Advanced Topics in Numerical Analysis: High Performance Computing

N-body problems and FMM

Georg Stadler, Dhairya Malhotra
Courant Institute, NYU

Spring 2019, Monday, 5:10–7:00PM, WWH #1302

May 13, 2019

Outline

Organization

Scheduling:

- Homework assignment #6 due Tonight.
- Let us know if you want to meet to discuss projects

Topics today:

- Summary of last class
- Fast Multipole Method
- Class evaluation

Organization

Final projects:

- Project presentation May 20/21. Sign up for your time slot (see google docs link on Slack)
- Each group is expected to present 10 minutes (8 slides), and summarize their work in a final report.
- Final report must acknowledge sources of code and outside help. It also must include a short statement of who did what in the team. Don't plagarize!

Organization

Final projects:

- Please focus presentation/project description on HPC aspects:
  - What algorithm, what parallelization and why?
  - Flop rate, main computational kernel, memory access,...
  - Scalability (weak/strong). Please run your code not only on your Laptop but on a "real" computing device (Prince, crunchy).
  - What's limits solving your problem faster, solving larger problems? Communication? Memory access? Amdahl's law?
  - Your personal experience/What have you learned?
  - Please put your code in a repository; We would like to post a link to your code and your final reports.

Outline

Sample Sort

Pick (p-1) local samples:

$local\_sample[i] = A[(i+1)N/p^2] \qquad i = 0, \cdots, p-2$

Gather samples on root and sort:

$sample \longleftarrow sort(\text{MPI\_Gather}(local\_sample))$

Pick (p-1) splitters:

$s_i = sample[(i+1)(p-1)] \qquad i = 0, \cdots, p-2$

# of elements $< s_i$

$\Longrightarrow [(p-1)(i+1)+1]\,\frac{N}{p^2}$ (at least)

# of elements $> s_{i+1}$

$\Longrightarrow [p(p-1)-(p-1)(i+2)]\,\frac{N}{p^2}$ (at least)

# of elements between s_i **and** s_{i+1}

$\Longrightarrow N - [(p-1)(i+1)+1+p(p-1)-(p-1)(i+2)]\,\frac{N}{p^2}$

$< 2N/p$

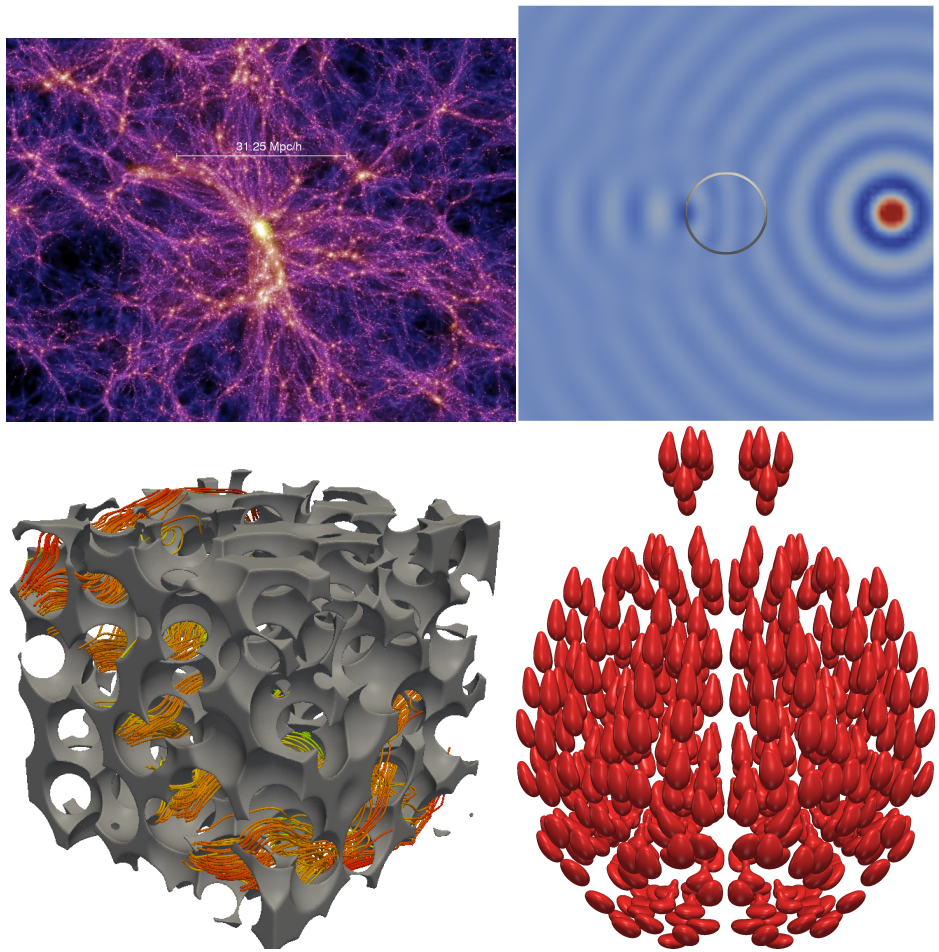
Outline

N-Body Problems

$$u_i = \sum_{j=1}^{N} G(x_i - y_j) f_j \qquad \text{for all i}$$

Applications:

- Cosmological simulations (Millennium simulation, Bolshoi simulation)
- Electromagnetic, acoustic scattering (radars, ultrasound)
- Fluid dynamics (Stokes flow, voretex methods)
- Machine learning (Kernel density estimation)

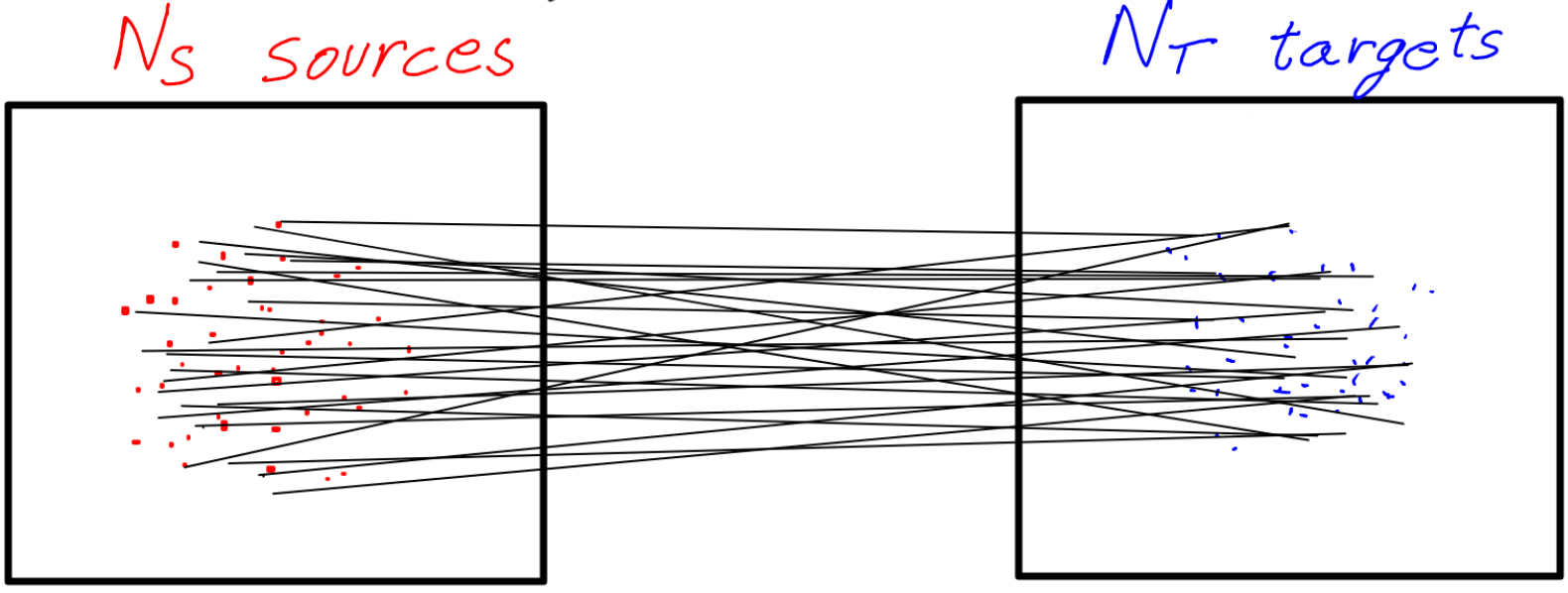


N-Body Problem

Compute Gravitational potential $\Rightarrow$ $u_i = \sum_{j=1}^{N_S} G(y_j - x_i) f_j$ for all $i = 1$ to N_T

