CMSC 858M/AMSC 698R

Fast Multipole Methods

Nail A. Gumerov & Ramani Duraiswami

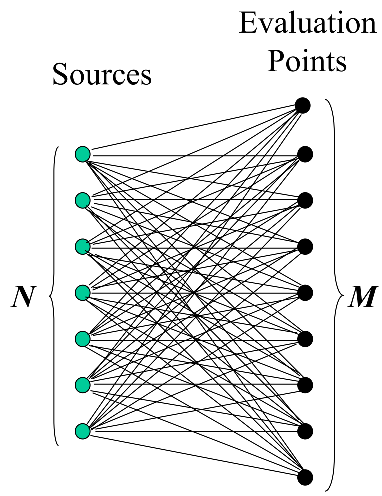
Lecture 9

Outline

- Basic Idea of Multilevel FMM (MLFMM);
- Formal Requirements for Functions (Potentials) in MLFMM;
- Setting Hierarchical Data Structure;
  - Hierarchical Domains and Associated Potentials (Functions);
  - Dimensionality Limits;
- MLFMM Algorithm;
  - Structure of the Algorithm;
  - Upward Pass;
  - Downward Pass;
  - Final Summation.

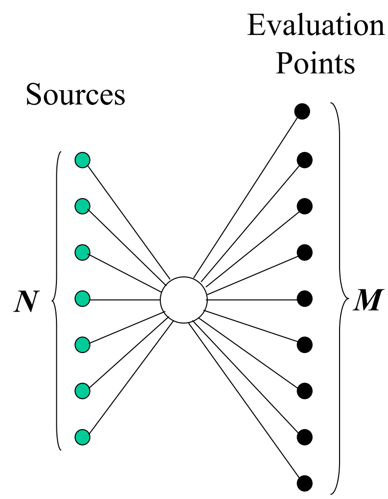
Middleman Algorithm

Standard algorithm



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

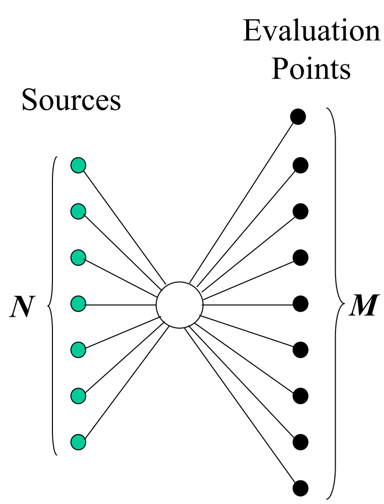
