JÜLICH
FORSCHUNGSZENTRUM

Implementation and evaluation of parallel integrators for the FMM

Guest Student Programme

September 27, 2010 | Valentina Banciu Advisers: Oliver Bücker Ivo Kabadshow

Overview

- Part 1: Introduction
- Part 2: Implementation and Evaluation of Integrators
- Part 3: Fast Multipole Method
- Part 4: Conclusion and Outlook

JÜLICH
FORSCHUNGSZENTRUM

Implementation and evaluation of parallel integrators for the FMM

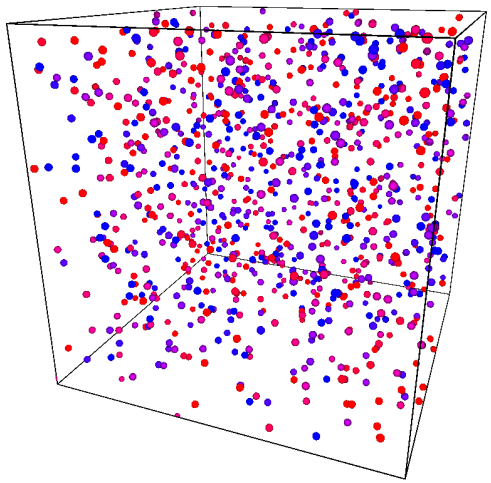
Part I: Introduction

September 27, 2010 | Valentina Banciu Advisers: Oliver Bücker Ivo Kabadshow

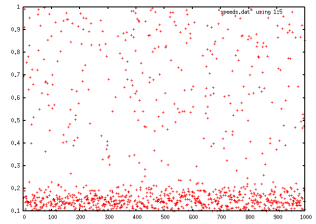
Molecular Dynamics

Molecular dynamics is a form of computer simulation in which atoms and molecules are allowed to interact for a given period of time by approximations of known physics.

