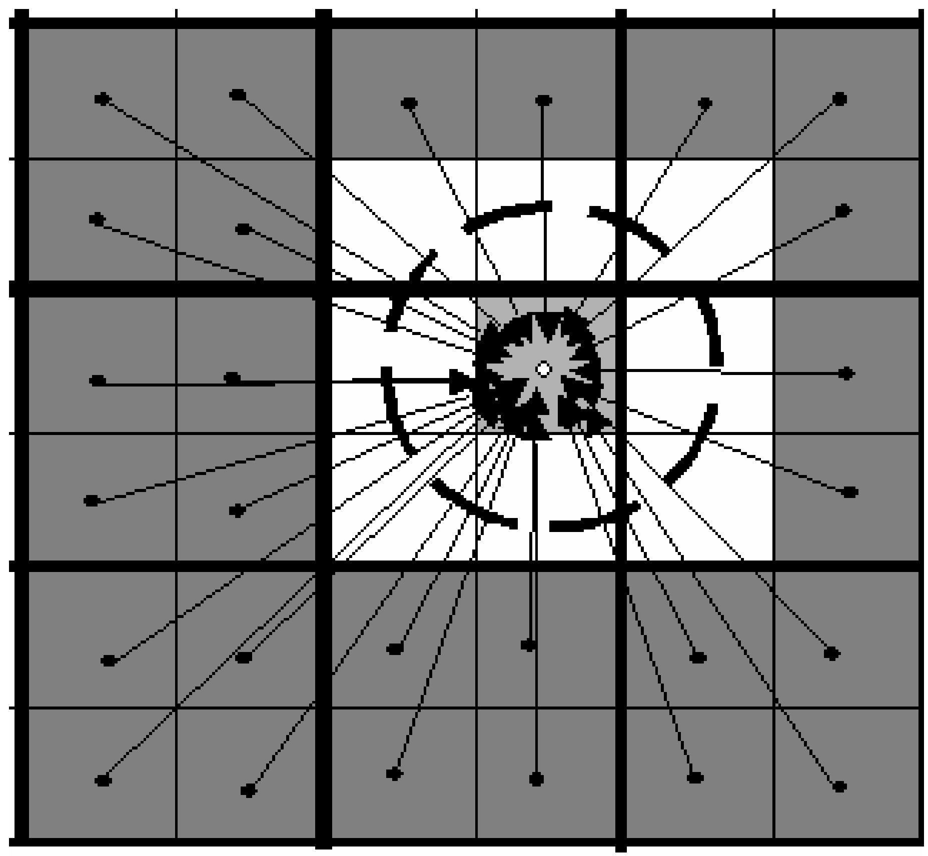
End;

End;

Implementation can be different!

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

S|R-translation



SLFMM Algorithm

Step 3. Final Summation

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

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

- By some magic we can easily find neighbors, and lists of points in each box.
  - Assume that cost of finding these is less than the most expensive operation
  - Magic will come from data-structures
- 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: $O(PN+ P^2K^2 +PM+Ms) =$

  $O(PN+ P^2K^2 +PM+MN/K)$

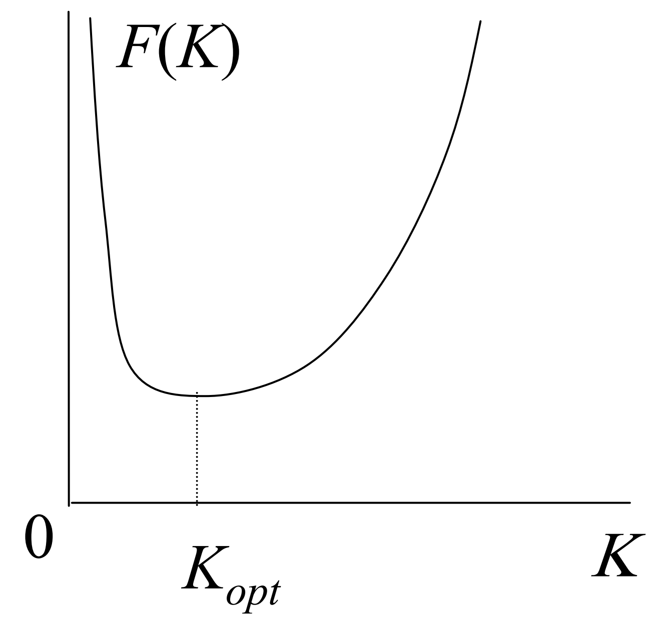
Selection of Optimal K (or s)

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

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

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

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



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:

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

SLFMM Characteristics

- Group source points into clusters
  - Group evaluation points into clusters
- Find the cluster center ($\mathbf{x}_*$) for each cluster
- Find distances from cluster center to points in cluster
- Find distances between clusters
- Build a *S* representation for points in each source cluster

$$\Phi(\mathbf{x}_i, \mathbf{y}) = \sum_{m=1}^{p} b_m(\mathbf{x}_i, \mathbf{x}_*)\, S_m(\mathbf{y}-\mathbf{x}_*)$$

  - Consolidate S series from all $\mathbf{x}_i$ in the cluster
  - For each evaluation box find clusters that are outside min. radius at which *S* expansion converges. Translate *S* series to *R* series
  - Evaluate *R* series at the evaluation points
  - Clean-up by evaluating at points which are inside min. radius

Algorithms needed

- Space division and grouping of points
- Center finding for groups
- Distance between points (group centers)
- Neighbor finding
- Hierarchical division in FMM adds
  - Hierarchical grouping, center finding
  - Finding parents, children, siblings
  - Finding neighbors
- Spatial data-structures
- Also, the structures used to store these relationships require memory
- Determine optimal parameters and “break-even” point

Spatial Data Structures

- Spatial data structures have developed since late 1980s,
  - Aside: one of the founders of the field (Hanan Samet) is local
- FMM developed over the same period
  - So don't use the latest spatial data structure algorithms
- For effective "generalization" we need to use state-of-the-art spatial data structures
- Optimal data structures for FMM is an open research area
- In this course we will mostly use hierarchical division into boxes using 2^d trees
  - Use asymptotically optimal algorithms for performing operations on them
- Next class …

- Constant time methods to
  - number the boxes in a way that can be generated from the coordinate
  - assign points to boxes using their binary coordinates
  - find box centers
  - Find neighboring boxes
- Use bit interleaving and bit shift
- Apply to the general *d*- dimensional case
- Storage is also minimized as most necessary relations are directly generated from point coordinates

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

- We have two sets of points:

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \quad \mathbf{x}_i \in \mathrm{R}^d, \quad i = 1, \ldots, N,$$

$$Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M\}, \quad \mathbf{y}_j \in \mathrm{R}^d, \quad j = 1, \ldots, M.$$

- We have functions (potentials):

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

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

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

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

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

- 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.

Matrix Representation of Linear Operators

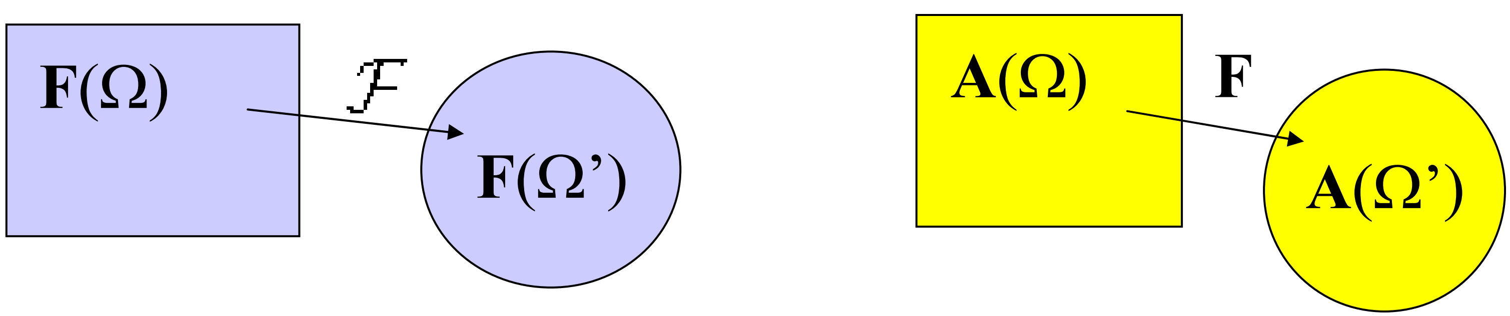
Let $\Omega' \subset \Omega$ and $\mathcal{F}$ is a mapping of $\mathbb{F}(\Omega)$ to $\mathbb{F}(\Omega')$. Such mapping can be considered as action of operator $\mathcal{F}$ on Φ :

$$\mathcal{F}[\Phi(\mathbf{y})] = \widetilde{\Phi(\mathbf{y})}, \quad \Phi(\mathbf{y}) \in \mathbb{F}(\Omega), \quad \widetilde{\Phi(\mathbf{y})} \in \mathbb{F}(\Omega') \subset \mathbb{F}(\Omega)$$

Respectively, operator $\mathcal{F}$ generates operator $\mathbf{F}$ that maps the space of expansion coefficients $\mathbb{A}(\Omega) \to \mathbb{A}(\Omega')$, which can be considered as *representation* of the operator $\mathcal{F}$ in the space of expansion coefficients:

$$\mathbf{FA} = \widetilde{\mathbf{A}}, \quad \mathbf{A} \in \mathbb{A}(\Omega), \quad \widetilde{\mathbf{A}} \in \mathbb{A}(\Omega') \subset \mathbb{A}(\Omega).$$

Inversly, if we introduce any transform of expansion coefficients $\mathbf{FA} = \widetilde{\mathbf{A}}$ which provides uniform convergence of function $\widetilde{\Phi(\mathbf{y})}$ corresponding to these coefficients in $\Omega' \subset \Omega$ then such transform can be treated as operator $\mathcal{F}$ that convert one function from $\mathbb{F}(\Omega)$ to another.