Middleman algorithm



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

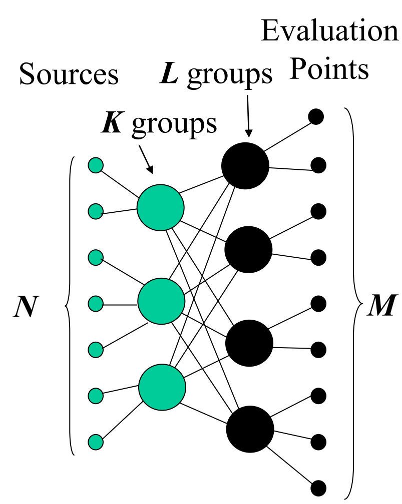
Single Level FMM

Middleman algorithm



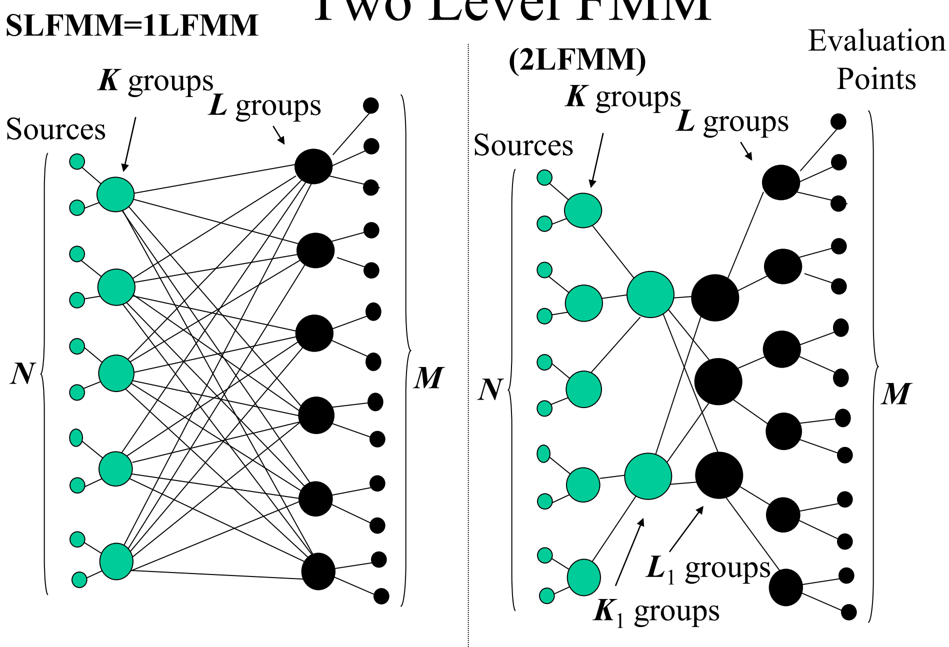
Total number of operations: ***O(N+M)***

SLFMM



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

Two Level FMM



Total number of operations: $O(N+M+KL)$ Total number of operations: $O(N+M+K+L+K_1L_1)$

KxL-interaction of two groups can be reduced by further grouping to $K+L+K_1L_1$

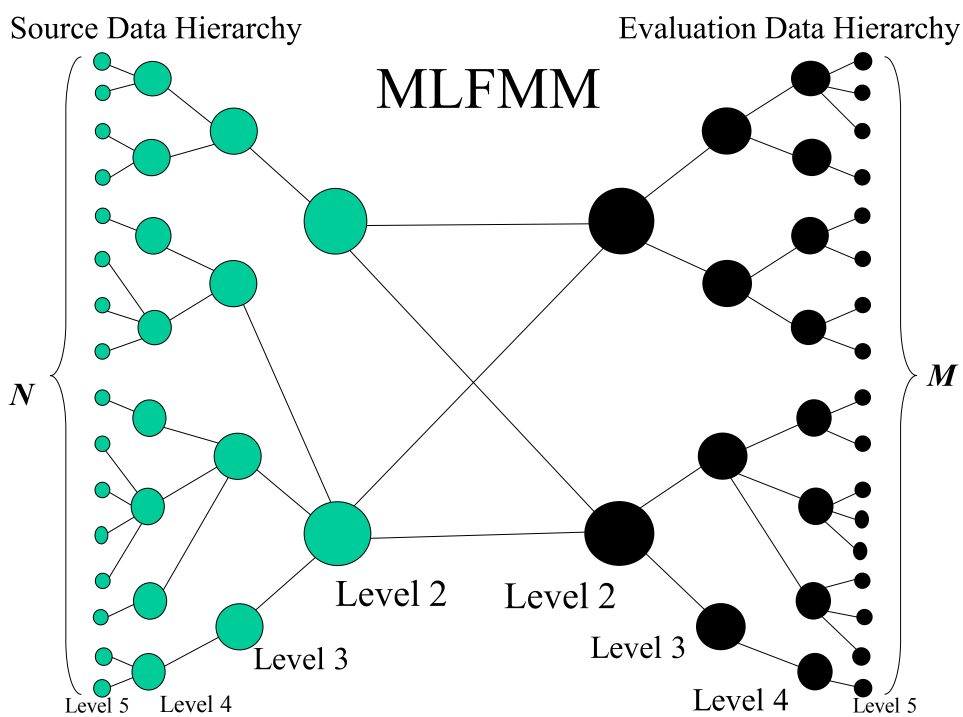
Indeed, if $K=nK_1$, $L=mL_1$, then

$K+L+K_1L_1= nK_1+ mL_1+K_1L_1$,

while $KL= nmK_1L_1$.

