Fast Multipole Methods

Lecture 8

Nail Gumerov & Ramani Duraiswami

Outline

- Background
- Translations
- Example of S|R-translation
- Requirements for functions 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

Matrix vector product

$$\mathbf{v} = \Phi\mathbf{u},$$

$$\Phi_{ji} = \Phi(\mathbf{y}_j, \mathbf{x}_i), \quad j = 1,\ldots,M, \quad i = 1,\ldots,N,$$

$$\Phi = \begin{pmatrix} \Phi_{11} & \Phi_{12} & \ldots & \Phi_{1N} \\ \Phi_{21} & \Phi_{22} & \ldots & \Phi_{2N} \\ \ldots & \ldots & \ldots & \ldots \\ \Phi_{M1} & \Phi_{M2} & \ldots & \Phi_{MN} \end{pmatrix} = \begin{pmatrix} \Phi(\mathbf{y}_1,\mathbf{x}_1) & \Phi(\mathbf{y}_1,\mathbf{x}_2) & \ldots & \Phi(\mathbf{y}_1,\mathbf{x}_N) \\ \Phi(\mathbf{y}_2,\mathbf{x}_1) & \Phi(\mathbf{y}_2,\mathbf{x}_2) & \ldots & \Phi(\mathbf{y}_2,\mathbf{x}_N) \\ \ldots & \ldots & \ldots & \ldots \\ \Phi(\mathbf{y}_M,\mathbf{x}_1) & \Phi(\mathbf{y}_M,\mathbf{x}_2) & \ldots & \Phi(\mathbf{y}_M,\mathbf{x}_N) \end{pmatrix}.$$

Entries of matrix are known as a function of two point sets in R^d

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

Matrix completely characterized by these points and the function Φ

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

Jargon

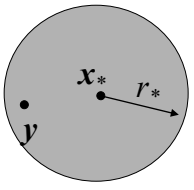
- x_i is the center of the source or "mother function"
- Φ is the mother function
  - It is often singular
- y is the evaluation point
- x_* is the expansion center

Factorization and power series

- Wish to factorize the entries of this function in a way that the "entanglement" of the point sets is broken
- Write it as a p term sum in which each term involves a product of a function of x alone and y alone
- Most functions do not allow a finite factorization
- Furthermore, functions of interest may be singular
  - Factorizations may only be valid inside or outside a neighborhood

Regular Function Expansion

Given a function known to be regular in a given domain centered around the point $\mathbf{x}_*$
Expand the function in a series



Basis Functions

$$\Phi(\mathbf{y},\mathbf{x}_i) = \sum_{m=0}^{\infty} a_m(\mathbf{x}_i,\mathbf{x}_*) R_m(\mathbf{y}-\mathbf{x}_*)$$

Expansion Coefficients

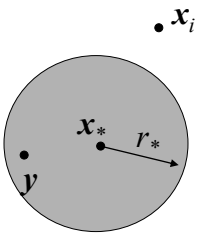
Location of center $\mathbf{x}_i$ *does not matter*

We may want to choose $\mathbf{x}_*$ *to control the number of terms*

We also call this the local or R-expansion, since basis functions R_m *should be regular*

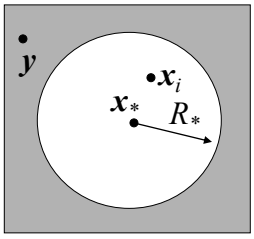
Expansions of a Singular Potential

Local



$|y - x_*| < r_* \leq |x_i - x_*|$

Singular or Multipole

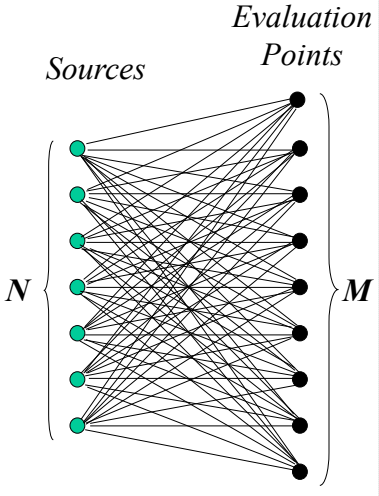


$|y - x_*| > R_* \geq |x_i - x_*|$

Because x_i is a singular point, there are two types of expansions:
Local in a finite neighborhood that excludes x_i
S