Representation of a Linear Operator

p-Truncation (Projection) Operator

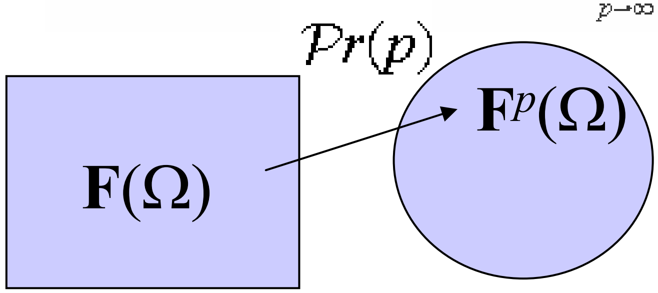
$$\mathbf{Pr}(p)\mathbf{A} = \widetilde{\mathbf{A}}, \quad \mathbf{A} \in \mathbb{A}(\boldsymbol{\Omega}), \quad \widetilde{\mathbf{A}} \in \mathbb{A}^p(\boldsymbol{\Omega}).$$

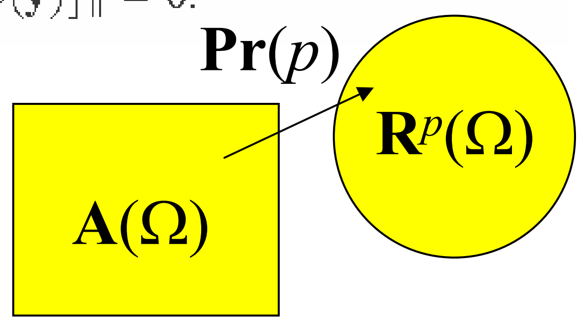
$$\mathbf{A} = \begin{pmatrix} A_0 \\ A_1 \\ \dots \\ A_{p-1} \\ A_p \\ A_{p+1} \\ \dots \end{pmatrix} \rightarrow \widetilde{\mathbf{A}} = \begin{pmatrix} A_0 \\ A_1 \\ \dots \\ A_{p-1} \\ 0 \\ 0 \\ \dots \end{pmatrix}, \qquad \widetilde{\mathbf{A}} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \mathbf{A}$$

In space $\mathbb{F}(\boldsymbol{\Omega})$:

$$\mathcal{P}r(p)[\Phi(\mathbf{y})] = \Phi^p(\mathbf{y}), \quad \Phi(\mathbf{y}) \in \mathbb{F}(\boldsymbol{\Omega}), \quad \Phi^p(\mathbf{y}) \in \mathbb{F}^p(\boldsymbol{\Omega}),$$

$$\lim_{p\to\infty} \| \Phi(\mathbf{y}) - \mathcal{P}r(p)[\Phi(\mathbf{y})] \| = 0.$$

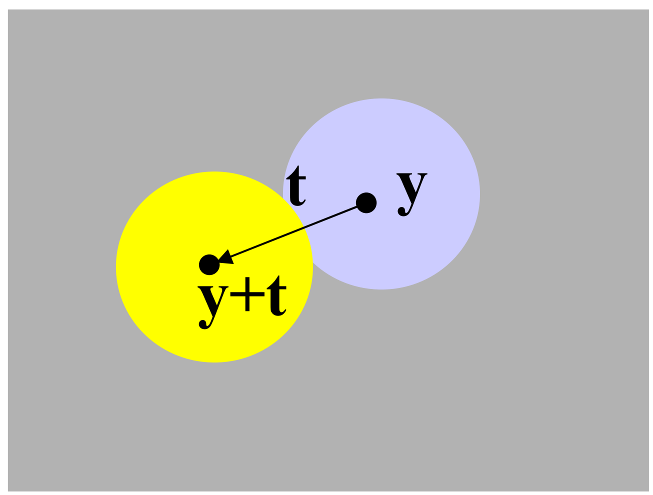




Translation Operator

Operator $\mathcal{T}(\mathbf{t}) : \mathbb{F}(\Omega) \to \mathbb{F}(\Omega'), \Omega' \subset \mathbb{R}^d, \quad \Omega \subset \mathbb{R}^d$ is called *translation* operator corresponding to *translation* vector $\mathbf{t}$, if

$$\mathcal{T}(\mathbf{t})[\Phi(\mathbf{y})] = \Phi(\mathbf{y} + \mathbf{t}), \quad (\mathbf{y} \in \Omega, \quad \mathbf{y} + \mathbf{t} \in \Omega').$$



Translation (Passive view point)

- Take a function or equivalently a function expressed as a series expansion and express it in another coordinate system (reference frame)
- Function is the same on the common parts of domains of definition
- but is represented in different forms
  - $\Phi(\mathbf{x}_i,\mathbf{y})$
  - $\sum_m b_m(\mathbf{x}_i,\mathbf{x}_{*1})\, S_m(\mathbf{y}-\mathbf{x}_*)$
  - $\sum_m a_m(\mathbf{x}_i,\mathbf{x}_{*2})\, R_m(\mathbf{y}-\mathbf{x}_{*2})$ with $a_m = \mathcal{T} b_m$

Translation (Active view point)

- In a fixed coordinate system move the vector (or function), and evaluate it at the same point
- $\Phi(\mathbf{x}_i,\mathbf{y}) \rightarrow \Phi(\mathbf{x}_i+\mathbf{t},\mathbf{y})$
- Functions evaluated at the same point are not the same
- Operator transforms the reference frame

R|R-reexpansion

Let $\mathbf{y}-\mathbf{x}_* \in \Omega_r(\mathbf{x}_*) \subset \mathbb{R}^d$, $\Omega_r(\mathbf{x}_*)$: $|\mathbf{y}-\mathbf{x}_*| < r$, and $\{R_n(\mathbf{y}-\mathbf{x}_*)\}$ be a regular basis in $C(\Omega)$. Let $\mathbf{y}-\mathbf{x}_*+\mathbf{t} \in \Omega_r(\mathbf{x}_*)$ and

$$R_n(\mathbf{y}-\mathbf{x}_*+\mathbf{t}) = \sum_{l=0}^{\infty}(R|R)_{ln}(\mathbf{t})R_l(\mathbf{y}-\mathbf{x}_*).$$

Coefficients $(R|R)_{ln}(\mathbf{t})$ are called $R|R-$ *reexpansion coefficients* (regular-to-regular) , and infinite matrix

$$(\mathbf{R}|\mathbf{R})(\mathbf{t}) = \begin{pmatrix} (R|R)_{00} & (R|R)_{01} & \ldots \\ (R|R)_{10} & (R|R)_{11} & \ldots \\ \ldots & \ldots & \ldots \end{pmatrix}$$

is called $R|R-$ *reexpansion matrix*.

Example of R|R-reexpansion

$$R_m(x) = x^m,$$

$$R_m(x+t) = (x+t)^m = x^m + \binom{m}{1} x^{m-1} t + \ldots + \binom{m}{m-1} x t^{m-1} + t^m$$

$$= \sum_{l=0}^{m} \binom{m}{l} t^l x^{m-l} = \sum_{l=0}^{m} \binom{m}{l} t^{m-l} x^l = \sum_{l=0}^{m} \binom{m}{l} t^{m-l} R_l(x),$$

$$(R|R)_{lm}(t) = \begin{cases} \binom{m}{l} t^{m-l}, & l \leqslant m, \\ 0, & l > m. \end{cases}.$$

R|R-translation operator

Translation operator $\mathcal{T}(\mathbf{t})$ which is represented in regular basis $\{R_n(\mathbf{y}-\mathbf{x}_*)\}$ by the $R|R-$ *reexpansion matrix* is called $\mathcal{R}|\mathcal{R}$-translation operator.

$$\mathcal{T}(\mathbf{t})[\Phi(\mathbf{y})] = \Phi(\mathbf{y}+\mathbf{t})$$

$$(\mathcal{R}|\mathcal{R})(\mathbf{t}) = \mathcal{T}(\mathbf{t}).$$

Different translation operators

$$\mathcal{T}(\mathbf{t})[\Phi(\mathbf{y})] = \Phi(\mathbf{y}+\mathbf{t})$$

The first letter shows the basis for $\Phi(\mathbf{y})$

The second letter shows the basis for $\Phi(\mathbf{y}+\mathbf{t})$

$$\mathcal{T}(\mathbf{t}) = \begin{cases} (\mathcal{R}|\mathcal{R})(\mathbf{t}) \\ (\mathcal{S}|\mathcal{S})(\mathbf{t}) \\ (\mathcal{S}|\mathcal{R})(\mathbf{t}) \\ (\mathcal{R}|\mathcal{S})(\mathbf{t}) \end{cases}$$

Needed only to show the expansion basis (for operator representation)

Matrix representation of R|R-translation operator

Consider $\Phi(\mathbf{y}) = \sum_{n=0}^{\infty} A_n(\mathbf{x}_*) R_n(\mathbf{y} - \mathbf{x}_*)$.

$$\Phi(\mathbf{y}+\mathbf{t}) = (\mathcal{R}|\mathcal{R})(\mathbf{t})[\Phi(\mathbf{y})] = \sum_{n=0}^{\infty} A_n(\mathbf{x}_*)(\mathcal{R}|\mathcal{R})(\mathbf{t})[R_n(\mathbf{y}-\mathbf{x}_*)]$$

$$= \sum_{n=0}^{\infty} A_n(\mathbf{x}_*) R_n(\mathbf{y}-\mathbf{x}_*+\mathbf{t})$$

$$= \sum_{n=0}^{\infty} A_n(\mathbf{x}_*) \sum_{l=0}^{\infty} (R|R)_{ln}(\mathbf{t}) R_l(\mathbf{y}-\mathbf{x}_*)$$

$$= \sum_{l=0}^{\infty} \left[\sum_{n=0}^{\infty} (R|R)_{ln}(\mathbf{t}) A_n(\mathbf{x}_*) \right] R_l(\mathbf{y}-\mathbf{x}_*)$$

$$= \sum_{l=0}^{\infty} \widetilde{A}_l(\mathbf{x}_*, \mathbf{t}) R_l(\mathbf{y}-\mathbf{x}_*),$$

Coefficients of shifted function

Coefficients of original function

$$\widetilde{A}_l(\mathbf{x}_*, \mathbf{t}) = \sum_{n=0}^{\infty} (R|R)_{ln}(\mathbf{t}) A_n(\mathbf{x}_*), \quad \widetilde{\mathbf{A}}(\mathbf{x}_*, \mathbf{t}) = (\mathbf{R}|\mathbf{R})(\mathbf{t}) \mathbf{A}(\mathbf{x}_*).$$

Reexpansion of the same function over shifted basis

Compact notation:

$$\Phi(\mathbf{y}) = \sum_{n=0}^{\infty} A_n(\mathbf{x}_*) R_n(\mathbf{y}-\mathbf{x}_*) = \mathbf{A}(\mathbf{x}_*) \circ \mathbf{R}(\mathbf{y}-\mathbf{x}_*),$$

$$\Phi(\mathbf{y}+\mathbf{t}) = \sum_{l=0}^{\infty} \widetilde{A}_l(\mathbf{x}_*,\mathbf{t}) R_l(\mathbf{y}-\mathbf{x}_*) = \widetilde{\mathbf{A}}(\mathbf{x}_*,\mathbf{t}) \circ \mathbf{R}(\mathbf{y}-\mathbf{x}_*)$$