Problem for Thinking: What are conditions for n,m,K_1,L_1 to have

$nmK_1L_1 > nK_1+ mL_1+K_1L_1$?



Example of Multi Level Structure (Post Offices)

Source Hierarchy (Area) ↓

- People (sources, Level 5)
- Mail Box, Post Master (Level 4)
- Local Post Offices (Level 3)
- City Post Office (Level 2)

↓ Mail Transfer

AIRCRAFT

Receiver Hierarchy (Area) ↑

- City Post Office (Level 2)
- Local Post Offices (Level 3)
- Post Master (Level 4)
- People (receivers, Level 5)

↓ Mail Transfer

Example of MLFMM
(Computation of Gravity Field)

Source Hierarchy (Mass, Distance) ↓

- Planets (Level 5)
- Solar Systems (Level 4)
- Galaxies (Level 3)
- Metagalaxies (Level 2)

Influence ↓

Far Field to Local Field

Receiver Hierarchy (Mass, Distance) ↑

- Metagalaxies (Level 2)
- Galaxies (Level 3)
- Solar Systems (Level 4)
- Planets (Level 5)

Influence ↓

Exercise:

Create your own example!

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:

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

- 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):

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

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

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

- The product is distributive operation with respect to addition

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

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

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

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

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

- 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.

Two Parts of the MLFMM

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

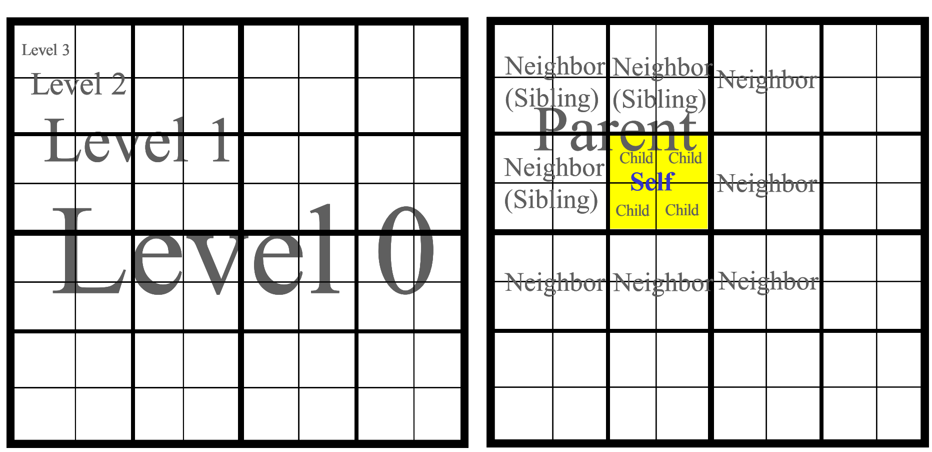
- MLFMM "Summator" $O(N+M)$ or $O(N\log^q N+M\log^q M)$,
Will evaluate the complexity in more details later.