$= \sum_j G_{ij} f_j$

where $G(r) = \frac{1}{4\pi |r|}$



Cost $= O(N_S N_T)$

Idea1: center of mass approximation

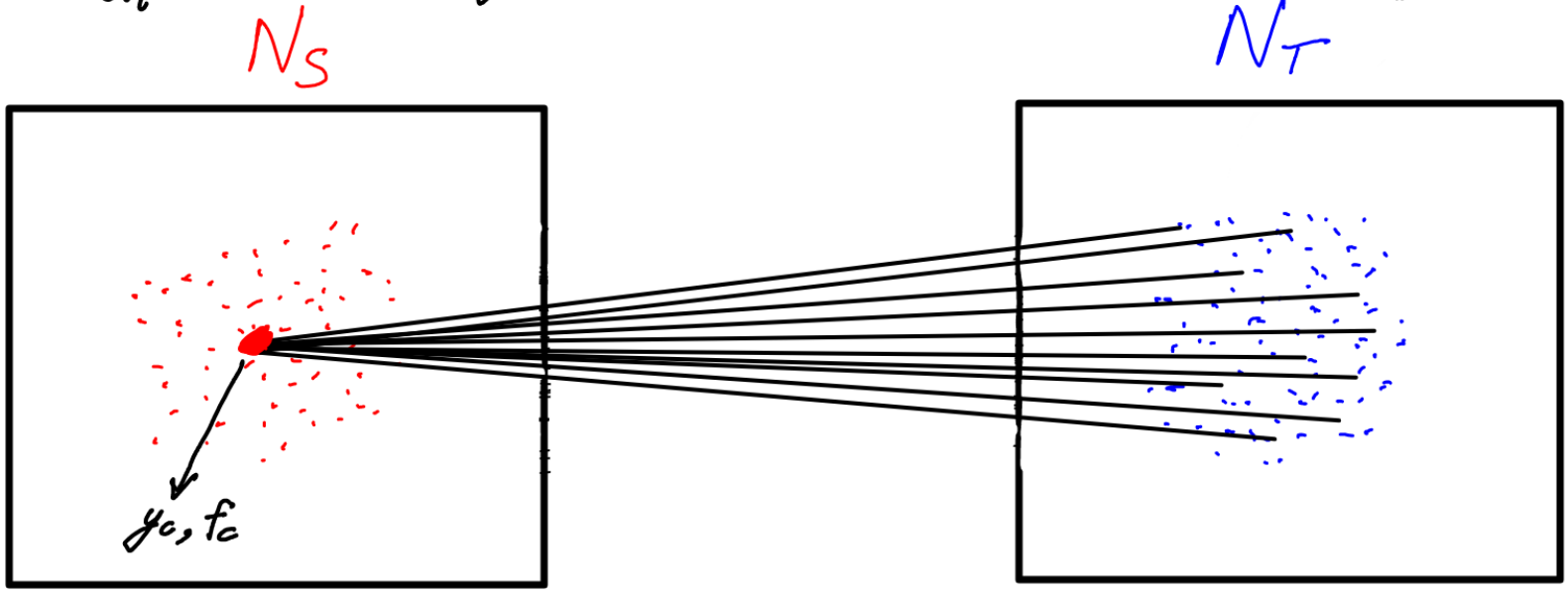
Center of mass approximation $\Rightarrow$ monopole or first order
→ higher order : dipole , quadrupole, ..., multipole.

$$f_c = \sum_j f_j$$

$$y_c = \frac{\sum_j f_j y_j}{f_c}$$

$\Big\}\, O(N_S)$

$$u_i = G_T(x_i - y_c)\, m_c \quad \text{for all } i = 1 \dots N_T \;\Big\}\, O(N_T)$$



Cost $= O(N_S + N_T)$

Works only when soures and targets are well-separated

separation > source box edge-length

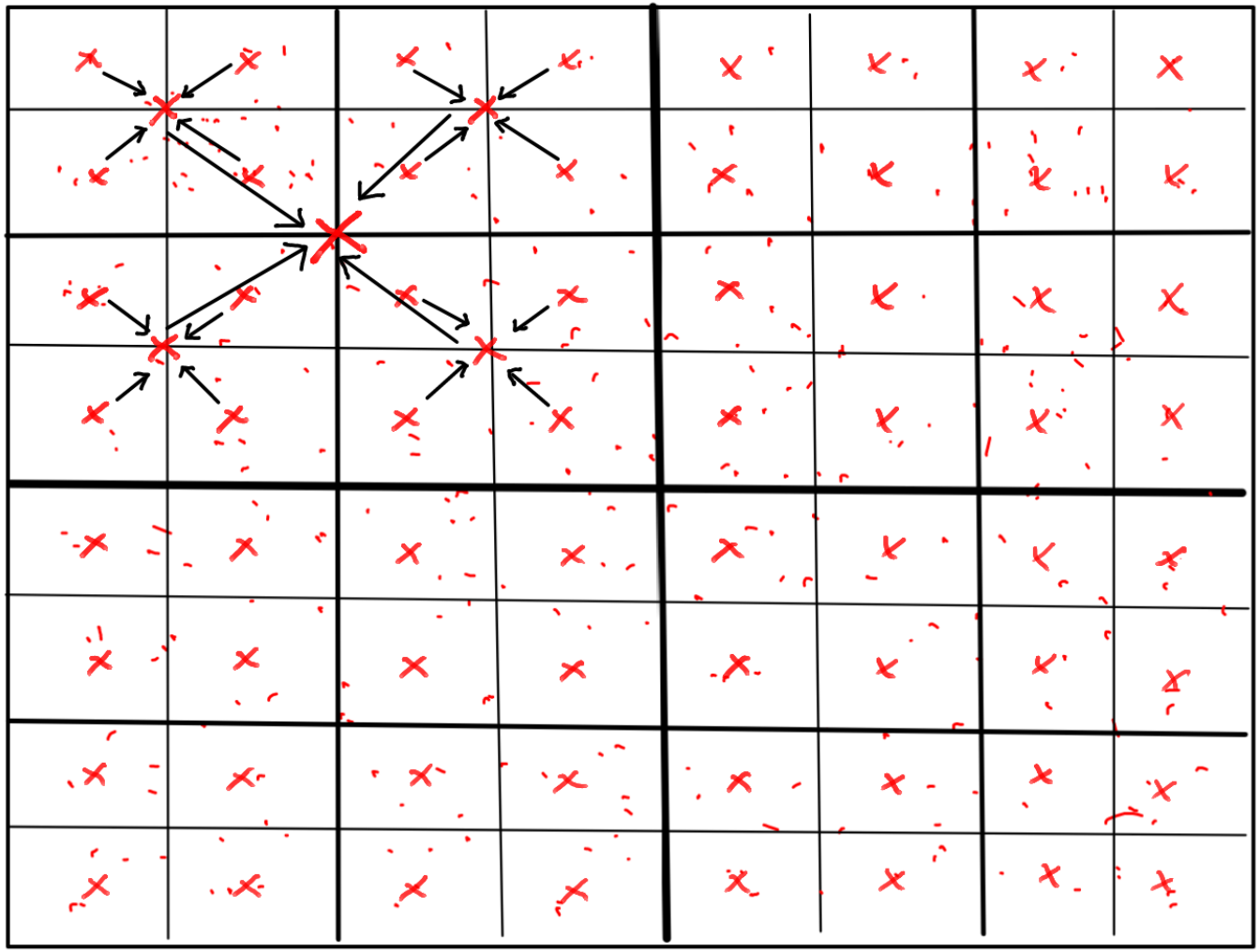
Idea2: apply center of mass approximation hierarchically

Step 0 : Build a tree such that the # of points per leaf is bounded.

Step1:
compute center of mass for each leaf box.

Step2:
compute center of mass for non-leaf boxes from the center of mass of its children.