We have:

$$\Phi(\mathbf{y}) = \Phi((\mathbf{y}-\mathbf{t})+\mathbf{t}) = \widetilde{\mathbf{A}}(\mathbf{x}_*,\mathbf{t}) \circ \mathbf{R}((\mathbf{y}-\mathbf{t})-\mathbf{x}_*)$$
$$= \widetilde{\mathbf{A}}(\mathbf{x}_*,\mathbf{t}) \circ \mathbf{R}(\mathbf{y}-\mathbf{x}_*-\mathbf{t}).$$

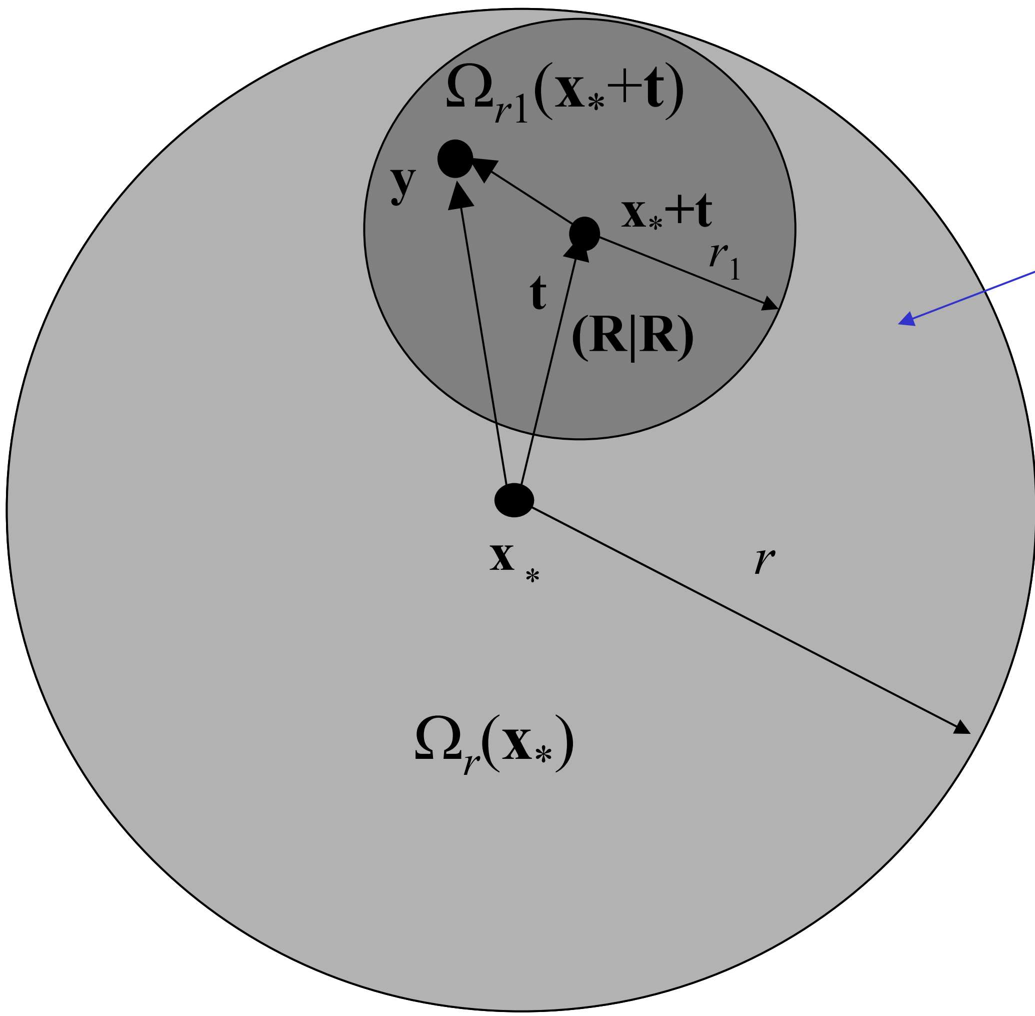
Also

$$\Phi(\mathbf{y}) = \mathbf{A}(\mathbf{x}_*) \circ \mathbf{R}(\mathbf{y}-\mathbf{x}_*) = \mathbf{A}(\mathbf{x}_*+\mathbf{t}) \circ \mathbf{R}(\mathbf{y}-\mathbf{x}_*-\mathbf{t}),$$

so

$$\mathbf{A}(\mathbf{x}_*+\mathbf{t}) = \widetilde{\mathbf{A}}(\mathbf{x}_*,\mathbf{t}) = (\mathbf{R}|\mathbf{R})(\mathbf{t})\mathbf{A}(\mathbf{x}_*).$$

R|R-reexpansion of the same function over shifted basis (2)



Original expansion
Is valid only here!

$$|\mathbf{y} - \mathbf{x}_* - \mathbf{t}| < r_1 = r - |\mathbf{t}|$$

Since $\Omega_{r1}(\mathbf{x}_*+\mathbf{t}) \subset \Omega_r(\mathbf{t})$!

Example of power series reexpansion

$$R_m(x) = x^m$$

$$\Phi(y,x_i) = \sum_{m=0}^{\infty} A_m(x_{*1},x_i) R_m(y - x_{*1}) = \sum_{m=0}^{\infty} A_m(x_{*2},x_i) R_m(y - x_{*2}),$$

$$\mathbf{A}(x_{*2},x_i) = (\mathbf{R}|\mathbf{R})(x_{*2} - x_{*1}) \bullet \mathbf{A}(x_{*1},x_i).$$

$$\begin{pmatrix} A_0(x_{*2},x_i) \\ A_1(x_{*2},x_i) \\ A_2(x_{*2},x_i) \\ \cdots \end{pmatrix} = \begin{pmatrix} 1 & \binom{1}{0}(x_{*2}-x_{*1}) & \binom{2}{0}(x_{*2}-x_{*1})^2 & \cdots \\ 0 & 1 & \binom{2}{1}(x_{*2}-x_{*1}) & \cdots \\ 0 & 0 & 1 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix} \begin{pmatrix} A_0(x_{*1},x_i) \\ A_1(x_{*1},x_i) \\ A_2(x_{*1},x_i) \\ \cdots \end{pmatrix}$$

Example of power series reexpansion (2). Relation to Taylor series.

Let's check this for Taylor series, when expansion coefficients are

$$A_m(x_{*1}, x_i) = \frac{1}{m!} \frac{\partial^m \Phi(x_{*1}, x_i)}{\partial x_{*1}^m}$$

For A_0 this yields Taylor series again!

$$\Phi(x_{*2}, x_i) = \sum_{m=0}^{\infty} \frac{1}{m!} \frac{\partial^m \Phi(x_{*1}, x_i)}{\partial x_{*1}^m} (x_{*2} - x_{*1})^m,$$

Check for A_l

$$\frac{1}{l!} \frac{\partial^l \Phi(x_{*2}, x_i)}{\partial x_{*2}^l} = \sum_{m=l}^{\infty} \binom{m}{l} \frac{1}{m!} \frac{\partial^m \Phi(x_{*1}, x_i)}{\partial x_{*1}^m} (x_{*2} - x_{*1})^{m-l}$$

$$= \sum_{m=l}^{\infty} \frac{m!}{l!(m-l)!} \frac{1}{m!} \frac{\partial^m \Phi(x_{*1}, x_i)}{\partial x_{*1}^m} (x_{*2} - x_{*1})^{m-l}$$

$$= \frac{1}{l!} \sum_{k=0}^{\infty} \frac{1}{k!} \frac{\partial^{k+l} \Phi(x_{*1}, x_i)}{\partial x_{*1}^{k+l}} (x_{*2} - x_{*1})^k,$$

$$\frac{\partial^l \Phi(x_{*2}, x_i)}{\partial x_{*2}^l} = \sum_{k=0}^{\infty} \frac{1}{k!} \frac{\partial^{k+l} \Phi(x_{*1}, x_i)}{\partial x_{*1}^{k+l}} (x_{*2} - x_{*1})^k.$$

For A_l we obtained Taylor series for the *l*-th derivative!

S|S-reexpansion

Let $\mathbf{y}-\mathbf{x}_* \in \Omega_r(\mathbf{x}_*) \subset \mathbb{R}^d$, $\quad \Omega_r(\mathbf{x}_*) : |\mathbf{y}-\mathbf{x}_*| > r$, and $\{S_n(\mathbf{y}-\mathbf{x}_*)\}$ be a singular basis in $C(\Omega)$. Let $\mathbf{y}-\mathbf{x}_*+\mathbf{t} \in \Omega_r(\mathbf{x}_*)$ and

$$S_n(\mathbf{y}-\mathbf{x}_*+\mathbf{t}) = \sum_{l=0}^{\infty} (S|S)_{ln}(\mathbf{t}) S_l(\mathbf{y}-\mathbf{x}_*).$$

Coefficients $(S|S)_{ln}(\mathbf{t})$ are called $S|S-$ *reexpansion coefficients* (singular-to-singular) , and infinite matrix

$$(\mathbf{S}|\mathbf{S})(\mathbf{t}) = \begin{pmatrix} (S|S)_{00} & (S|S)_{01} & \dots \\ (S|S)_{10} & (S|S)_{11} & \dots \\ \dots & \dots & \dots \end{pmatrix}$$

is called $S|S-$ *reexpansion matrix*.