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

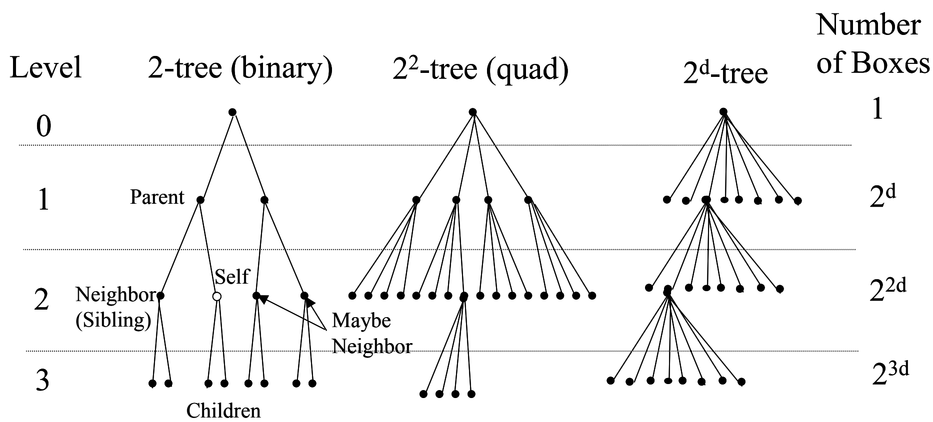
Setting Hierarchical Data Structure

- 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 *(In 2 weeks)*
- Determine the level of space subdivision with 2^d-tree to have *s* sources at the finest subdivision level, L_{max} *(In 2 weeks)*
- 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



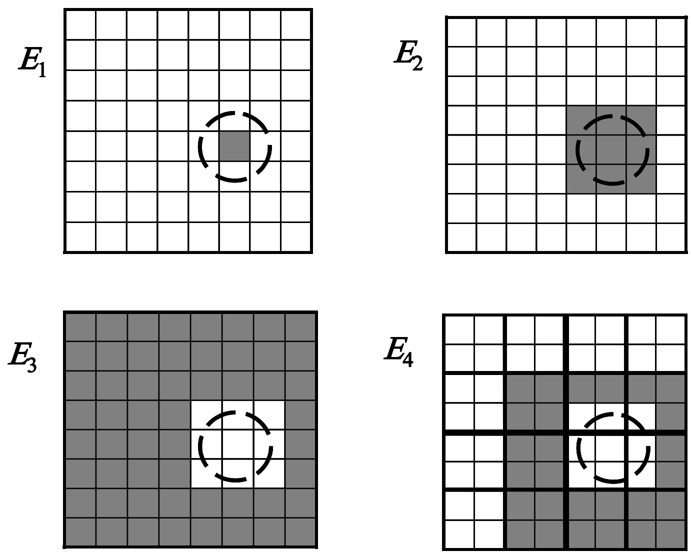
2^d-trees

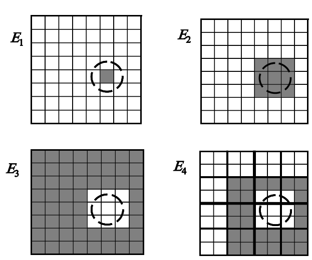


Hierarchical Indexing and Functions

- 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 in 2 weeks). 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 $n = Number = 0, \dots, 2^{dl} - 1$ at level $l = 0, \dots, L$.

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

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

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

$E_4(n,l) = E_2(Parent(n), l-1) \setminus E_2(n,l)$ 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:

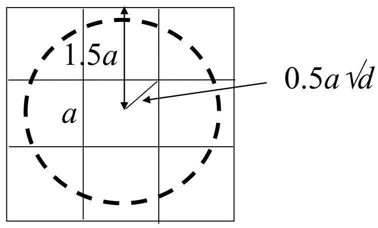
$I_1(n,l) = (n,l)$,

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

$I_3(n,l) = \{0, 1, \dots, 2^{dl} - 1\} \setminus I_2(n,l)$,

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

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



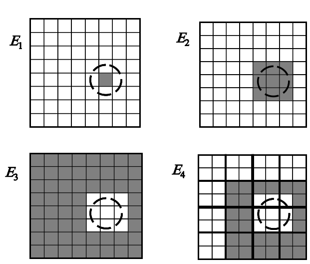
$$0.5a\sqrt{d} < 1.5a,$$
$$\sqrt{d} < 3,$$
$$d < 9.$$

For larger dimensions larger neighborhoods should be considered (but seems it is not practical to use 2^d-trees in this case and something better should be invented).