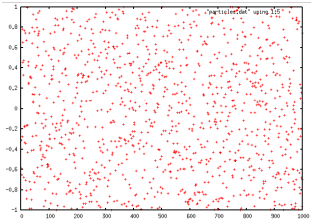
Molecular Dynamics



1000 randomly distributed particles



random speeds

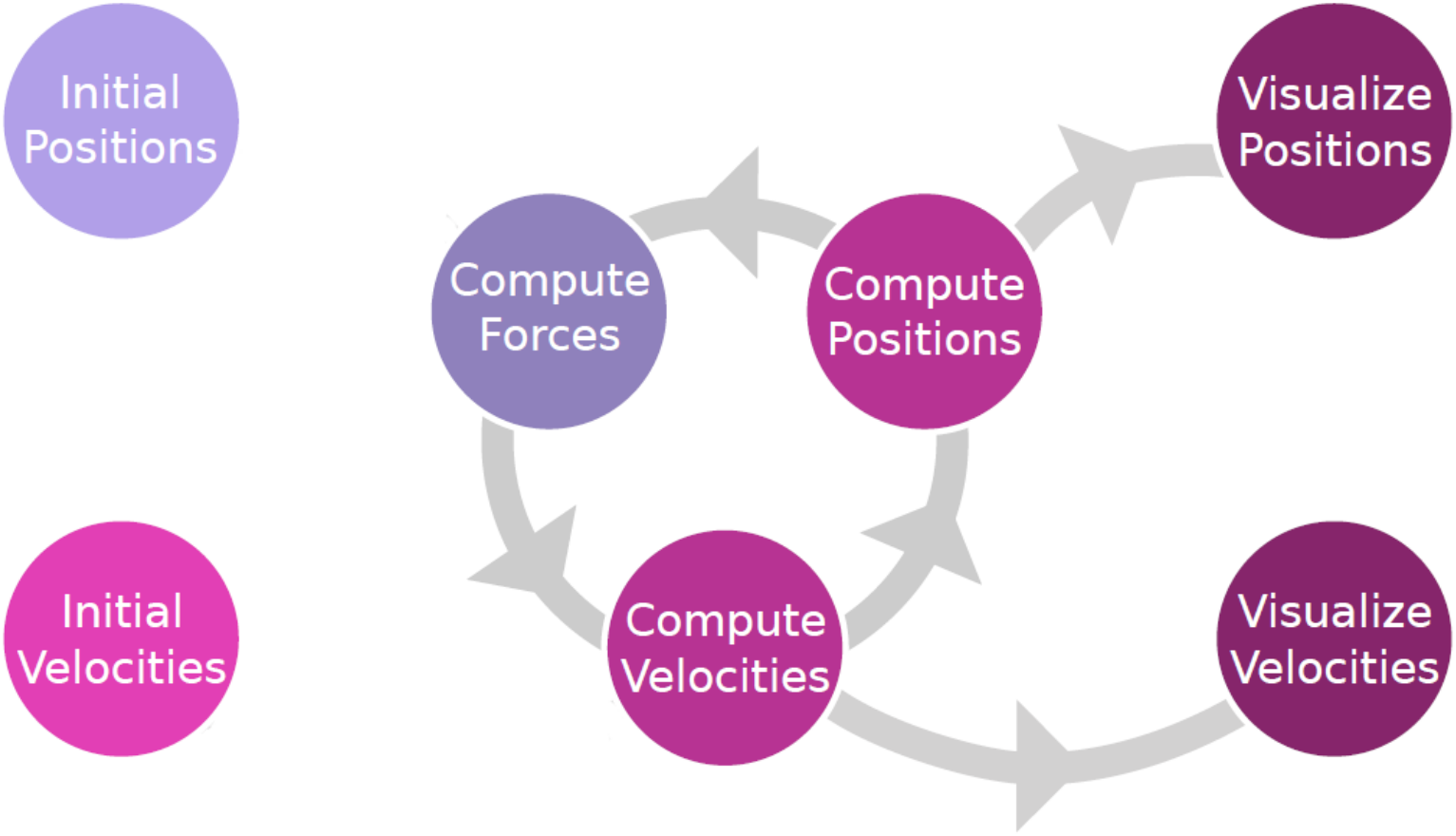


random charges

Simulation time: t

n timesteps: $dt \cdot n = t$

Molecular Dynamics Simulation Loop



JÜLICH
FORSCHUNGSZENTRUM

Implementation and evaluation of parallel integrators for the FMM

Part II: Integrators

September 27, 2010 | Valentina Banciu Advisers: Oliver Bücker Ivo Kabadshow

Evaluation Criteria

A suitable integrator for $N-body$ problems must meet the following requirements:

- Accuracy
- Stability
- Conservativity
- Reversibility
- Effectiveness
- Symplecticity

Classification of Integrators

- First order:
  Euler method, Backward Euler, Semi-implicit Euler
- Second order:
  Leapfrog integration, Verlet integration (position/velocity), Crank-Nicolson method, Beeman's algorithm, Midpoint method, Heun's method
- Higher order:
  Runge-Kutta methods, Linear multistep method

Euler Integration

- The most basic kind of explicit method

$$y'(t) = f(t, y(t)) \qquad\qquad y(t_0) = y_0$$

- One step of the Euler method from t_n to $t_{n+1} = t_n + h$ is:

$$y_{n+1} = y_n + h \cdot f(t_n, y_n)$$

Leapfrog Integration

- A simple method for dynamic systems
- The equations for leapfrog integration can be written:

$$y_{i+1} = y_i + y'_{i+1/2} \cdot h$$
$$y'_{i+1/2} = y'_{i-1/2} + y''_i \cdot h$$

- The equations can be manipulated into a form which writes velocity at integer steps as:

$$y_{i+1} = y_i + y'_i \cdot h + y''_i \frac{h^2}{2}$$
$$y'_{i+1} = y'_i + \frac{y''_i + y''_{i+1}}{2} h$$