S|S-translation operator

Translation operator $\mathcal{T}(\mathbf{t})$ which is represented in singular basis $\{S_n(\mathbf{y}-\mathbf{x}_*)\}$ by the $S|S-$ *reexpansion matrix* is called $\mathcal{S}|\mathcal{S}$-translation operator.

$$\mathcal{T}(\mathbf{t})[\Phi(\mathbf{y})] = \Phi(\mathbf{y}+\mathbf{t})$$

$$(\mathcal{S}|\mathcal{S})(\mathbf{t}) = \mathcal{T}(\mathbf{t}).$$

S|R-translation operator

Translation operator $\mathcal{T}(\mathbf{t})$ which is represented in singular basis by the $S|R-$ *reexpansion matrix* is called $\mathcal{S}|\mathcal{R}$-translation operator if the basis of expansion is changed with the translation operation from singular $\{S_n(\mathbf{y}-\mathbf{x}_*)\}$ to regular $\{R_n(\mathbf{y}-\mathbf{x}_*+\mathbf{t})\}$

$$\mathcal{T}(\mathbf{t})[\Phi(\mathbf{y})] = \Phi(\mathbf{y}+\mathbf{t})$$

$$(\mathcal{S}|\mathcal{R})(\mathbf{t}) = \mathcal{T}(\mathbf{t}).$$

S|R-operator has almost the same properties as S|S and R|R

(**t** cannot be zero)

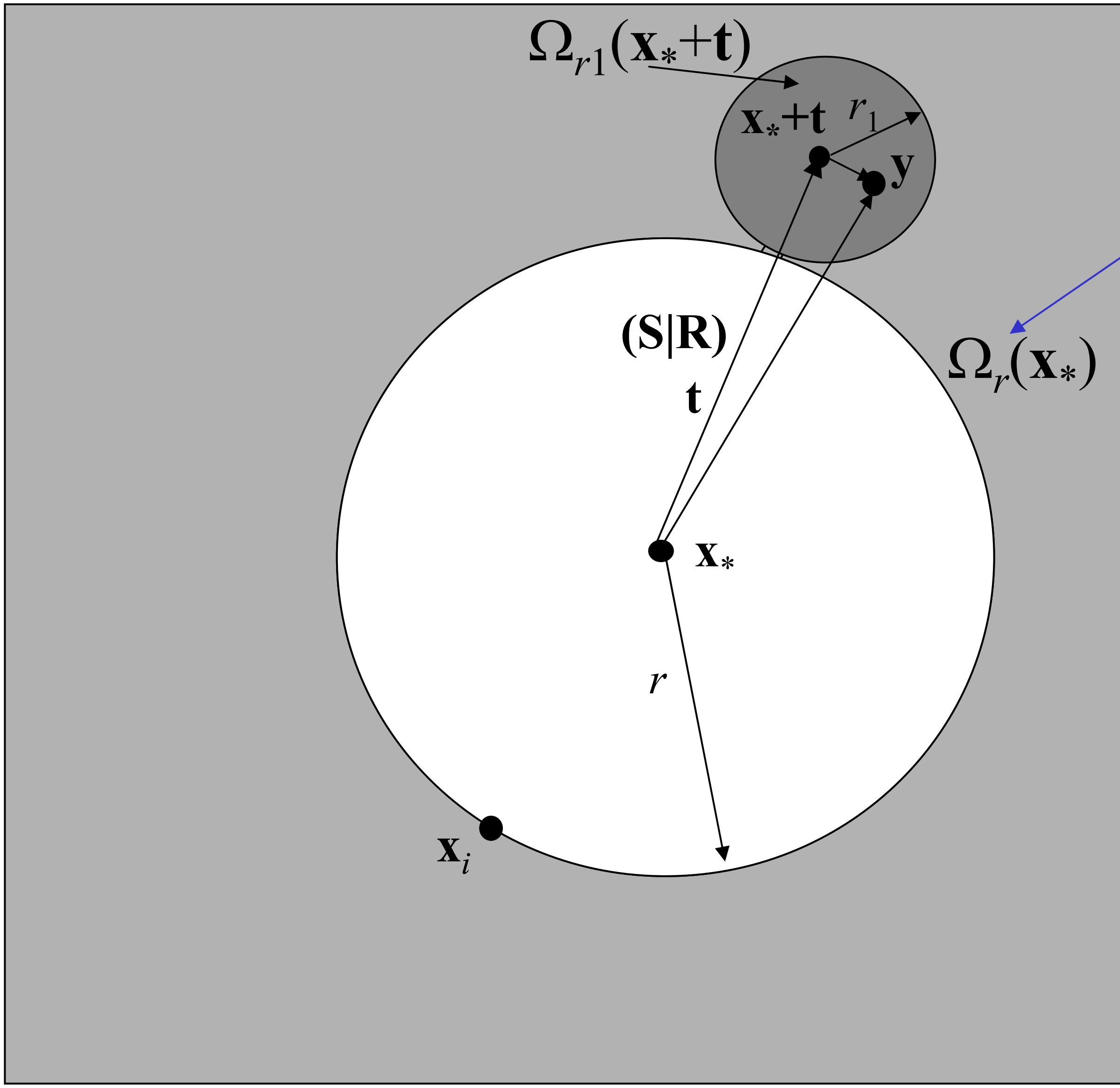
$$\Phi(\mathbf{y}) = B(\mathbf{x}_*) \circ \mathbf{S}(\mathbf{y} - \mathbf{x}_*),$$

$$\Phi(\mathbf{y} + \mathbf{t}) = \widetilde{A}(\mathbf{x}_*, \mathbf{t}) \circ R(\mathbf{y} - \mathbf{x}_*)$$

$$\Phi(\mathbf{y}) = \widetilde{A}(\mathbf{x}_*, \mathbf{t}) \circ R(\mathbf{y} - \mathbf{x}_* - \mathbf{t}).$$

$$\widetilde{A}(\mathbf{x}_*, \mathbf{t}) = (\mathbf{S}|R)(\mathbf{t})B(\mathbf{x}_*).$$

Picture is different…



Original expansion
Is valid only here!

$|\mathbf{y} - \mathbf{x}_* - \mathbf{t}| < r_1 = |\mathbf{t}| - r$

Since
$\Omega_{r1}(\mathbf{x}_*+\mathbf{t}) \subset \Omega_r(\mathbf{t})$!

Also

$|\mathbf{x}_i - \mathbf{x}_*| < r$

singular point !

Example from previous lectures

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

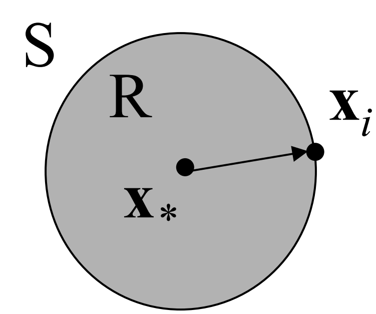
$|y - x_*| < |x_i - x_*|$:

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

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

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

Five Key Stones of FMM

- Factorization
- Error
- Translation
- Grouping
- Data Structure

Complexity of MLFMM

Definitions:

Upward Pass: Going Up on SOURCE Hierarchy

Downward Pass: Going Down on EVALUATION Hierarchy

We have N sources. Let us we group them hierarchically. At level l we have N_l source groups. Each group at level $l + 1$ contains $N_l S$ sources, so

$$N_{l+1} = N_l S, \quad l = 2, 3, \ldots, L,$$

and

$$N_L = N.$$

Then the number of operations for the Upward Pass is of order

$$N_L + N_{L-1} + \ldots + N_2 = N + \frac{N}{S} + \frac{N}{S^2} + \ldots + \frac{N}{S^{L-2}}$$

$$= N\left(1 + \frac{1}{S} + \ldots + \frac{1}{S^{L-2}}\right) = N\frac{1 - 1/S^{L-1}}{1 - 1/S} = O(N).$$

Similarly, the number of operations for the Downward Pass is of order $O(M)$.

$$MLFMM\_Complexity = O(M + N)\ !$$

Summary of requirements for functions that can be used in FMM

- We have two sets of points:

$$\mathsf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \quad \mathbf{x}_i \in \mathsf{R}^d, \quad i = 1, \ldots, N,$$

$$\mathsf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M\}, \quad \mathbf{y}_j \in \mathsf{R}^d, \quad j = 1, \ldots, M.$$

- We have functions (potentials):

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

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

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

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

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

- 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 requirements for functions that can be used in FMM (2)

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

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

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

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

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

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

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

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

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

- 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.

Two Parts of the MLFMM

- Setting Hierarchical Data Structure (MLFMM Constructor) $O(N\log N+M\log M)$

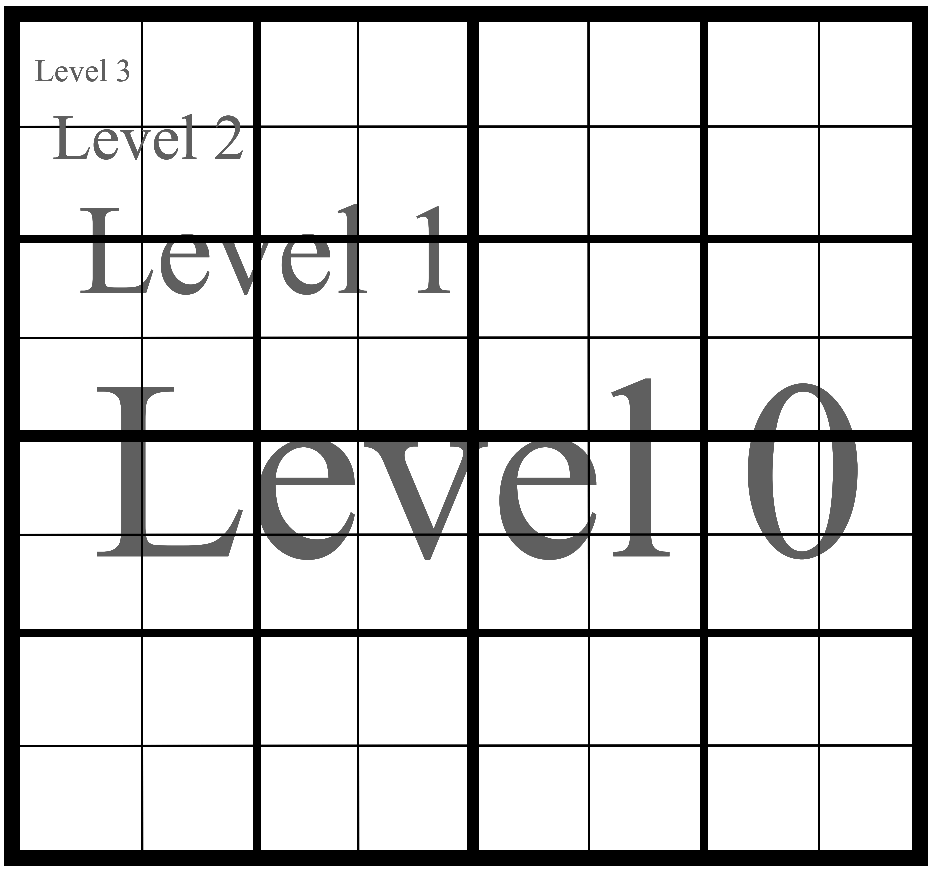
- MLFMM Solver $O(N+M)$ or $O(N\log^q N+M\log^q M)$, *Will evaluate the complexity in more details later.*