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 (we also call it sometimes ``Regular FMM") 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 $\mathbf{x}_c^{(n,L)}$:

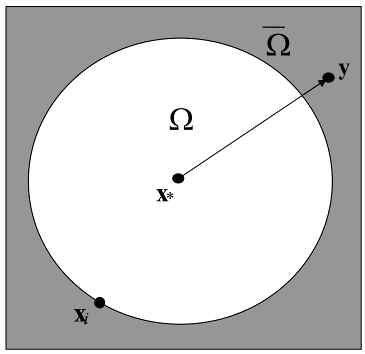
$$\Phi_1^{(n,L)}(\mathbf{y}) = \mathbf{C}^{(n,L)} \circ \mathbf{S}(\mathbf{y} - \mathbf{x}_c^{(n,L)}),$$

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

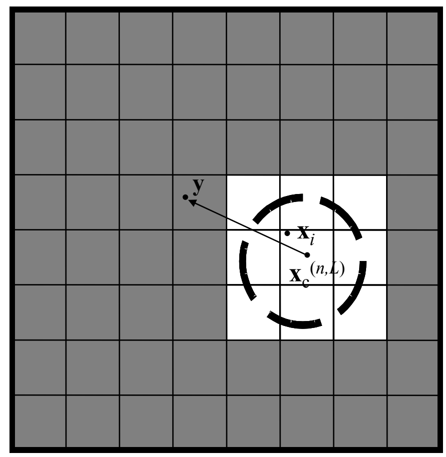
In the algorithm this means generation of the expansion coefficients $\mathbf{B}\left(\mathbf{x}_i, \mathbf{x}_c^{(n,L)}\right)$ and determination of $\mathbf{C}^{(n,L)}$ for each box. If at the finest level each non-empty box contains only one source $\mathbf{x}_i$, then for such box $\mathbf{C}^{(n,L)} = u_i \mathbf{B}\left(\mathbf{x}_i, \mathbf{x}_c^{(n,L)}\right)$. Note that this expansion for nth box is valid in domain $E_3(n,L)$. If the nth 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:

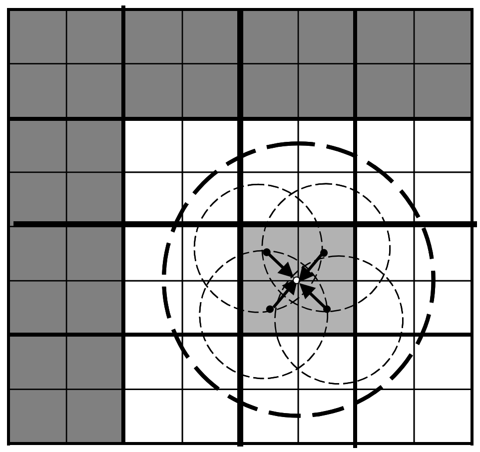
$$\Phi_1^{(n,l)}(\mathbf{y}) = \mathbf{C}^{(n,l)} \circ \mathbf{S}(\mathbf{y} - \mathbf{x}_c^{(n,l)}),$$

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

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

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

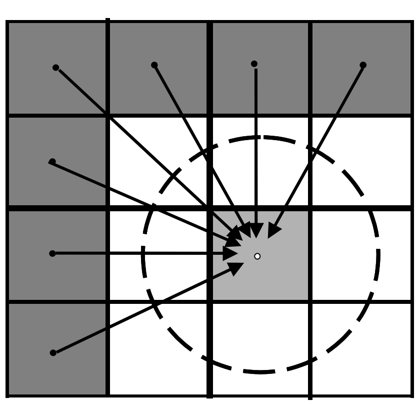
$$\widetilde{\mathbf{D}}^{(n,l)} = \sum_{m \in I_4(n,l)} (\mathbf{S}|\mathbf{R})\left(\mathbf{x}_c^{(n,l)} - \mathbf{x}_c^{(m,l)}\right)\mathbf{C}^{(m,l)}.$$

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 X can be skipped.