Cost = $O(N_s)$

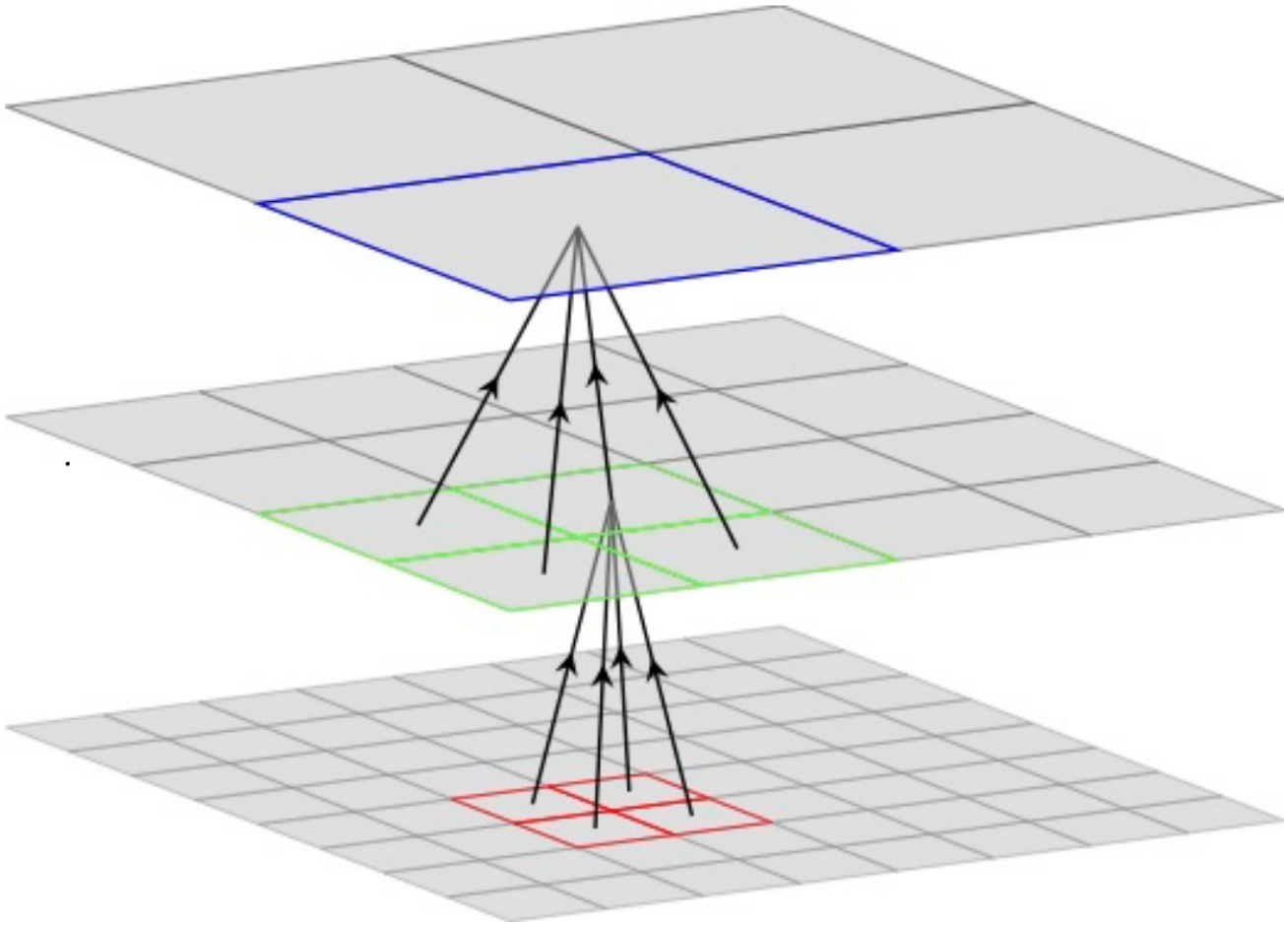


Idea2: apply center of mass approximation hierarchically

Step 0 : Build a tree such that the # of points per leaf is bounded.

Step1:
compute center of mass for each leaf box.

Step 2:
compute center of mass for non-leaf boxes from the center of mass of its children.



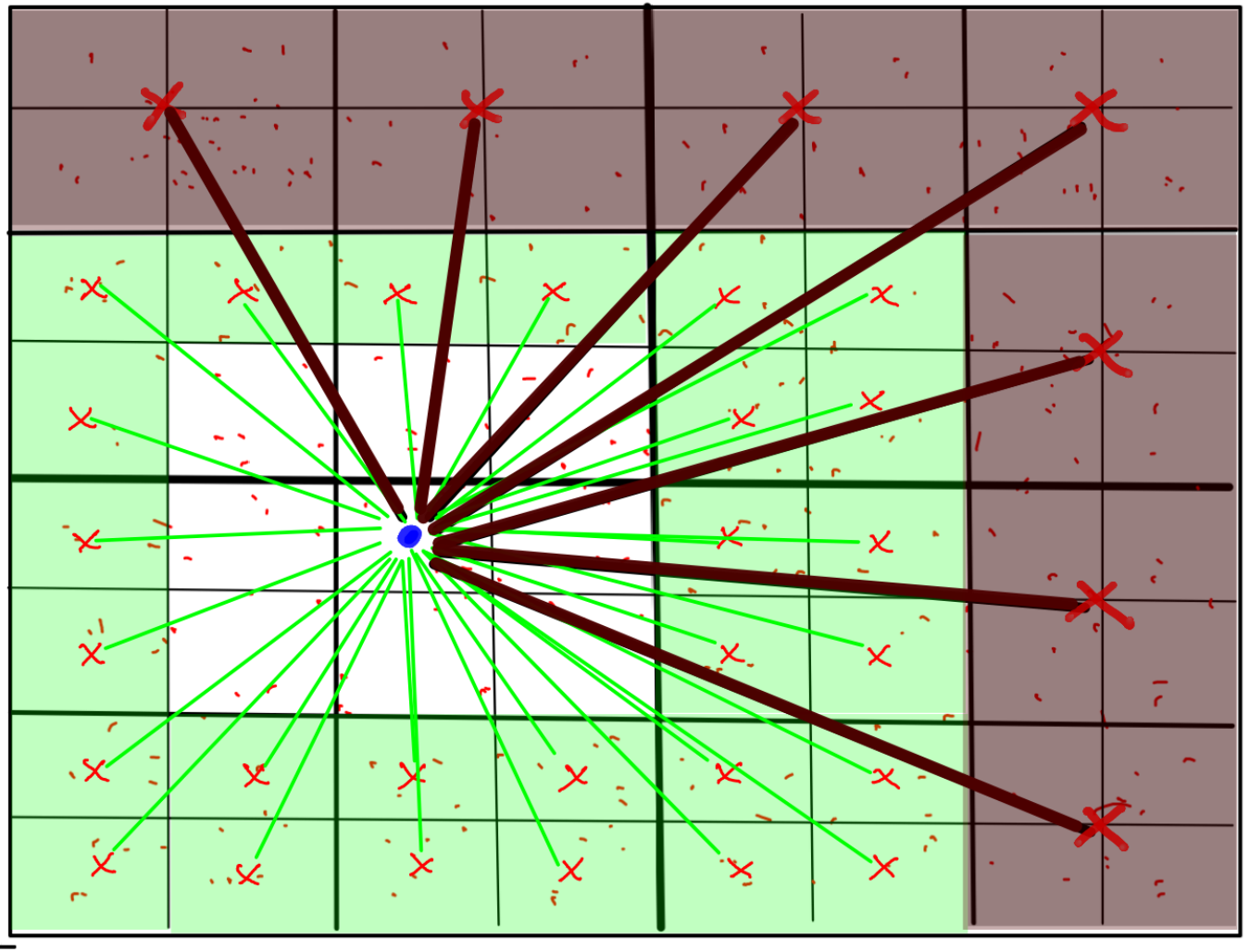
Cost = $O(N_s)$

Idea2: apply center of mass approximation hierarchically

Step3:
Evaluate target potentials using the coarsest level center of mass that is well-separated.
(At most 27 interactions in each level.)
cost = $N_T \times O(27 \log N_S)$

Step4:
Evaluate remaining interactions directly.
cost = $N_T \times O(1)$

Total cost = $O(N \log N)$



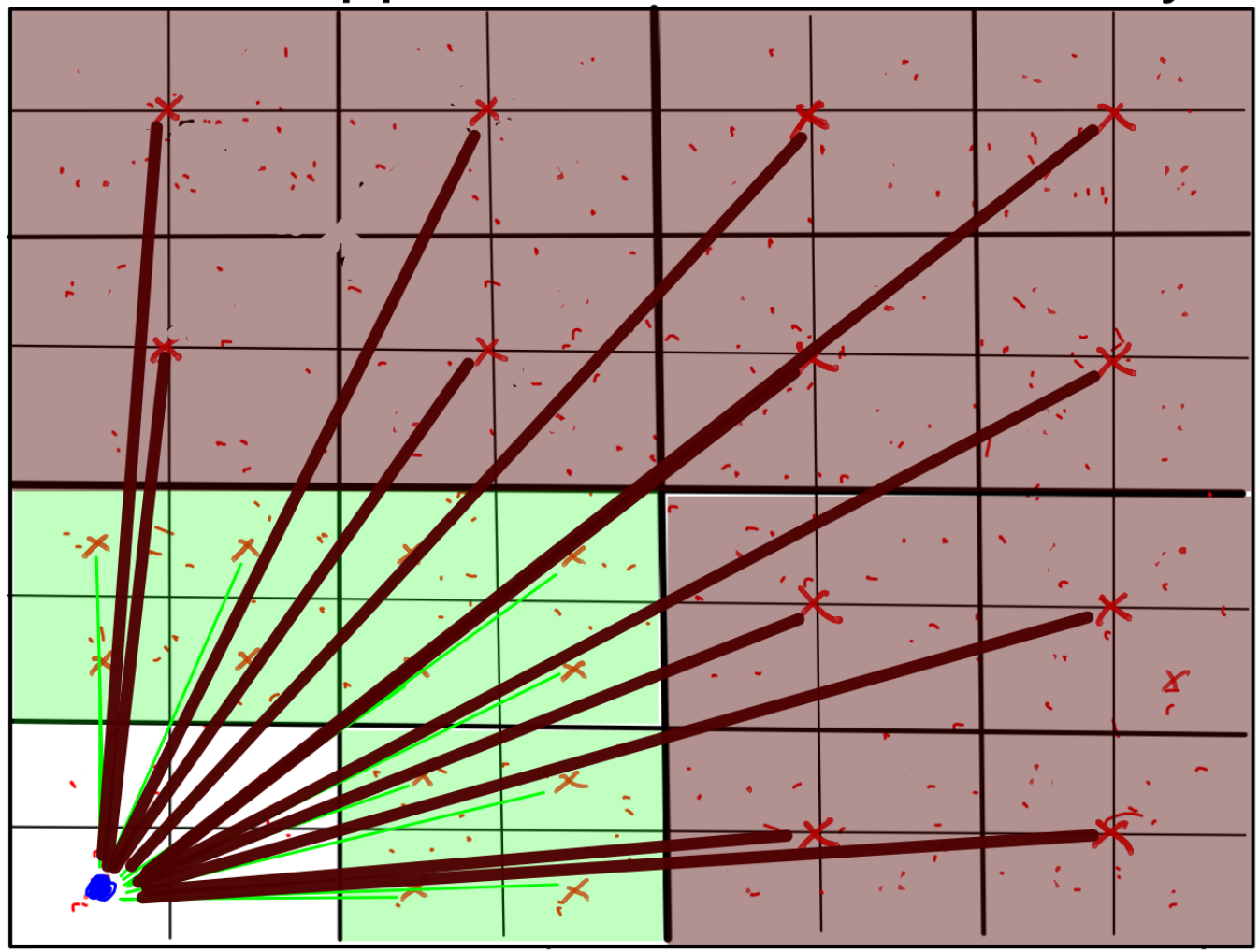
Barnes & Hut: Tree Code (1986)