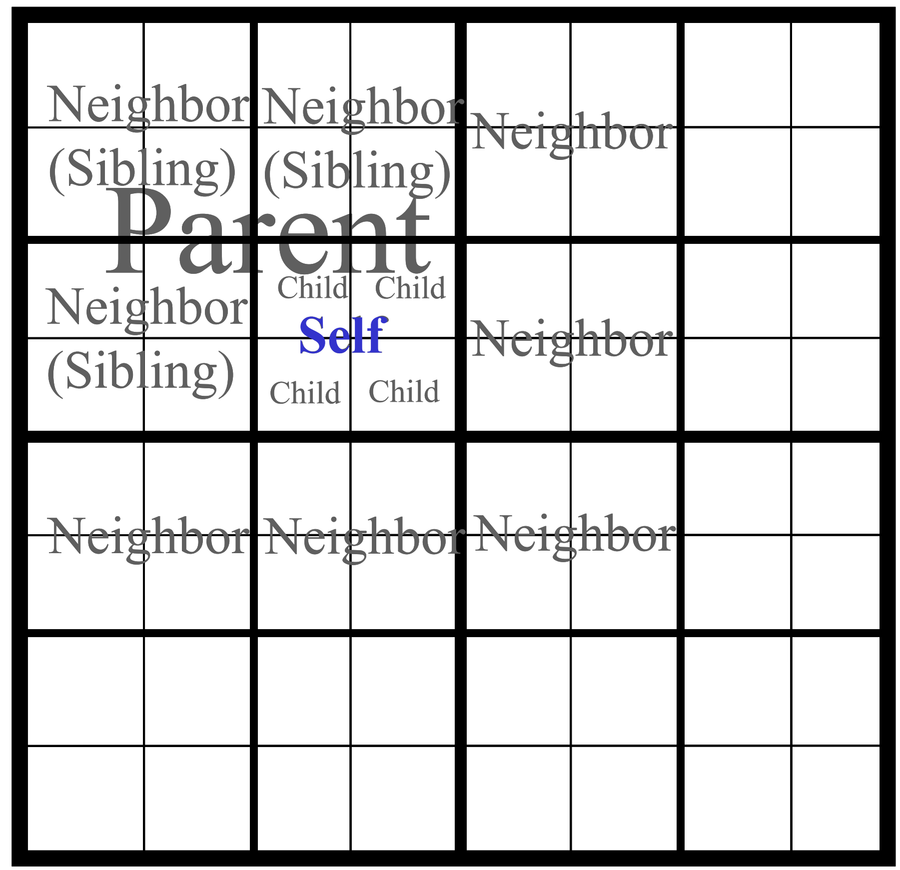
If iterative or multiple solutions of the same system are required, the MLFMM Constructor should be called only once.

Setting Hierarchical Data Structure

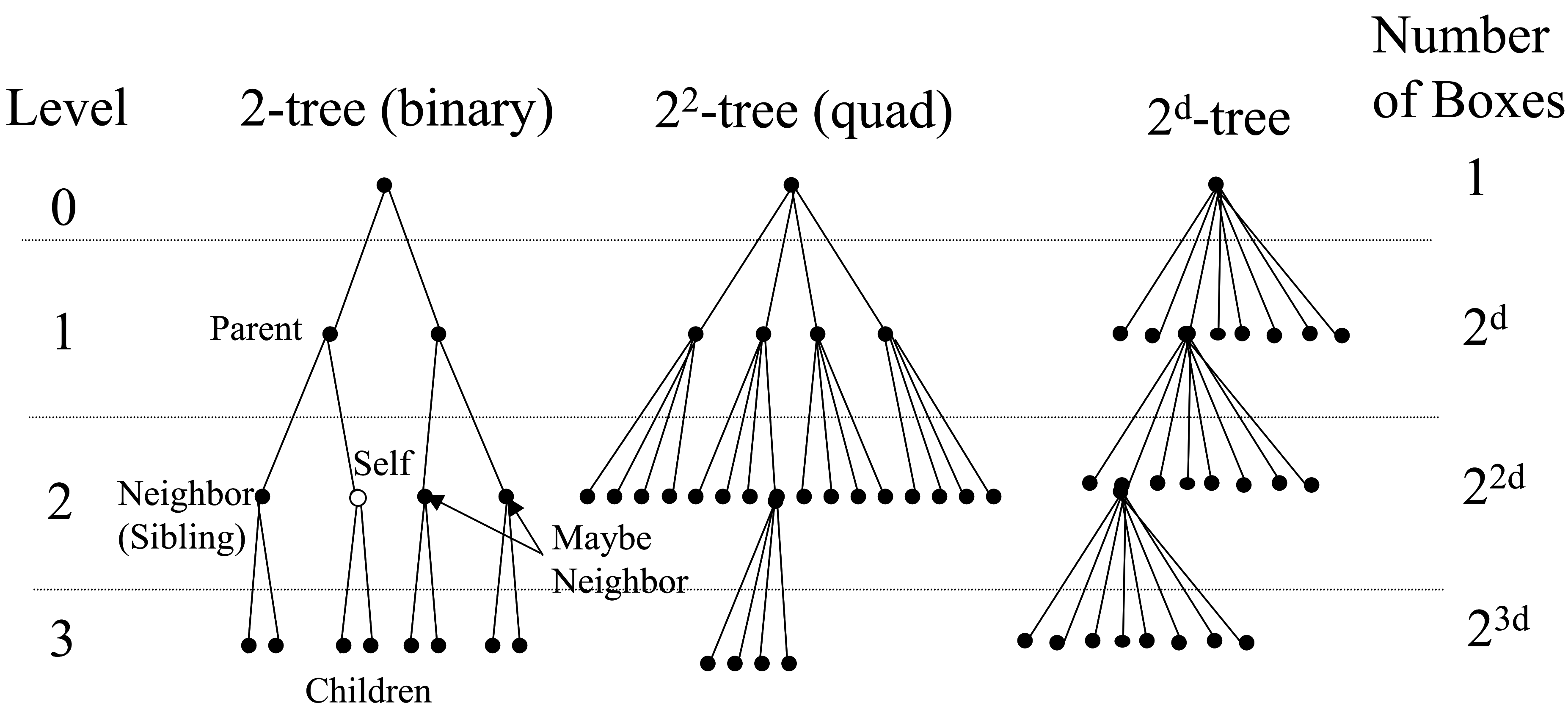
- Scale source and evaluation data to have the computational domain of size of a unit box.
- Sort data (spatially order data) using bit interleaving technique *(Next week)*
- Determine the level of space subdivision with 2^d-tree to have s sources at the finest subdivision level, L_{max} *(Next week)*
- If you choose to spend memory for trees, neighbor lists, and so on, compute and store information that does not change in the process of execution of the MLFMM solver.

Hierarchy in 2^d-tree





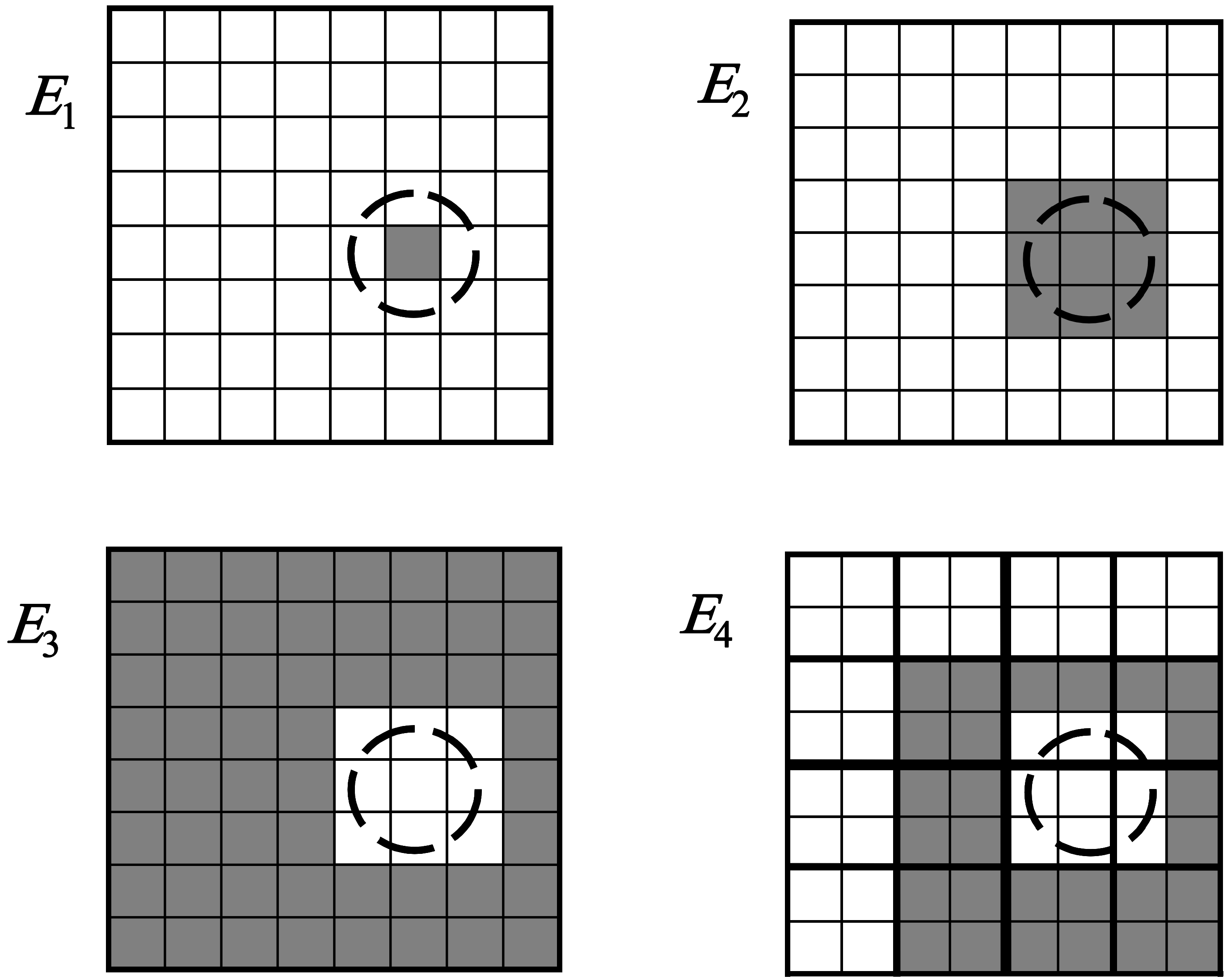
2^d-trees

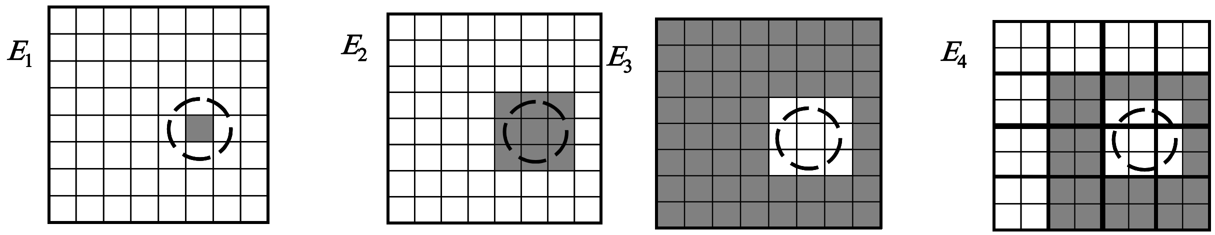

Level
2-tree (binary)
2^2-tree (quad)
2^d-tree
Number of Boxes
0
1
2
3
Parent
Self
Neighbor (Sibling)
Maybe Neighbor
Children
1
2^d
2^{2d}
2^{3d}

Hierarchical Indexing and Functions

- We assign to each box on level l some number (index) n; Global index of any box is (n,l).
- We assume that functions, such as *Parent(n), ChildrenAll(n), Children(*X*;n,l), NeighborsAll(n,l), Neighbors(*X*;n,l),* for given d-dimensional data set, X, are available
    - (will consider their implementation next week).
    - These functions return sets of indexes of boxes at proper levels which are relatives (or neighbors) to the given box (n,l).
- We drop X in many cases, to have shorter notation.