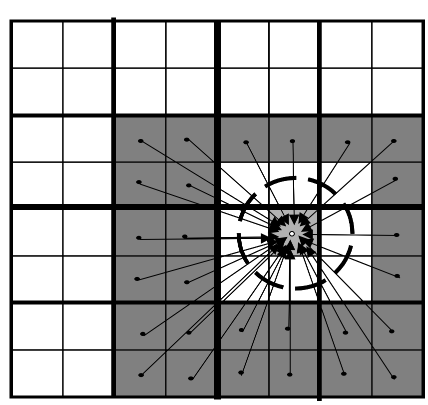
Note that this is conversion from the Source Hierarchy to Evaluation Hierarchy!

Downward Pass. Step 1.

Level 2:

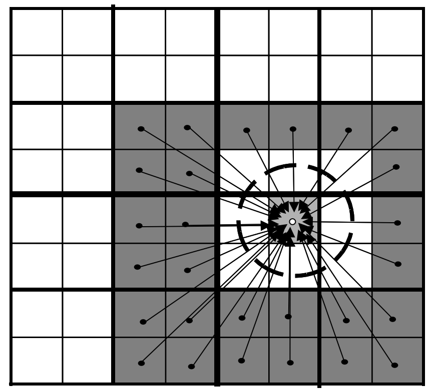


Level 3:



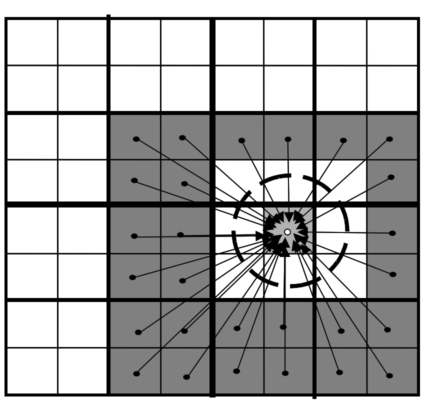
Downward Pass. Step 1.

THIS MIGHT BE
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.

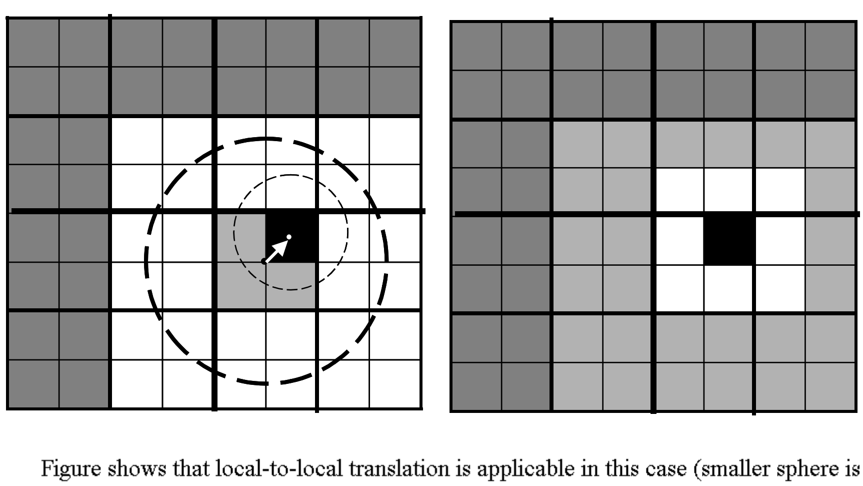


Figure shows that local-to-local translation is applicable in this case (smaller sphere is located completely inside the larger sphere), and junction of structures $E_3(n,l)$ and $E_4(n,l+1)$ produces $E_3(n,l+1)$:

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

Result of the Downward Pass

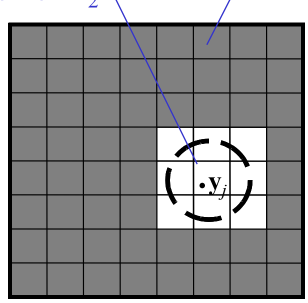
In the entire hierarchy of boxes containing *evaluation points* R-expansion coefficients for potentials due to *sources* outside each *evaluation point* neighborhood (domains E_3) are found. Expansions are valid in E_1 domains.