Idea2: apply center of mass approximation hierarchically

Step3:
Evaluate target potentials using the coarsest level center of mass that is well-separated.
(At most 27 interactions in each level.
$cost = N_T \times O(27 \log N_S)$

Step4:
Evaluate remaining interactions directly.
$cost = N_T \times O(1)$

Total cost $= O(N \log N)$



Barnes & Hut: Tree Code (1986)

Idea3: local expansions

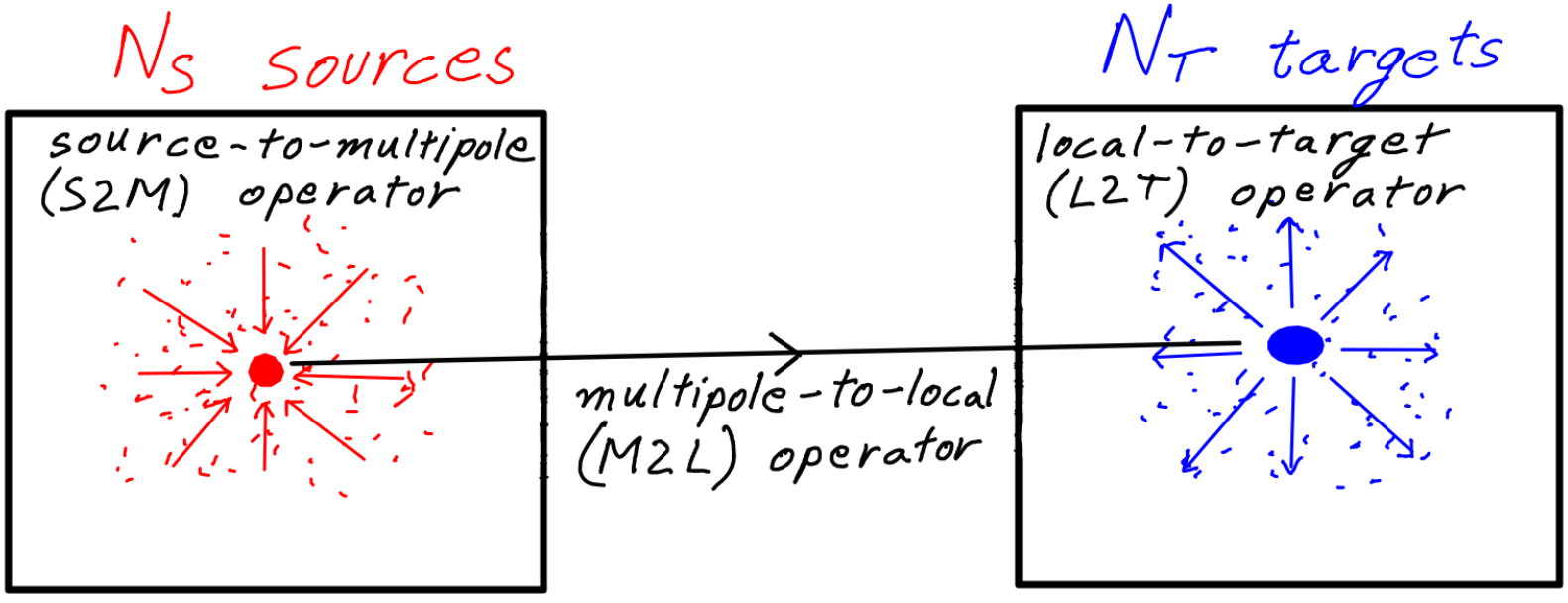
$$u_i = \sum_j G_{ij} f_j$$

Barnes & Hut

$G_{ij} \approx$ [1 1 1 ..... 1] → $G(x_i - y_c)$

FMM

$G_{ij} \approx$ [1 1 1 ..... 1] → $G(x_c - y_c)$

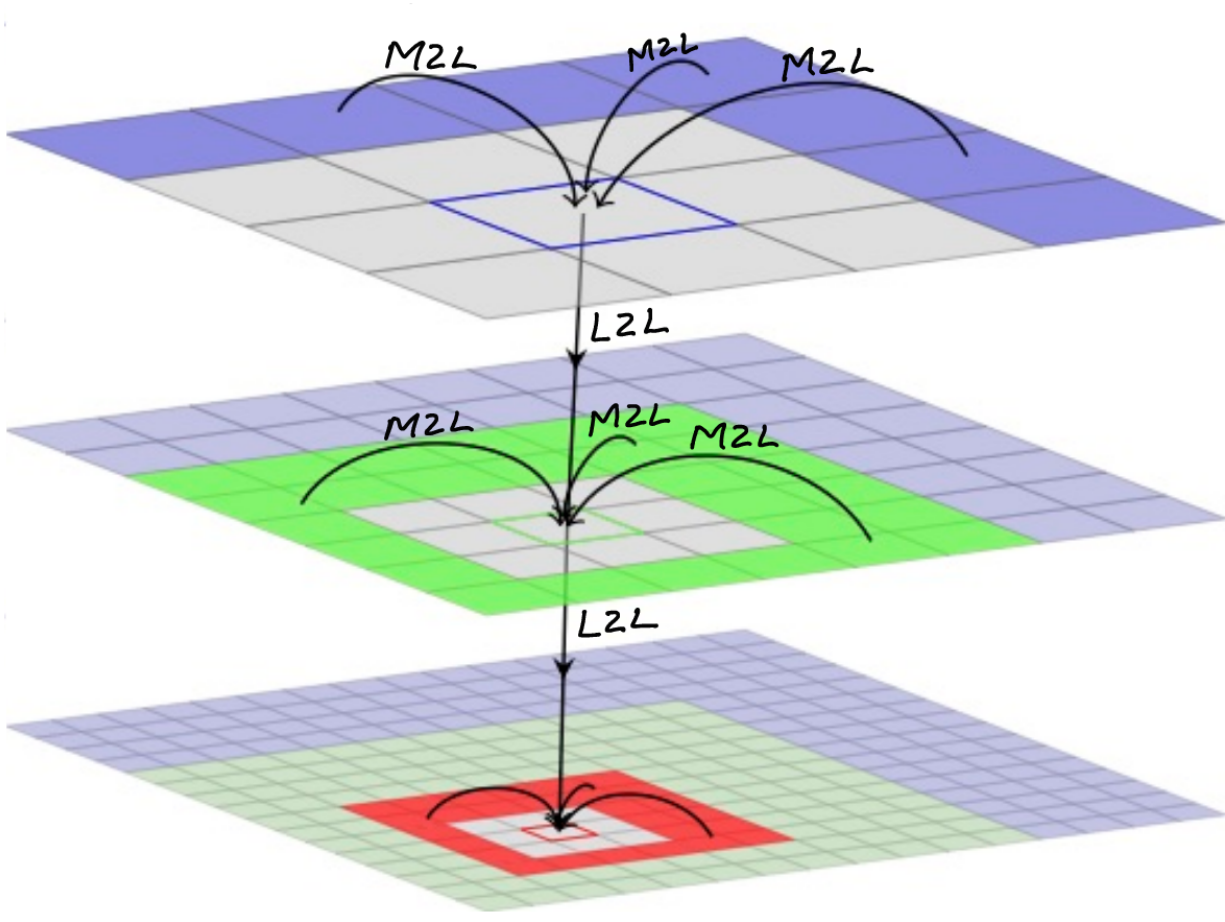


Idea3: local expansions (hierarchically)

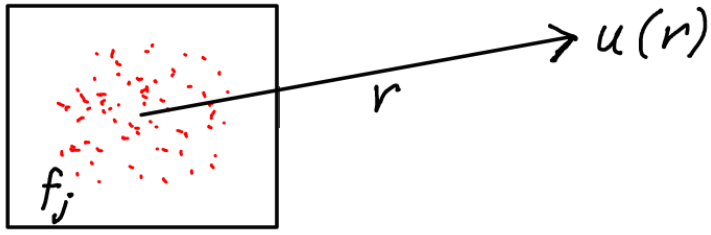
Step0: Build tree and multipole expansions (center of mass) for all leaf & non-leaf nodes. (Upward-pass)

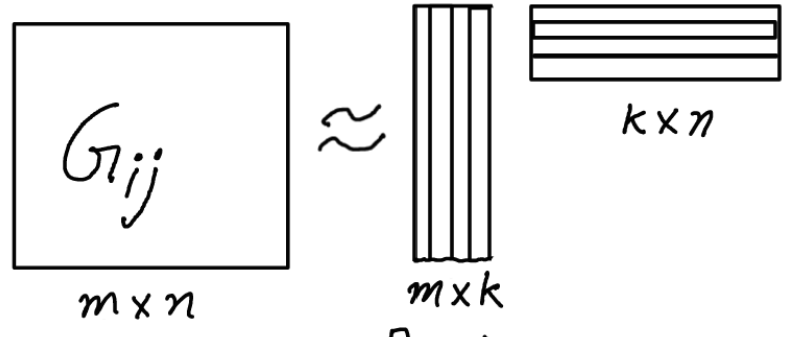
Step1: starting from the coarsest level in the tree, compute local expansions fom the multipole expansions of well-separated nodes.