Hierarchical Spatial Domains





Hierarchical Spatial Domains

We accept the hierarchical numbering system described above and define four elements (domains) of fractal structure for each box with number $n = Number = 0, \ldots, 2^{ld} - 1$ at level $l = 0, \ldots, L$.

$E_1(n,l) \subset \mathrm{R}^d$ denotes spatial points *inside* the box (n,l);

$E_2(n,l) \subset \mathrm{R}^d$ denotes spatial points *inside* the box (n,l) and its neighbors, $\{(Neighbor(n,l),l)\}$;

$E_3(n,l) = E_1(0,0) \backslash E_2(n,l)$ denotes spatial points *outside* the box (n,l) and its neighbors, $\{(Neighbor(n,l),l)\}$.

$E_4(n,l) = E_2(Parent(n), l-1) \backslash E_2(n,l)$ denotes spatial points *inside* the parent box $(Parent(n), l-1)$ and its neighbors, $\{(Neighbor(Parent(n), l-1), l-1)\}$, from which the domain $E_2(n,l)$ is excluded.

Accordingly we associate sets of boxes of level l which constitute each domain $E_m(n,l)$. Their numbers we denote as $I_m(n,l)$. So we have:

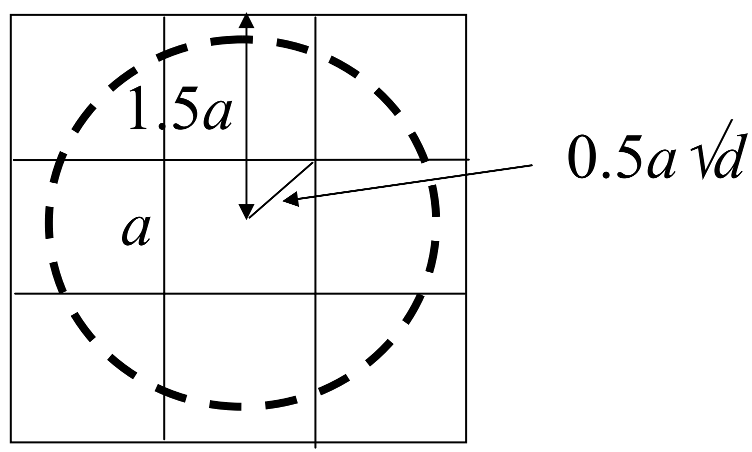
$I_1(n,l) = (n,l)$,

$I_2(n,l) = \{(n,l), (Neighbor(n,l),l)\}$,

$I_3(n,l) = \{0, 1, \ldots, 2^{ld} - 1\} \backslash I_2(n,l)$,

$I_4(n,l) = \{(Children(Neighbor(Parent(n), l-1), l-1))\} \backslash \{(n,l), (Neighbor(n,l),l)\}$.

With Such Neighborhood the dimensionality of space in FMM cannot exceed d=9.



$$0.5a\sqrt{d} < 1.5a,$$
$$\sqrt{d} < 3,$$
$$d < 9.$$

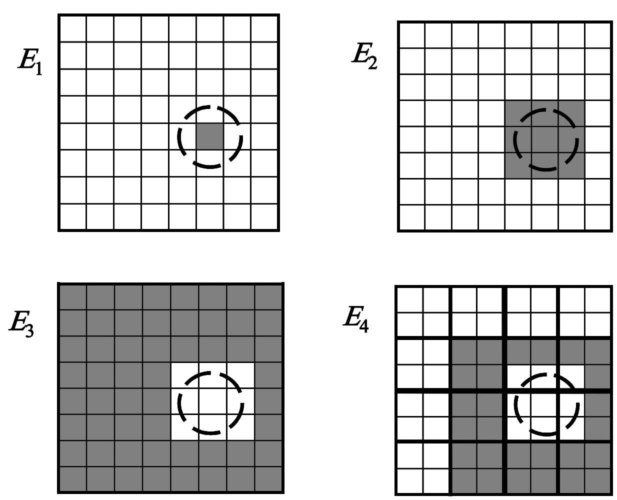
For larger dimensions larger neighborhoods could be considered

However all issues have to be reconsidered … not practical to use 2^d-trees in this case

In fact, we will show later that 1-neighborhoods can be used only for dimensions $d < 4$.

Hierarchical Potentials (Functions)

Based on these domains for each box the following functions (potentials) are defined:



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

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

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

$$\Phi_4^{(n,l)}(\mathbf{y}) = \sum_{\mathbf{x}_i \in E_4(n,l)} u_i \Phi(\mathbf{y}, \mathbf{x}_i),$$

Note that since domains $E_2(n,l)$ and $E_3(n,l)$ are complimentary, and

$$\Phi(\mathbf{y}) = \Phi_2^{(n,l)}(\mathbf{y}) + \Phi_3^{(n,l)}(\mathbf{y})$$

for arbitrary l and n.

The MLFMM Algorithm (Solver)

- ``Build Function” or ``Build Potential” means find its expansion coefficients over some basis;
- The MLFMM Algorithm consists of
  - Upward Pass;
  - Downward Pass;
  - Final Summation;

Upward Pass. Step 1.

Step 1. At the finest level of space subdivision, build far-field expansion for sources inside each non-empty box of set $\mathbb{X}$ near the center of that box $\mathbf{x}_c^{(n,L)}$:

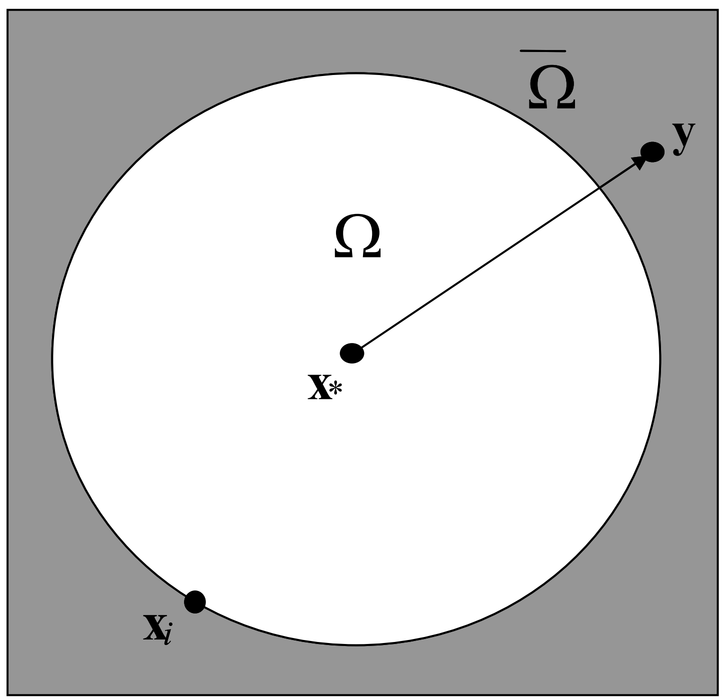
$$\Phi_1^{(n,L)}(\mathbf{y}) = \mathbf{C}^{(n,L)} \circ \mathbf{S}(\mathbf{y} - \mathbf{x}_c^{(n,L)}),$$

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

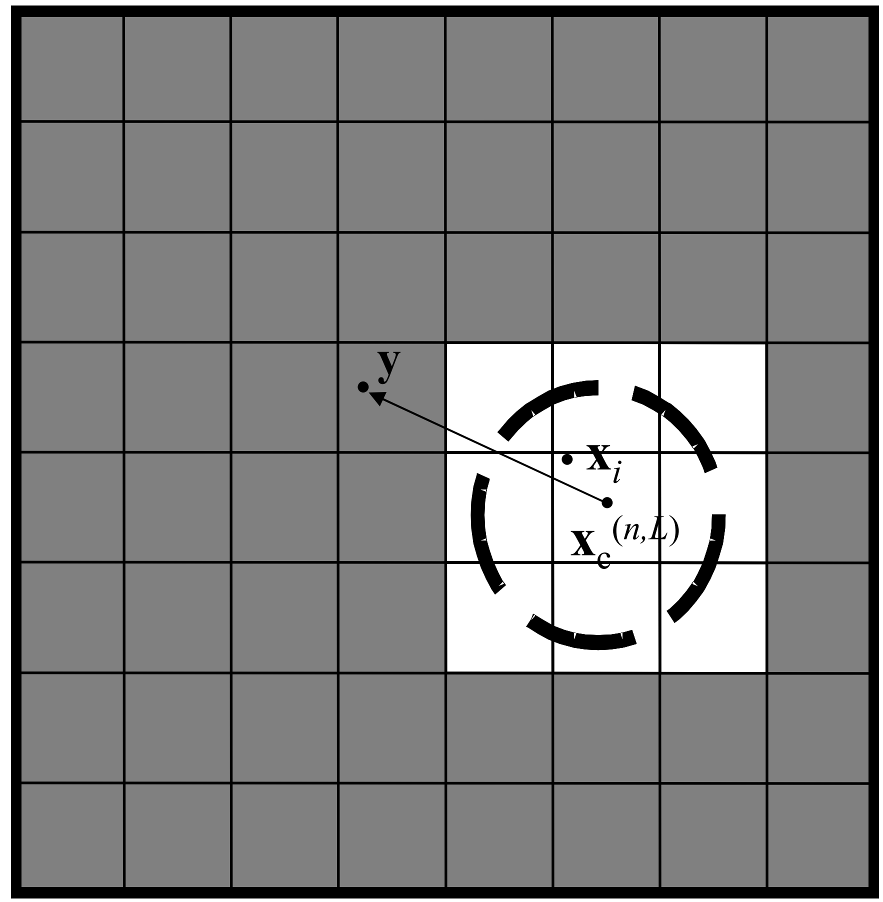
In the algorithm this means generation of the expansion coefficients $\mathbf{B}\left(\mathbf{x}_i, \mathbf{x}_c^{(n,L)}\right)$ and determination of $\mathbf{C}^{(n,L)}$ for each box. If at the finest level each non-empty box contains only one source $\mathbf{x}_i$, then for such box $\mathbf{C}^{(n,L)} = u_i \mathbf{B}\left(\mathbf{x}_i, \mathbf{x}_c^{(n,L)}\right)$. Note that this expansion for *n*th box is valid in domain $E_3(n,L)$. If the *n*th box is empty $\Phi_1^{(n,L)}(\mathbf{y}) = 0$ (or $\mathbf{C}^{(n,L)} = 0$) for such a box. There is no need to keep zero $\mathbf{C}^{(n,L)}$ in the memory, since the empty boxes can be skipped in the procedure.