Runge-Kutta Integration

- The n-th order explicit Runge-Kutta scheme to advance a set of differential equations $y'(t) = f(t, y(t))$ over a step h is:

$$y(h) = y_0 + \sum_{j=1}^{n} w_j k_j$$

$$k_j = h \cdot f(t_i, y_0 + \sum_{i=1}^{j-1} \beta_{ji} k_i)$$

$$\alpha_j = \sum_{i=1}^{j-1} \beta_{ji}$$

$$\sum_{j=1}^{n} w_j = 1$$

Low-Storage Runge-Kutta

Second Order

$$q_j = a_j q_{j-1} + h \cdot f(t_{j-1}, y_{j-1})$$
$$y_j = y_{j-1} + b_j q_j$$

Coefficients ($n = 2$)

$$b_1 = \beta_{21} = \alpha_2 \qquad a_1 = 0$$
$$b_2 = w_2 = \frac{1}{2}\alpha_2^{-1} \qquad a_2 = \frac{w_1 - b_1}{w_2}$$

Error Estimation

$$e_j = (y_j - y_{j-1}) - b_j q_j$$

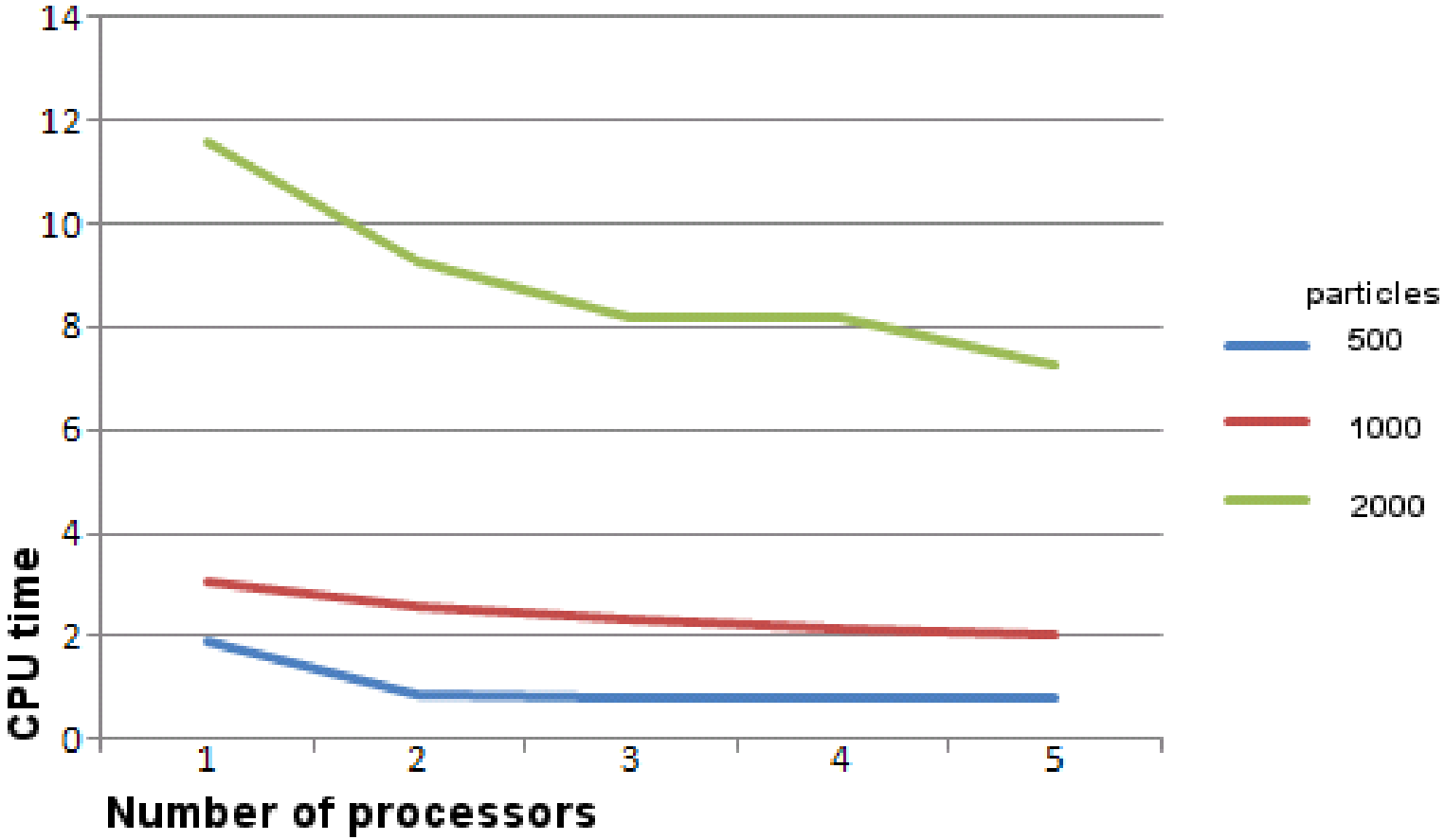
Low-Storage Runge-Kutta

Discussion

α_2	w_1	w_2	roundoff error	truncation error	remarks