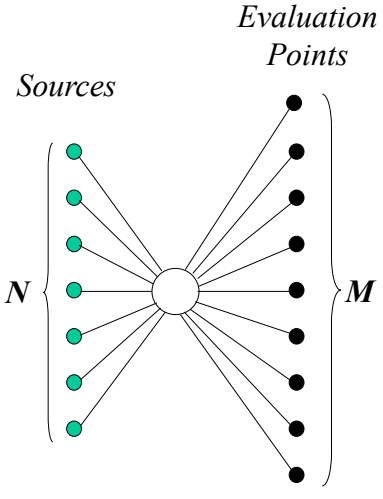
Middleman Algorithm

Standard algorithm



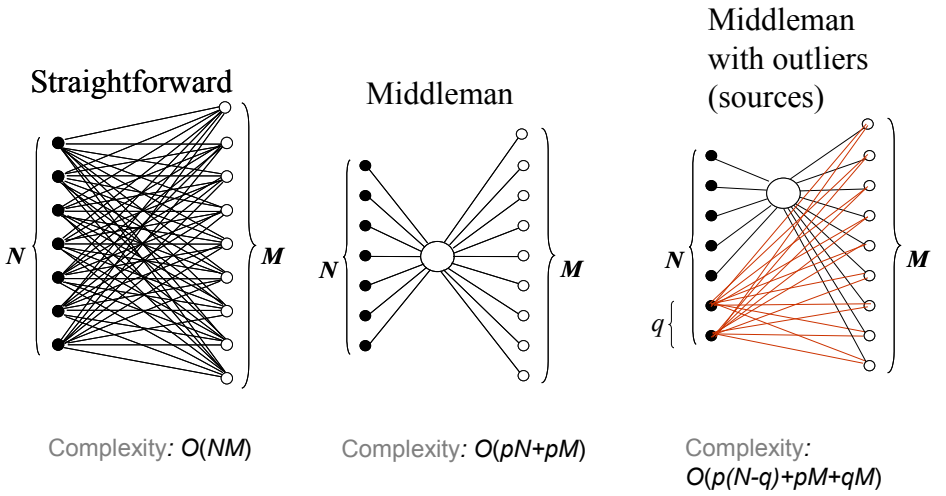
Total number of operations: ***O(NM)***

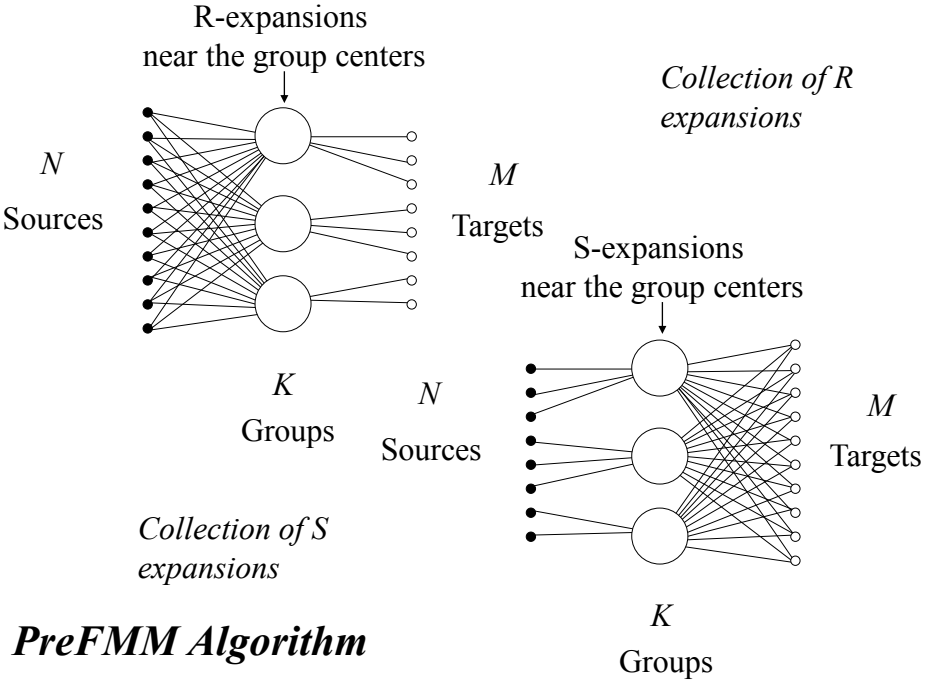
Middleman algorithm



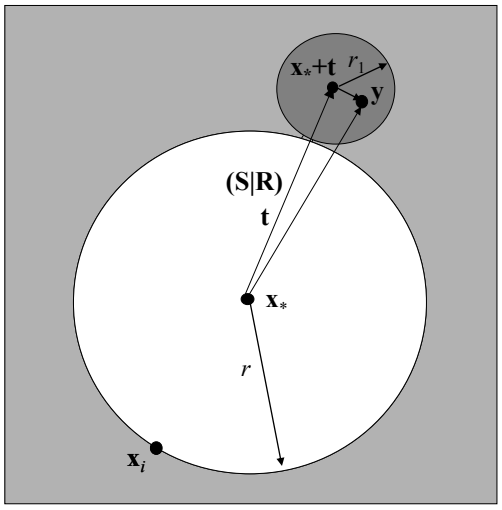
Total number of operations: ***O(Np+Mp)***