Step2: propagate local expansions from parents to the children at the next finer level in the tree (local-to-local or L2T operation)
Repeat step1 for the finer level.



Idea4: higher order expansions





Spherical harmonics

$$u(r) \approx \sum_{n=0}^{k} \sum_{m=-n}^{n} \underbrace{\frac{Y_n^m(\theta,\phi)}{r^{n+1}}}_{\text{Basis vectors}} M_n^m$$

(Cheng, Greengard, Rokhlin - 1999)

$$u(r) = \sum_k \underbrace{G(r - y_k)}_{\text{Basis vectors}} M_k$$

(Kernel Independent FMM
Ying, Biros, Zorin - 2004
parallel implementation pvfmm.org)

$$u(r) = \sum_k T_k(r)\, M_k$$

T_k → Chebyshev basis

(Black Box FMM - Fong, Darve - 2009)

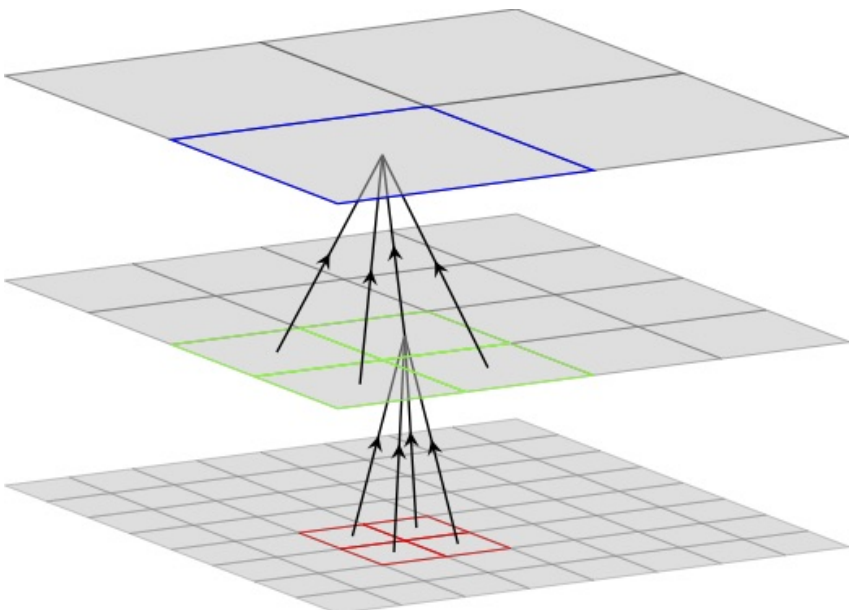
(parallel implementation ExaFMM
Barba, Yokota - 2011)

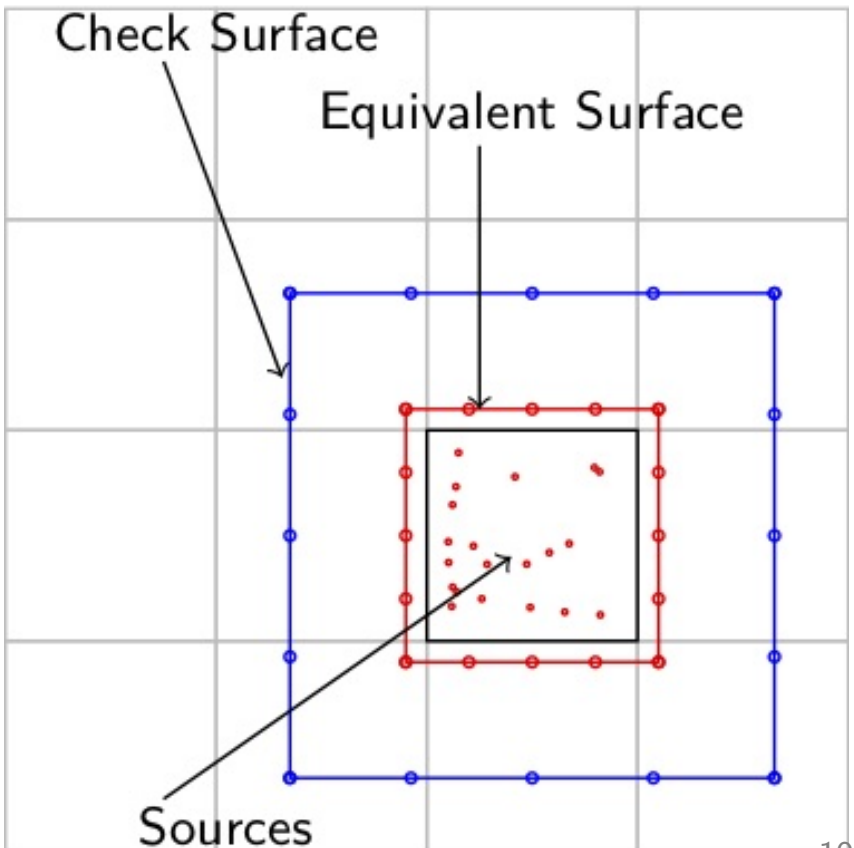
Directional FMM - Ying, Engquist
Askit FMM - March, Yu, Biros et. al.

KIFMM Upward Pass

Multipole Expansion: S2M, M2M

- **Source to Multipole:** from the source points compute multipole expansion of leaf nodes
- **Multipole to Multipole:** from the multipole expansion of children compute multipole expansion of parent node