$(\frac{1}{2})^{1/2}$	$1-(\frac{1}{2})^{1/2}$	$(\frac{1}{2})^{1/2}$	0.414	0.374	least roundoff error
$2/3$	$1/4$	$3/4$	0.417	0.333	least truncation error
$1/2$	0	1	n/a	0.500	classic 2nd order

[Journal of Computational Physics **35**, 48-56 (1980)]

Leapfrog Integration



JÜLICH
FORSCHUNGSZENTRUM

Implementation and evaluation of parallel integrators for the FMM

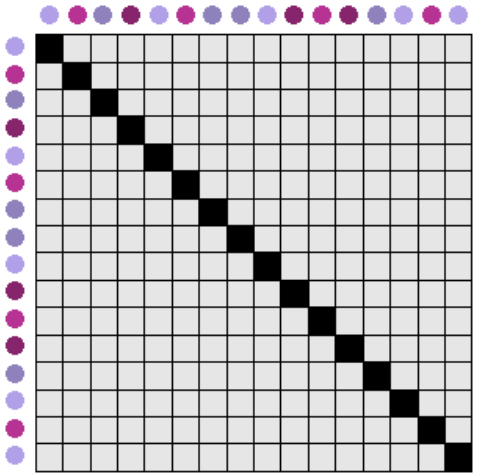
Part III: Fast Multipole Method

September 27, 2010 | Valentina Banciu Advisers: Oliver Bücker Ivo Kabadshow

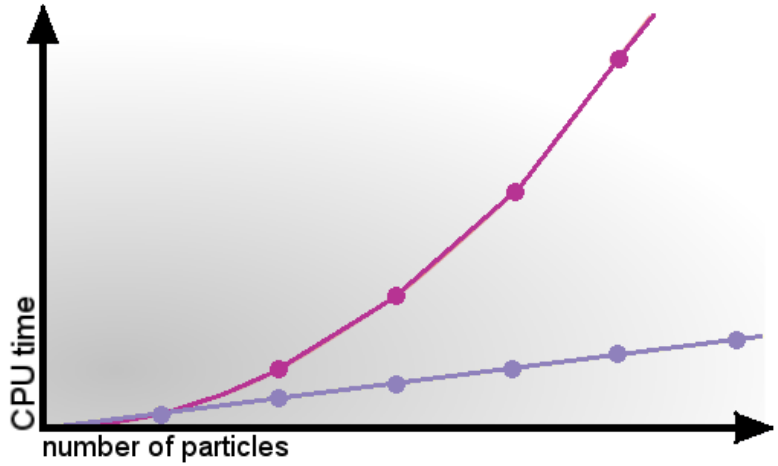
Coulomb Problem

- Computational complexity of $O(N^2)$
  - $E = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{q_i q_j}{r_{ij}} \quad (i \neq j)$
  - $F(r_j) = q_j \sum_{i=1}^{N} \frac{q_i}{r_{ij}^3} r_{ij} \quad (i \neq j)$