Modification of the "Middleman" for outliers





Multipole-to-Local S|R Translation

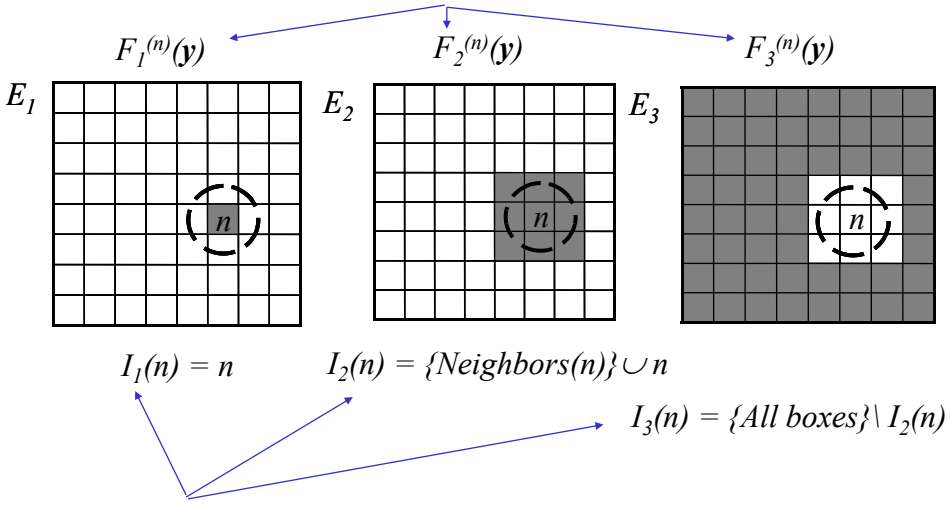


- Given an S-expansion centered around $\boldsymbol{x}_*$ and valid outside the white circle (in the shaded region)
- Create an expansion of the same function, valid in the dark shaded region
- Since the function is regular in this region we can build a regular expansion here
- Both the S-expansion and the R-expansion are valid in the dark region

Both represent the same function object in functional space
Use different basis functions

Using boxes to define regions

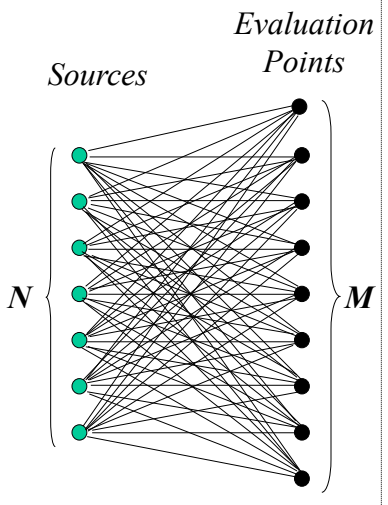
Potentials due to sources in these spatial domains



Boxes with these numbers belong to these spatial domains

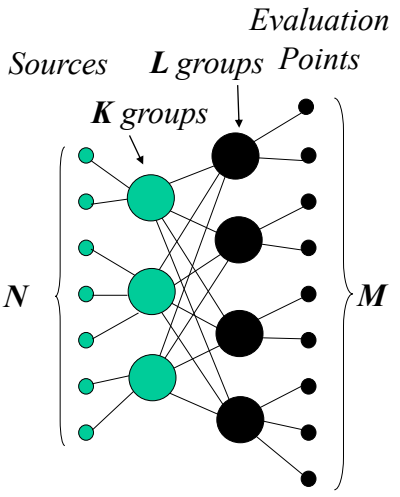
Idea of a Single Level FMM

Standard algorithm



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

SLFMM



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

Four Keys in the FMM

- Factorization
- Error
- Translation
- Grouping

Summary of requirements to use FMM

- Two sets of points: source and target

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

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

- A function defined on the X point set

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

- Sum of these to be evaluated at the Y points

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

- $\Phi(\mathbf{x}_i, \mathbf{y}\}$ has local and far field expansions

$$\Phi(\mathbf{x}_i, \mathbf{y})=A(\mathbf{x}_i, \mathbf{x}_*)\cdot R(\mathbf{y}, \mathbf{x}_*) \qquad |\mathbf{y}-\mathbf{x}_*| < r < |\mathbf{x}_i - \mathbf{x}_*|$$