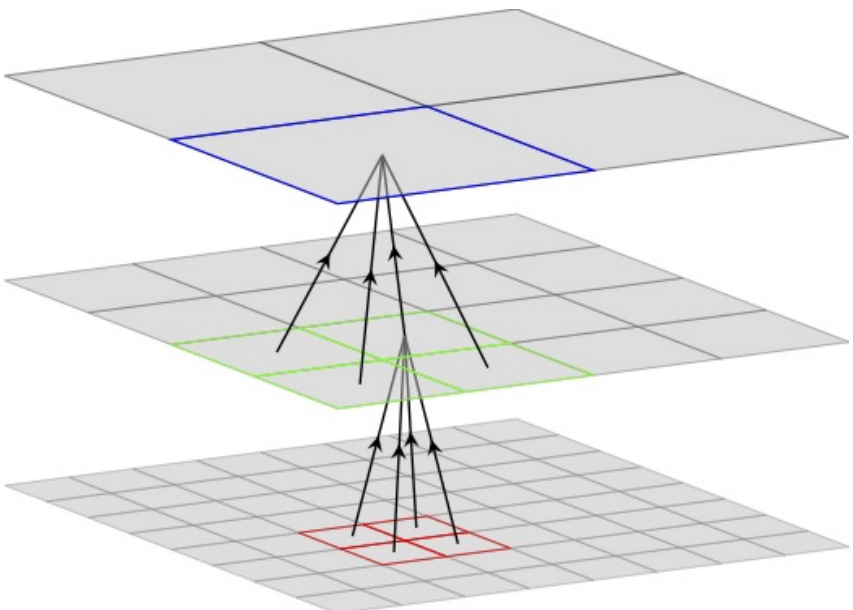


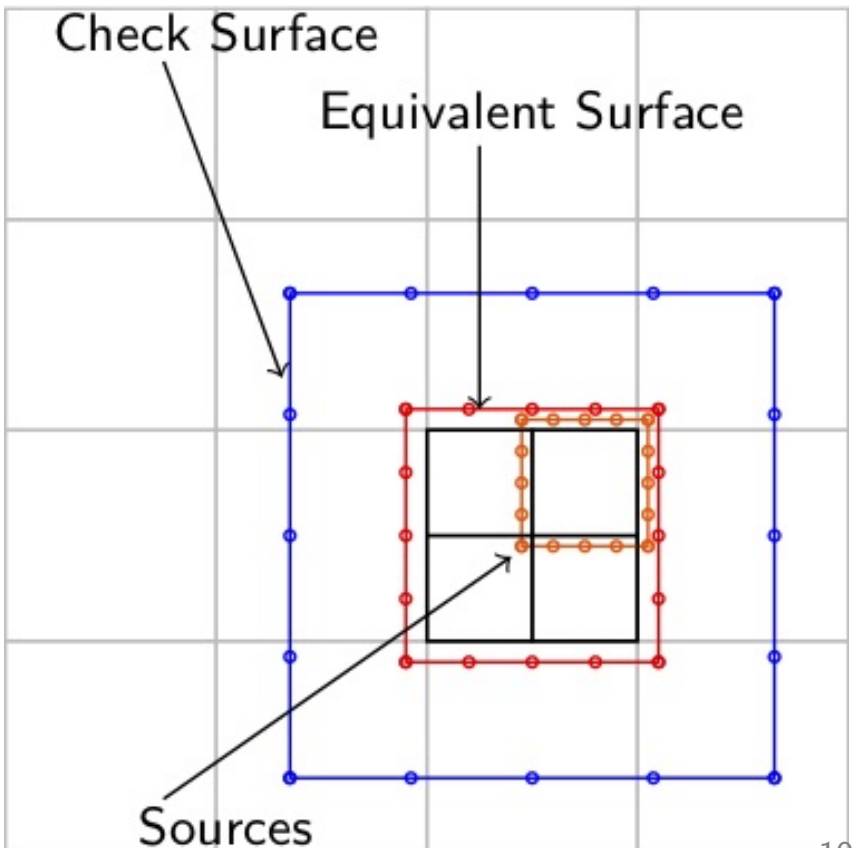


KIFMM Upward Pass

Multipole Expansion: S2M, M2M

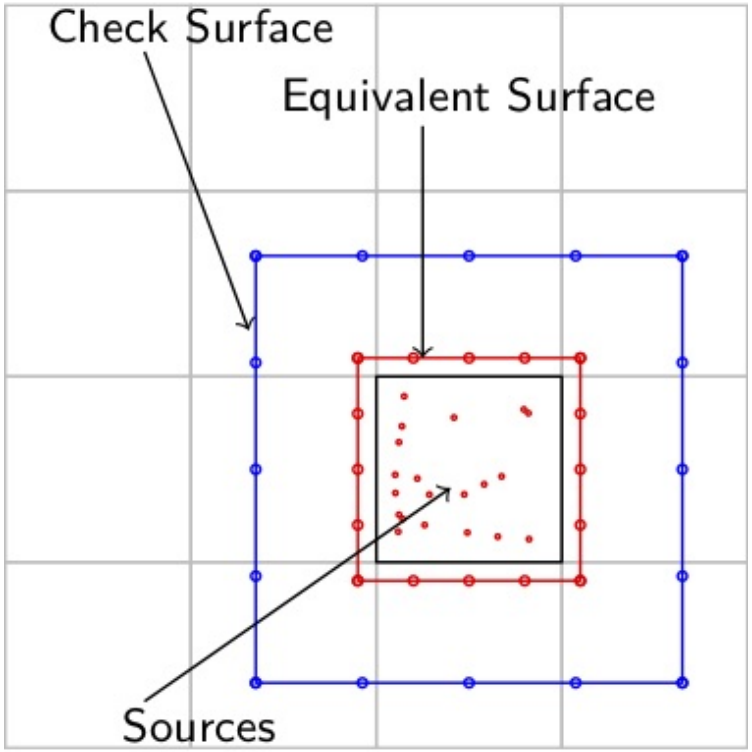
- **Source to Multipole:** from the source points compute multipole expansion of leaf nodes
- **Multipole to Multipole:** from the multipole expansion of children compute multipole expansion of parent node

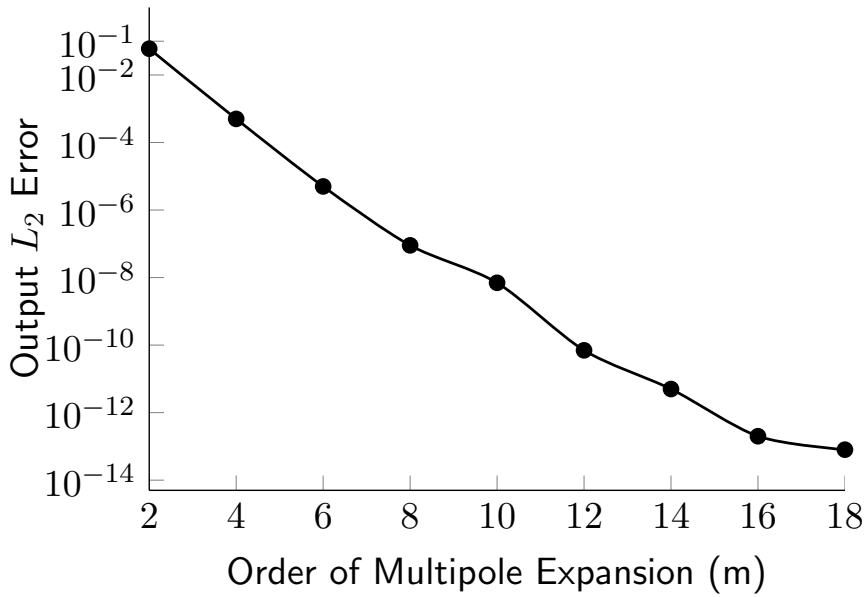




KIFMM Multipole Expansion

convergence with number of multipole points (m^2 in 3D)



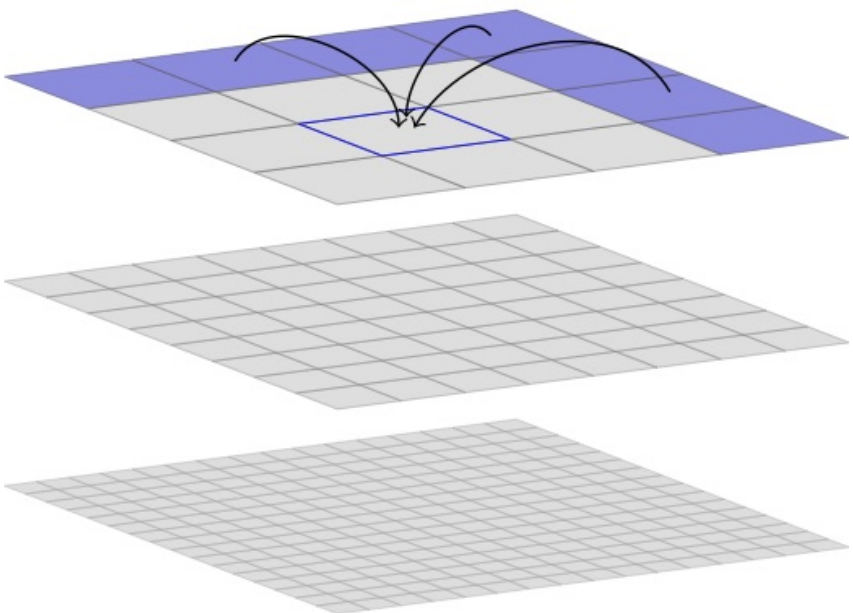


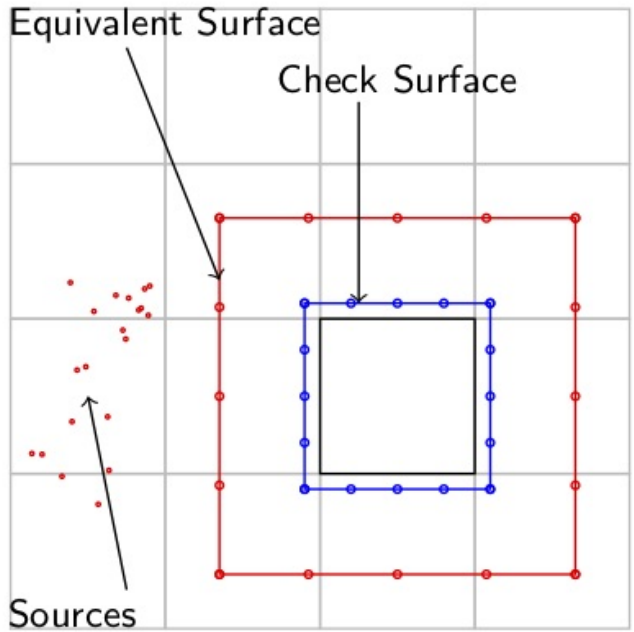
Spectral convergence with multipole order m

KIFMM Downward Pass

Far Interactions: M2L + L2L

- **Multipole-to-Local:** compute local expansion from multipole expansions of well separated nodes
- **Local-to-Local:** add contributions from the parent's local expansion

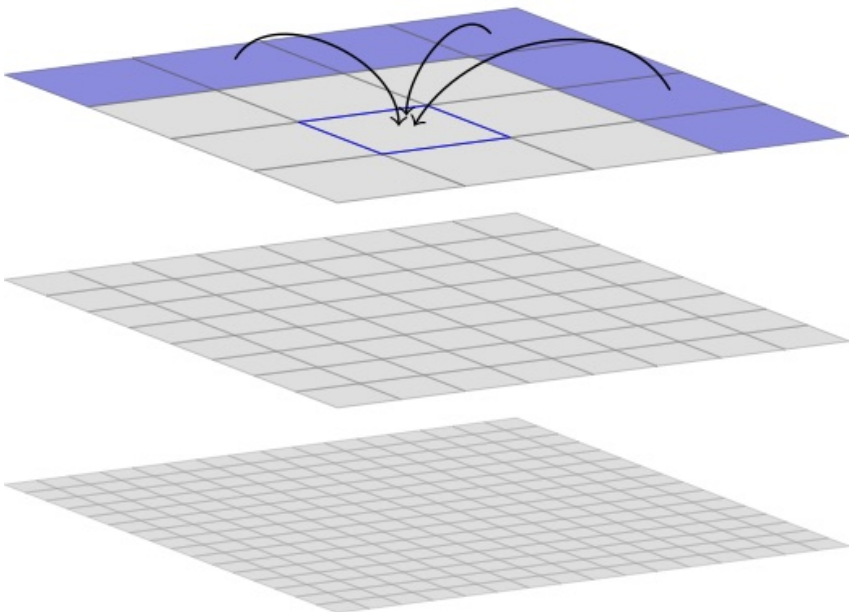


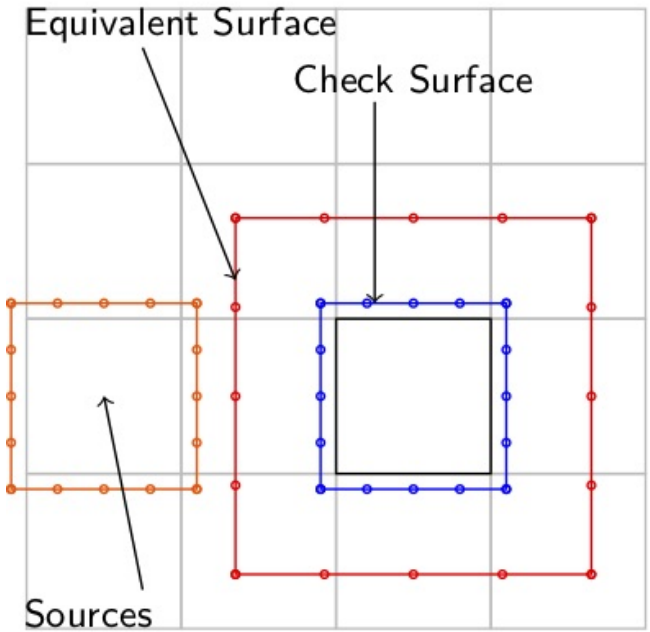


KIFMM Downward Pass

Far Interactions: M2L + L2L

- **Multipole-to-Local:** compute local expansion from multipole expansions of well separated nodes
- **Local-to-Local:** add contributions from the parent's local expansion