Particle Pairwise Interactions



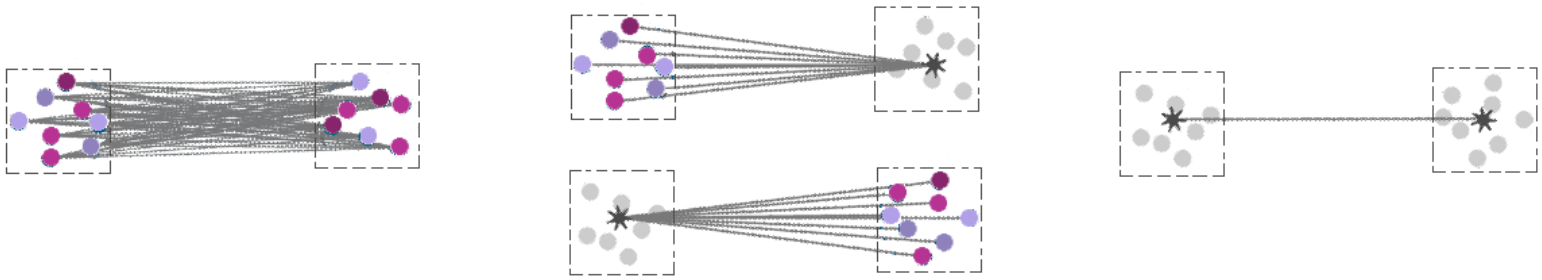
Direct Interaction



$O(N^2)$ *complexity*

How to Reduce Complexity?

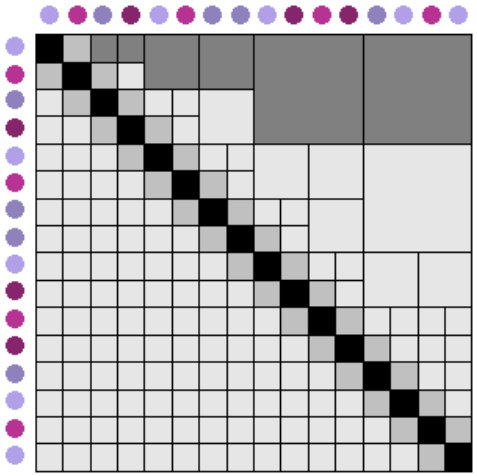
Reducing the number of interactions



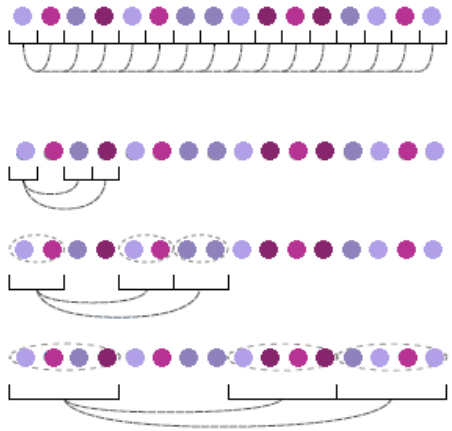
Grouped target and source particles

1 pseudo-particle - pseudo-particle interaction

Sneak Peak: FMM

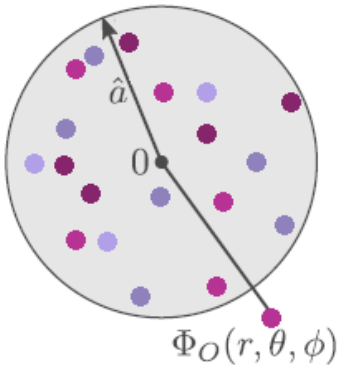


FMM Interaction



Particle groups

Multipole and Taylor-like Expansions



Multipole expansion

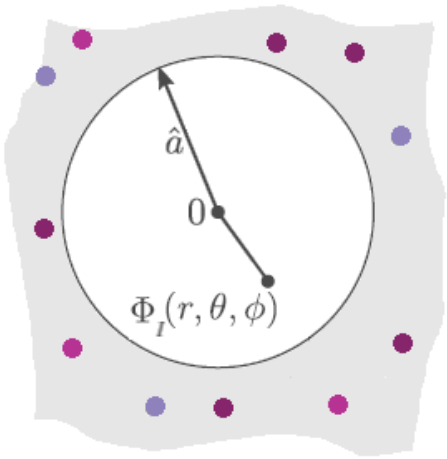
$$\Phi_O(P) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \omega_{lm}(q,a) \frac{1}{r^{l+1}} \tilde{\tilde{P}}_{lm}(cos\theta) e^{im\phi}$$

$$= \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \omega_{lm}(q,a) M_{lm}(r)$$