Upward Pass. Step 1.

S-expansion valid in $\overline{\Omega}$



E_3



S-expansion valid in $E_3(n,L)$

Upward Pass. Step 2.

Step 2. For $l = L-1, \ldots, 2$ recursively form $\Phi_1^{(n,l)}(\mathbf{y})$ (in other words determine expansion coefficients of this function) by reexpansion of $\Phi_1^{(Children(n),l+1)}(\mathbf{y})$ near the center of the parent box and summing up of contribution of all children boxes:

$$\Phi_1^{(n,l)}(\mathbf{y}) = \mathbf{C}^{(n,l)} \circ \mathbf{S}(\mathbf{y} - \mathbf{x}_c^{(n,l)}),$$

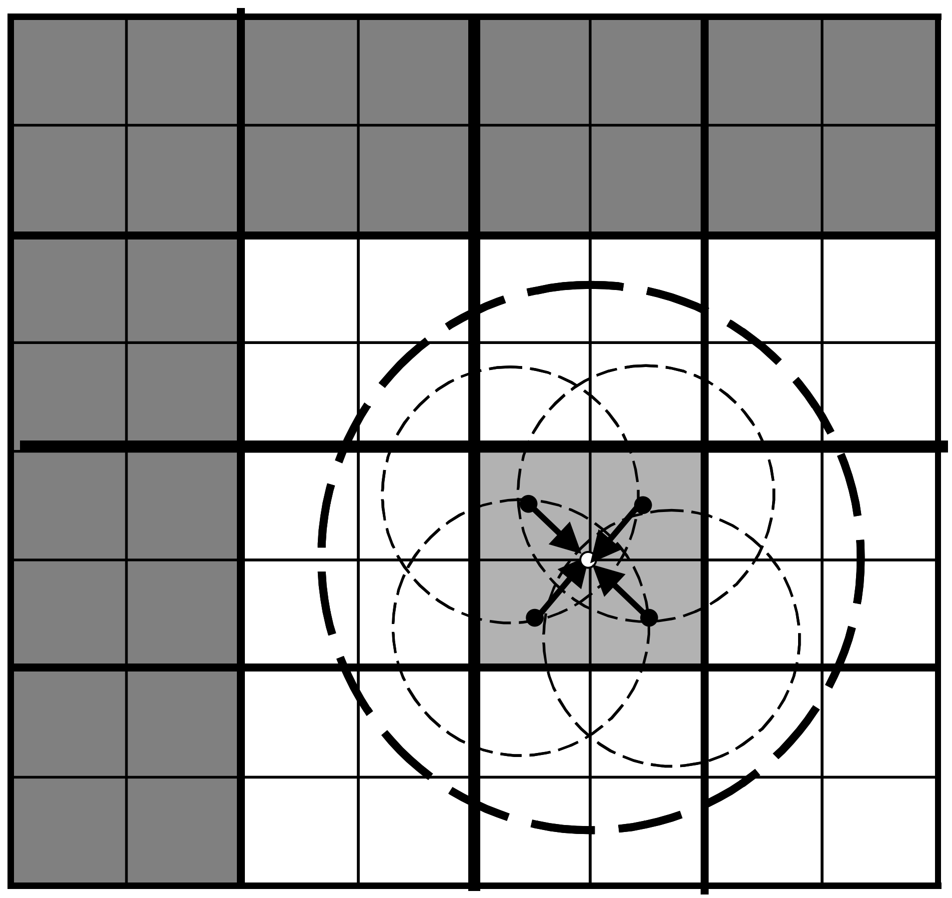
$$\mathbf{C}^{(n,l)} = \sum_{n' \in Children(n)} (\mathbf{S}|\mathbf{S})\left(\mathbf{x}_c^{(n',l+1)} - \mathbf{x}_c^{(n,l)}\right)\mathbf{C}^{(n',l+1)}.$$

For the nth box this expansion is valid in domain $E_3(n,l)$ which is a subdomain, where far-to-far translation is applicable. The set $Children(n)$ has 2^d entries, and summation over empty boxes of set $\mathbb{X}$ can be skipped (anyway for such boxes $\mathbf{C}^{(n',l+1)} = 0$).

Upward Pass. Step 2.

S|S-translation.

Build potential for the parent box (find its S-expansion).



Result of the Upward Pass

In the entire hierarchy of boxes containing *sources* S-expansion coefficients for potentials due to *sources* in each box (domains E_1) are found. Expansions are valid in E_3 domains.

Downward Pass. Step 1.

Step 1. Steps 1 and 2 should be performed recursively for levels $l = 2, \ldots L$ of space subdivision. At this step form coefficients of regular expansion for function $\Phi_4^{(n,l)}(\mathbf{y})$. To build local expansion near the center of each box at level l coefficients $\mathbf{C}^{(m,l)}, m \in I_4(n,l)$ should be $(\mathbf{S}|\mathbf{R})$- translated to the center of this box. So we have

$$\Phi_4^{(n,l)}(\mathbf{y}) = \widetilde{\mathbf{D}}^{(n,l)} \circ \mathbf{R}(\mathbf{y} - \mathbf{x}_c^{(n,l)}),$$

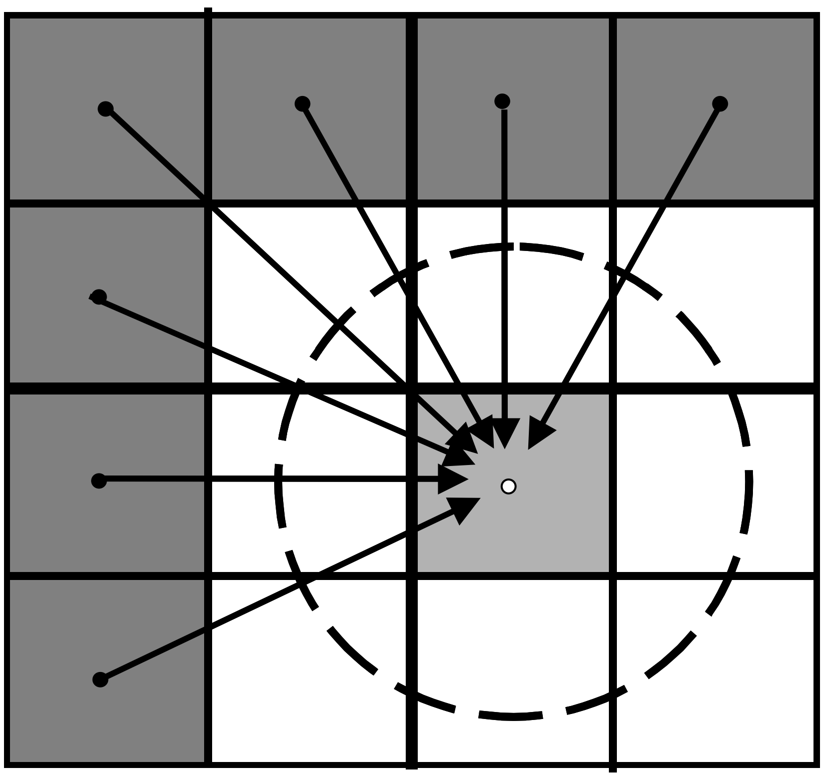
$$\widetilde{\mathbf{D}}^{(n,l)} = \sum_{m \in I_4(n,l)} (\mathbf{S}|\mathbf{R})\left(\mathbf{x}_c^{(n,l)} - \mathbf{x}_c^{(m,l)}\right)\mathbf{C}^{(m,l)}.$$

Since each box of level l is separated from boxes of $I_4(n,l)$ by a sphere drown near its center, then the far-to-local translation is applicable. Note that summation over empty boxes $m \in I_4(n,l)$ of set $\mathbf{X}$ can be skipped.

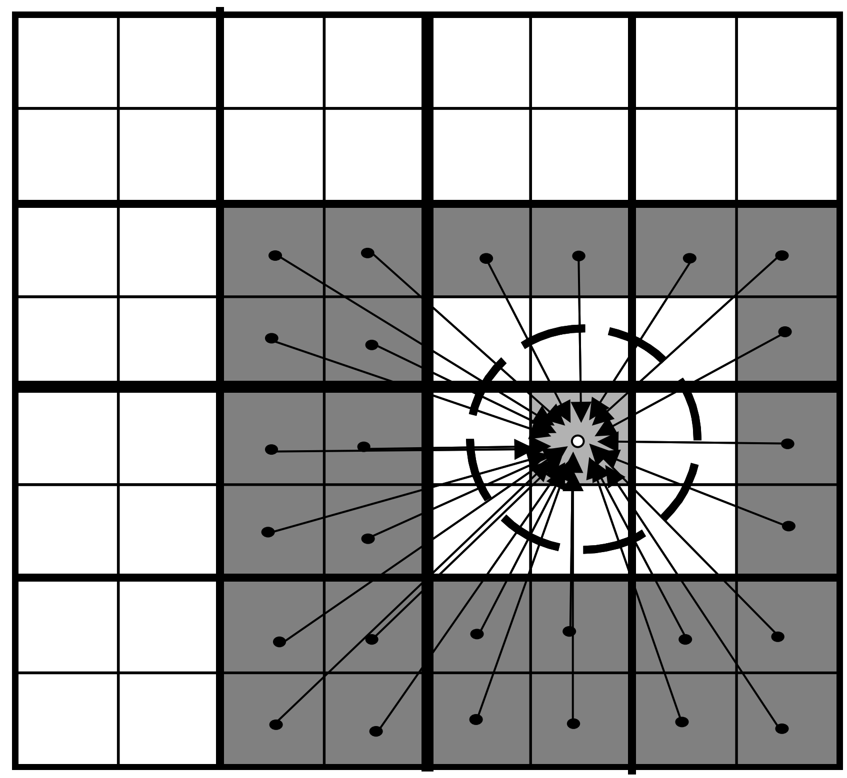
Note that this is conversion from the Source Hierarchy to Evaluation Hierarchy!

Downward Pass. Step 1.