Final Summation

As soon as coefficients $\mathrm{D}^{(n,L)}$ 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:

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

Contribution of E_2 — Contribution of E_3



Itemized Cost of MLFMM

Regular mesh:

$$N = 2^{L_* d}, \quad s = 2^{L_s d}, \quad L = L_{\max} = L_* - L_s.$$

Assume that all translation costs are the same, *CostTranslation*(*P*)

$$CostUpward_1 = NCostExpansion(P) = O(NP).$$

$$CostUpward_2 = 2^d\left(2^{(L-1)d} + 2^{(L-2)d} + \ldots + 2^{2d}\right)CostSS(P)$$
$$< \frac{2^d}{2^d - 1}\left(2^{Ld} - 1\right)CostSS(P) \sim \frac{N}{s}CostSS(P)$$

$$CostDownward_1 \lesssim P_4(d)\left(2^{2d} + \ldots + 2^{Ld}\right)CostSR(P) \sim P_4(d)\frac{N}{s}CostSR(P),$$

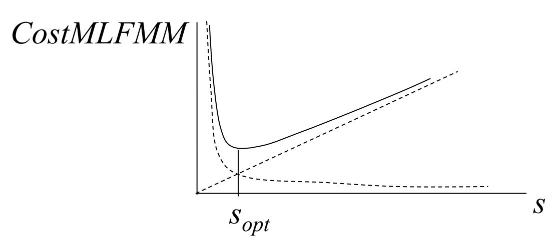
$$CostDownward_2 = 2^d\left(2^{2d} + \ldots + 2^{(L-1)d}\right)CostRR(P) \sim \frac{N}{s}CostRR(P),$$

$$CostEvaluation = M(P_2(d)sCostFunc + P).$$

Powers of E_4 and E_2 neighborhoods

$$CostMLFMM = (M+N)P + (P_4(d)+2)\frac{N}{s}CostTranslation(P) + P_2(d)sMCostFunc$$

Optimization of the Grouping Parameter



$$CostMLFMM = (M+N)P + (P_4(d)+2)\frac{N}{s}CostTranslation(P) + P_2(d)sMCostFunc$$

$$\frac{\partial CostMLFMM}{\partial s} = -(P_4(d)+2)\frac{N}{s^2}CostTranslation(P) + P_2(d)MCostFunc = 0$$

$$s_{opt} = \left[\frac{N(P_4(d)+2)CostTranslation(P)}{MP_2(d)CostFunc}\right]^{1/2}.$$

$$CostMLFMM_{opt} = (M+N)P + 2[MN(P_4(d)+2)P_2(d)CostTranslation(P)CostFunc]^{1/2}.$$

Optimization of the Grouping Parameter (Example)

$$s_{opt} = \left[\frac{N(P_4(d)+2)CostTranslation(P)}{MP_2(d)CostFunc} \right]^{1/2}.$$

$$CostMLFMM_{opt} = (M+N)P + 2[MN(P_4(d)+2)P_2(d)CostTranslation(P)CostFunc]^{1/2}.$$

Example:

$$N = M, \quad P_4(d) = 3^d\left(2^d - 1\right), \quad P_2(d) = 3^d,$$

$$CostTranslation(P) = P^2, \quad CostFunc = 1$$

$$s_{opt} \sim 2^{d/2}P; \quad CostMLFMM_{opt} \sim 2NP\left(1 + 3^d 2^{d/2}\right)$$

$$For\ d = 2, \quad P = 10, \quad s_{opt} \sim 38, \quad CostMLFMM_{opt} \sim 38NP = 380N.$$

If non-optimized,

$$s = 1; \quad CostMLFMM \quad \sim NP\left(2 + 3^d 2^d P\right)$$

$$For\ d = 2, \quad P = 10, \quad s = 1, \quad CostMLFMM \quad \sim 360NP = 3600N.$$

In this example optimization results in about 10 times savings!