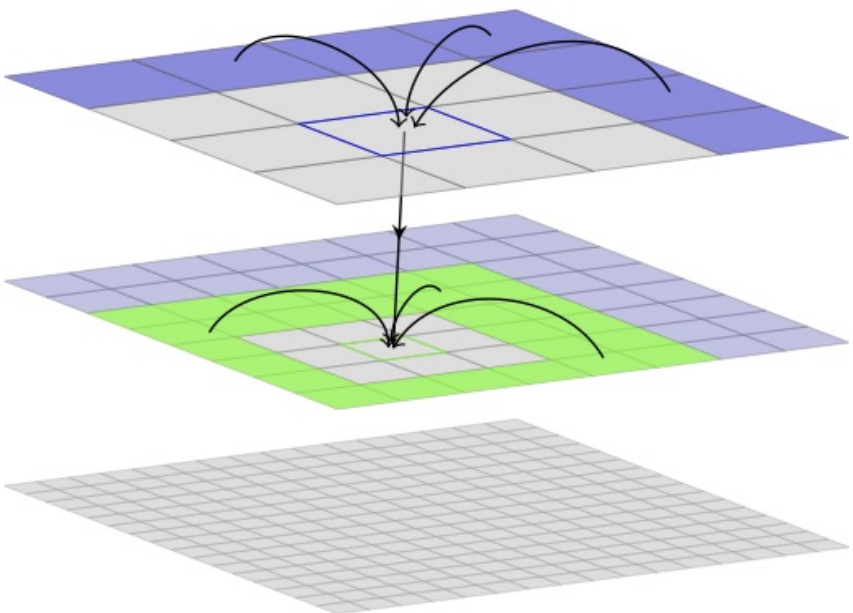


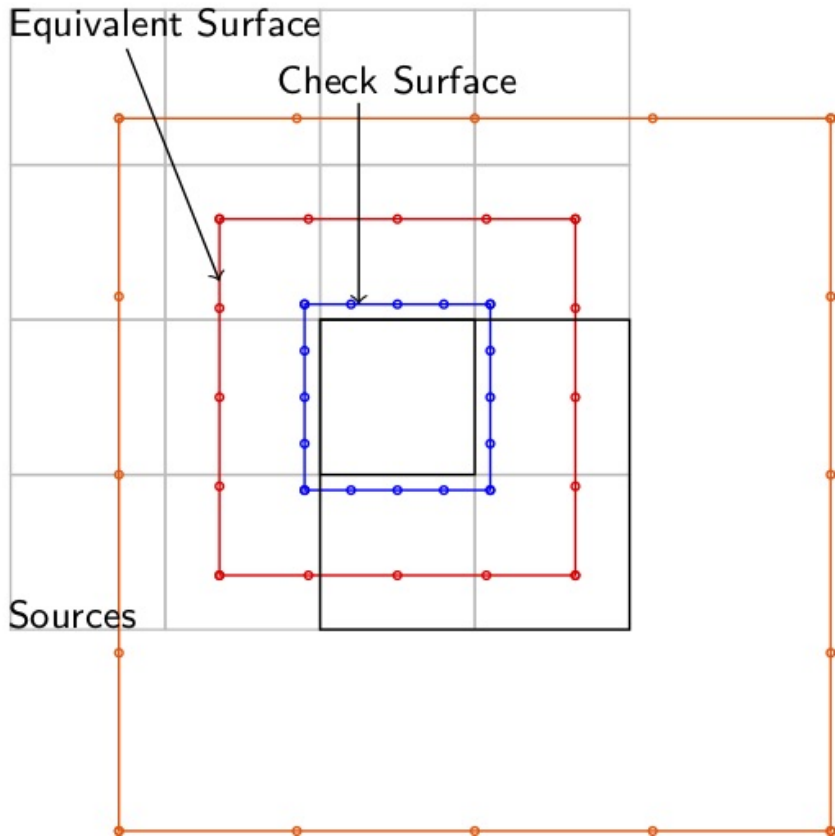


KIFMM Downward Pass

Far Interactions: M2L + L2L

- **Multipole-to-Local:** compute local expansion from multipole expansions of well separated nodes
- **Local-to-Local:** add contributions from the parent's local expansion

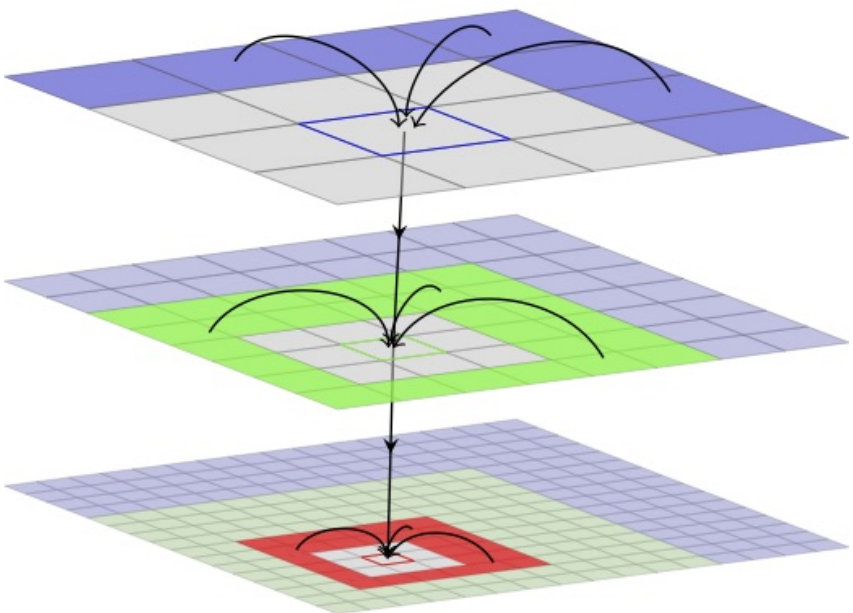


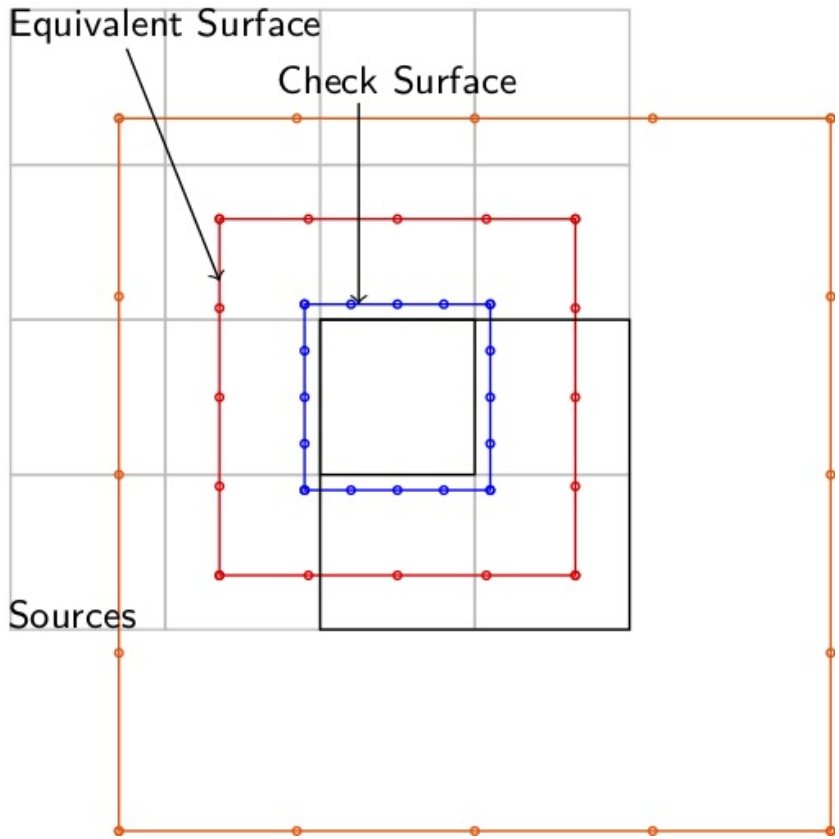


KIFMM Downward Pass

Far Interactions: M2L + L2L

- **Multipole-to-Local:** compute local expansion from multipole expansions of well separated nodes
- **Local-to-Local:** add contributions from the parent's local expansion