$$\Phi(\mathbf{x}_i, \mathbf{y})=B(\mathbf{x}_i, \mathbf{x}_*)\cdot S(\mathbf{y}, \mathbf{x}_*) \qquad |\mathbf{y}-\mathbf{x}_*| > R > |\mathbf{x}_i - \mathbf{x}_*|$$

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

- The product is distributive operation with respect to addition

$$(u_1 A_1 + u_2 A_2) \circ \mathbf{F} = u_1 A_1 \circ \mathbf{F} + u_2 A_2 \circ \mathbf{F}, \quad \mathbf{F} = \mathbf{S}, \mathbf{R}$$

- *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)

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

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

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.

SLFMM Algorithm

Step 1. Generate S-expansion coefficients for each box

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

loop over all non-empty source boxes

For $n \in NonEmptySource$

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

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

For $\mathbf{x}_i \in E_1(n)$ ← loop over all sources in the box

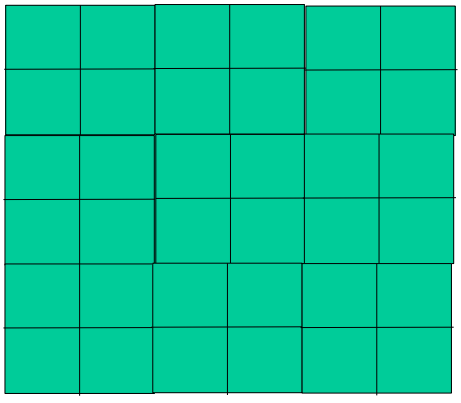
Get $\mathbf{B}(\mathbf{x}_i, \mathbf{x}_c^{(n)})$, the S-expansion coefficients near the center of the box;

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

End;

End;

S-expansion



SLFMM Algorithm

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

$$\Phi_3^{(n)}(\mathbf{y}) = \mathrm{D}^{(n)} \circ \mathrm{R}(\mathbf{y} - \mathbf{x}_c^{(n)}),$$
$$\mathrm{D}^{(n)} = \sum_{m \in I_3(n)} (\mathbf{S}|\mathrm{R})\left(\mathbf{x}_c^{(n)} - \mathbf{x}_c^{(m)}\right)\mathrm{C}^{(m)}.$$

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;

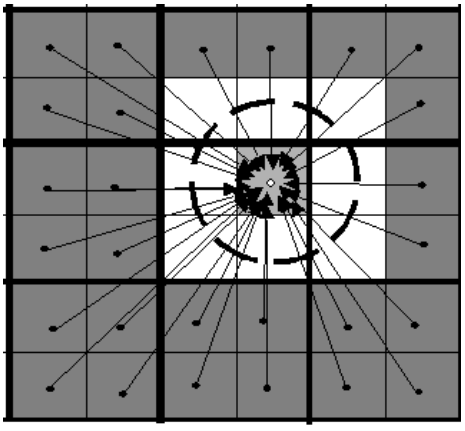
$\mathbf{D}^{(n)} = \mathbf{D}^{(n)} + (\mathbf{S}|\mathbf{R})(\mathbf{x}_c^{(n)} - \mathbf{x}_c^{(m)})\ \mathbf{C}^{(m)}$;

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

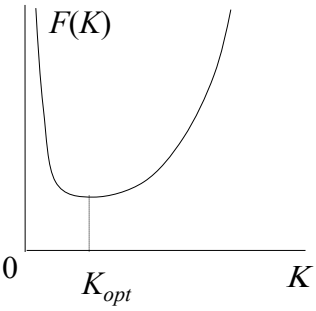
Selection of Optimal K (or s)

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

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

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

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



Complexity of Optimized SLFMM

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

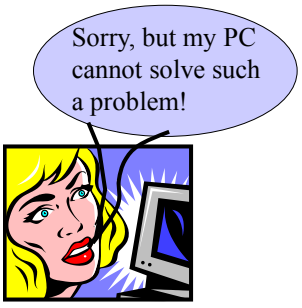
At $K = K_{opt}$, and $M = O(N)$, the complexity of SLFMM is:

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

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 !