Level 2:

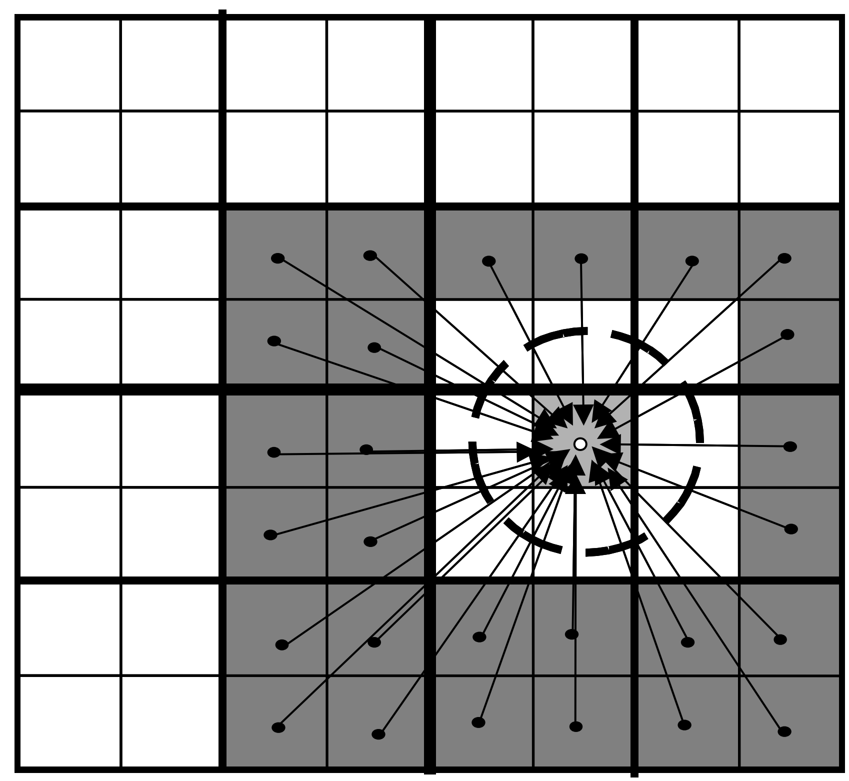


Level 3:



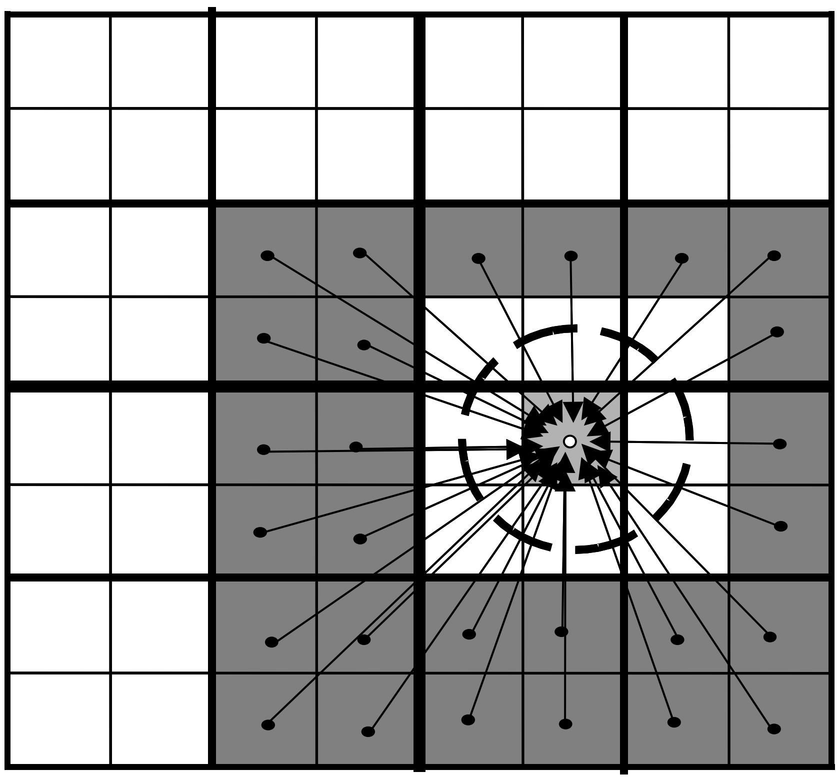
Downward Pass. Step 1.

THIS IS USUALLY THE
MOST EXPENSIVE STEP OF
THE ALGORITHM



Downward Pass. Step 1.

$$P_4 = PowerOfE_4Neighborhood = 3^d 2^d - 3^d = 3^d \left(2^d - 1\right)$$



$d = 1: \quad P_4 = 3,$

$d = 2: \quad P_4 = 27,$

$d = 3: \quad P_4 = 189$

Exponential Growth

Total number of S|R-translations per 1 box in *d*-dimensional space

(far from the domain boundaries)

It is worth to think about optimizations

Downward Pass. Step 2.

Step 2. At $l = 2$ we have

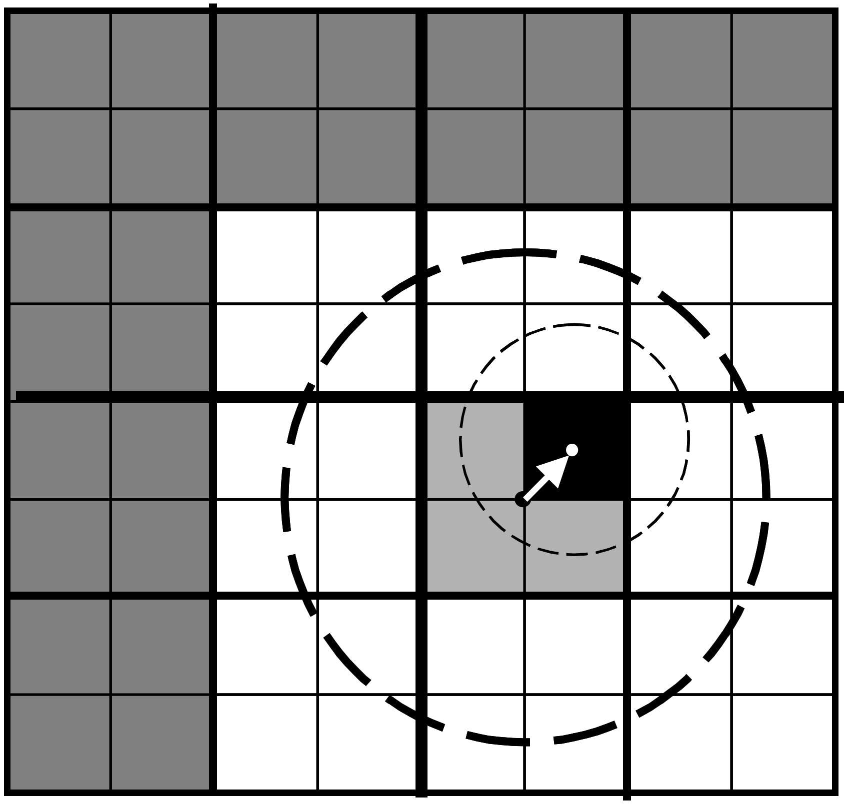
$$\Phi_3^{(n,2)}(\mathbf{y}) = \Phi_4^{(n,2)}(\mathbf{y}), \quad \mathbf{D}^{(n,2)} = \widetilde{\mathbf{D}}^{(n,2)},$$

Form $\Phi_3^{(n,l)}(\mathbf{y})$ (or expansion coefficients of this function) by adding $\Phi_4^{(Parent(n),l-1)}(\mathbf{y})$ to $(\mathbf{R}|\mathbf{R})$- translated coefficients of the parent box to the child center:

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

$$\mathbf{D}^{(n,l)} = \widetilde{\mathbf{D}}^{(n,l)} + (\mathbf{R}|\mathbf{R})\left(\mathbf{x}_c^{(n,l)} - \mathbf{x}_c^{(m,l-1)}\right)\mathbf{D}^{(m,l-1)}, \quad m = Parent(n).$$

Downward Pass. Step 2.



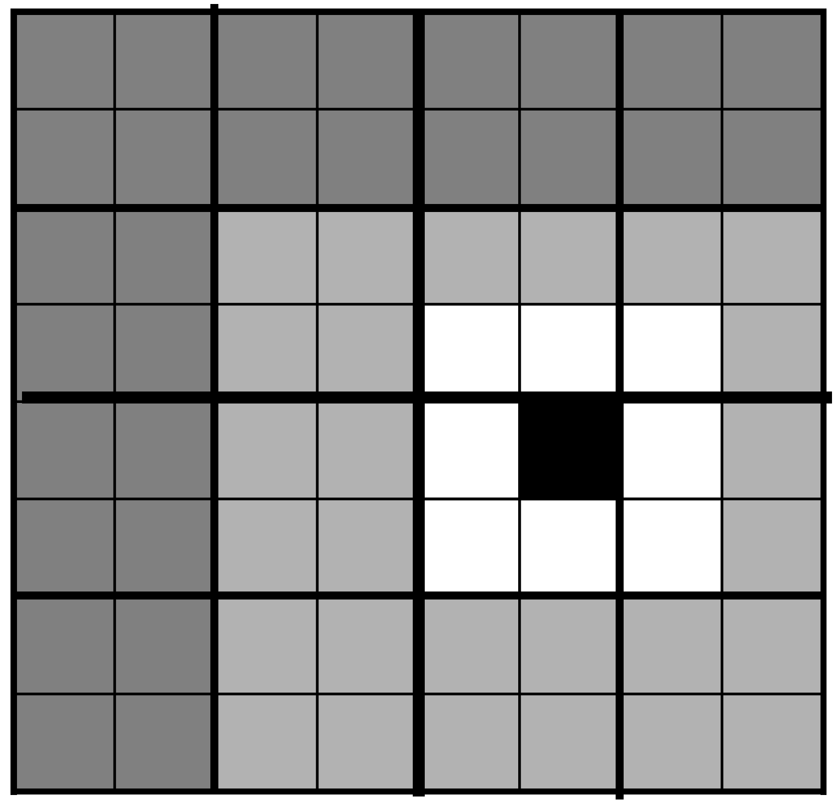


Figure shows that local-to-local translation is applicable in this case (smaller sphere is located completely inside the larger sphere), and junction of structures $E_3(n,l)$ and $E_4(n,l+1)$ produces $E_3(n,l+1)$:

$$E_3(n,l+1) = E_3(n,l) \cup E_4(n,l+1).$$

Result of the Downward Pass

In the entire hierarchy of boxes containing *evaluation points* R-expansion coefficients for potentials due to *sources* outside each *evaluation point* neighborhood (domains E_3) are found. Expansions are valid in E_1 domains.

Final Summation

As soon as coefficients $\mathbf{D}^{(n,L)}$ are determined total potential can be computed for any point $\mathbf{y}_j \in E_1(0,0)$, where $\Phi_2^{(n,l)}(\mathbf{y})$ can be computed straightforward. So:

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

Contribution of E_2

Contribution of E_3