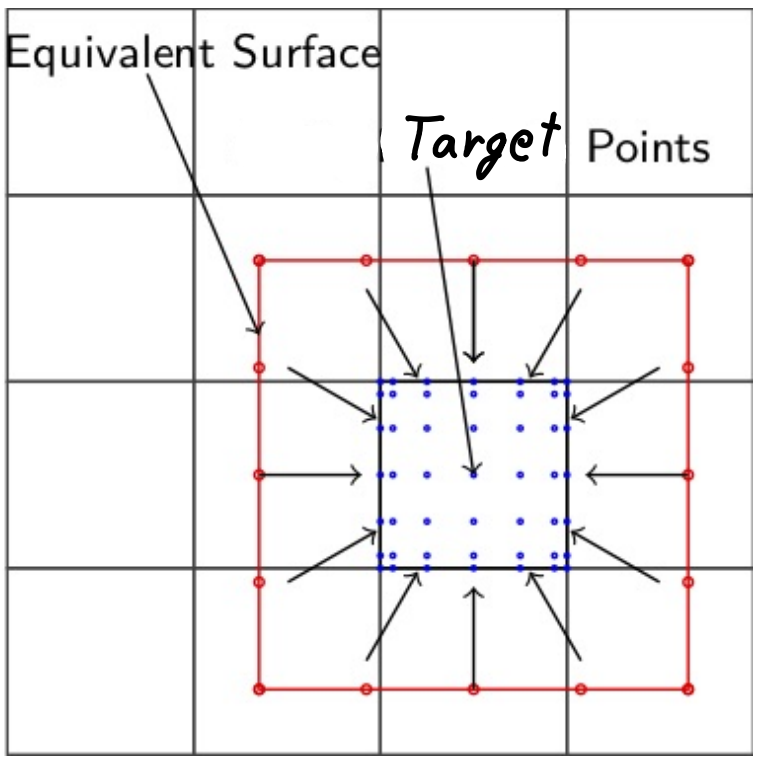


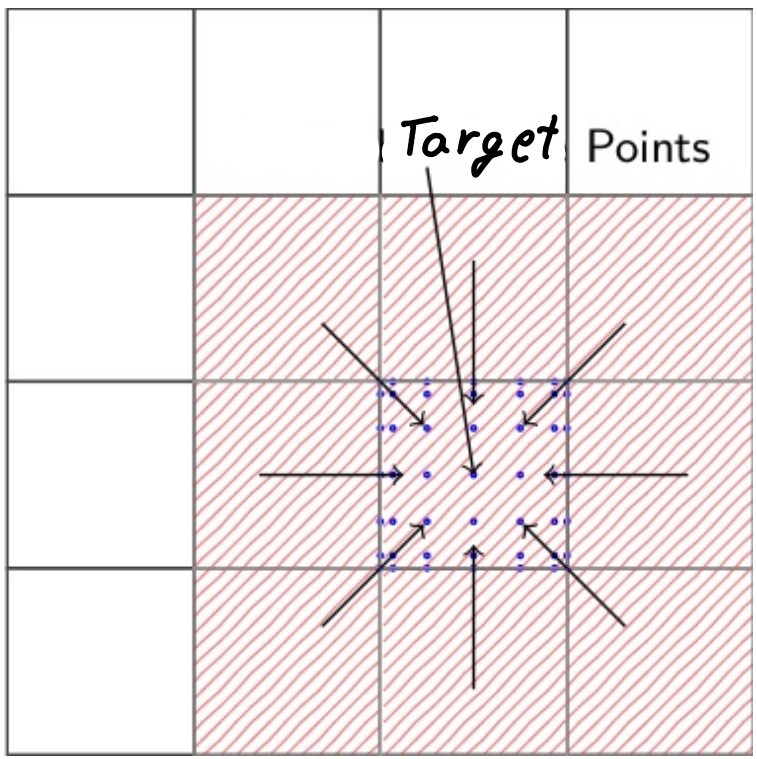
KIFMM Downward Pass

Target Potential: L2T + S2T

At leaf nodes,

- **Local to Target:** evaluate local expansion at target points
- **Source to Target:** add near interactions from neighbouring leaf nodes and itself using direct sum



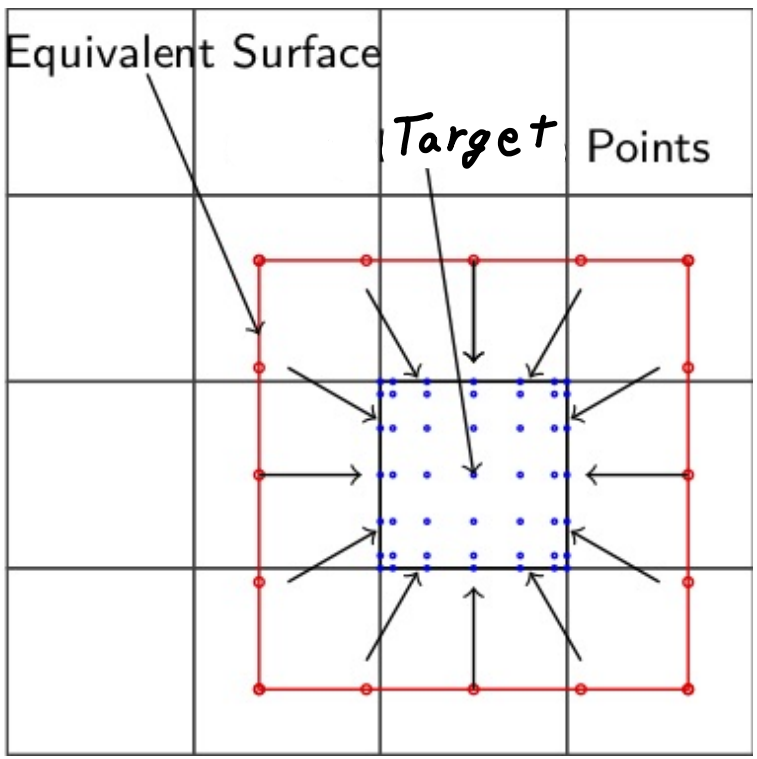


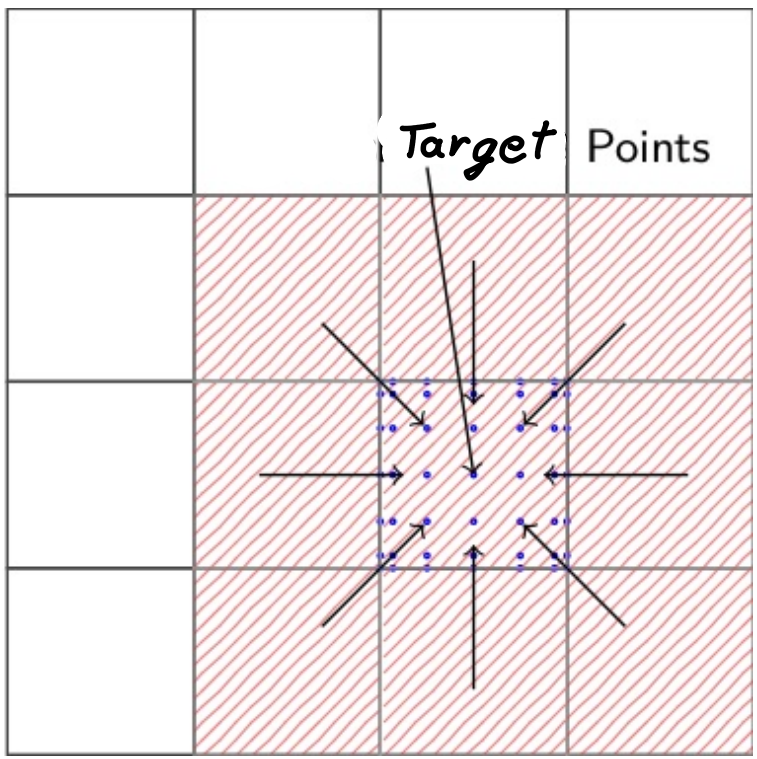
KIFMM Downward Pass

Target Potential: L2T + S2T

At leaf nodes,

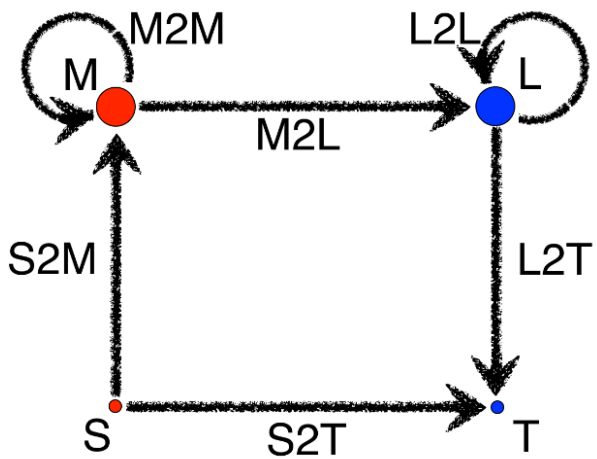
- **Local to Target:** evaluate local expansion at target points
- **Source to Target:** add near interactions from neighbouring leaf nodes and itself using direct sum





FMM Summary

- Upward pass:
  - S2M: build multipole expansions for leaf nodes.
  - M2M: build multipole expansions for non-leaf nodes.
- Downward pass:
  - M2L: build local expansions from multipole expansions on the same level
  - L2L: add local contribution from parent's local expansion
  - L2T: evaluate local expansion at target points of leaf nodes
  - S2T: compute remaining near interactions directly



- S: "Source"
- T: "Target"
- M: "Multipole"
- L: "Local"

Fast Multipole Method (References)

- Lexing Ying's talk on FMM
- A short course on fast multipole methods