Multipole and Taylor-like Expansions

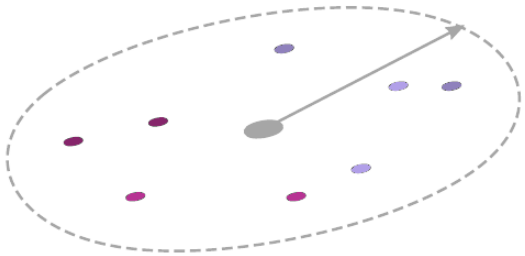
$$\Phi_l(P) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \mu_{lm}(q,r) a^l \tilde{P}_{lm}(cos\alpha) e^{-im\beta}$$

$$= \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \mu_{lm}(q,r) O_{lm}(a)$$



Taylor-like expansion

Expanding Particles into Multipoles

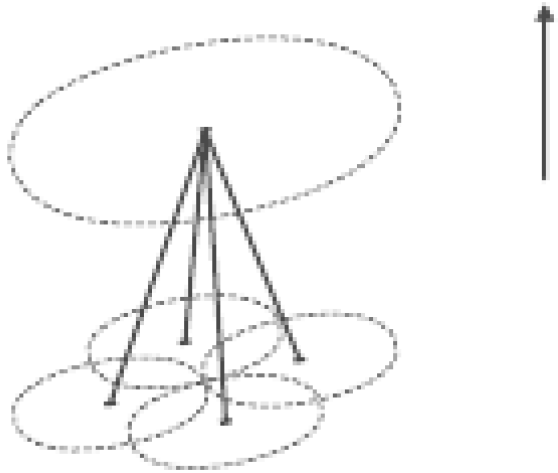


Setup Multipole Moments

Particles inside each box are expanded aroud the box center

$$\omega_{lm} = \sum_{j=1}^{N_{box}} q_j a_j^l P_{lm}(cos\alpha_j) e^{-im\phi_j}$$

Multipole2Multipole Operator (M2M)

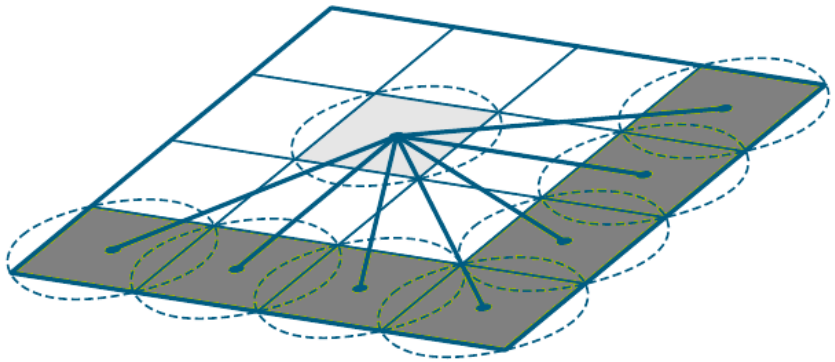


Vertical Operator

Shifts multipole expansions from level L to $L-1$ up the tree

$$\omega_{lm}(a+b) = \sum_{j=0}^{l} \sum_{k=-j}^{j} O_{l-j,m-k}(b)\omega_{jk}(a)$$

Multipole2Local Operator (M2L)

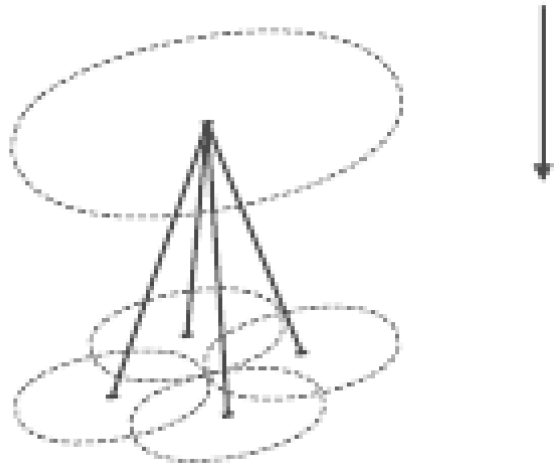


Horizontal Operator

Transforms remote multipole expansions into local Taylor-like expansions on level L

$$\mu_{lm}(R_i - a) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} M_{l+j,m+k}(R_i)\omega_{jk}(a)$$

Local2Local Operator (L2L)



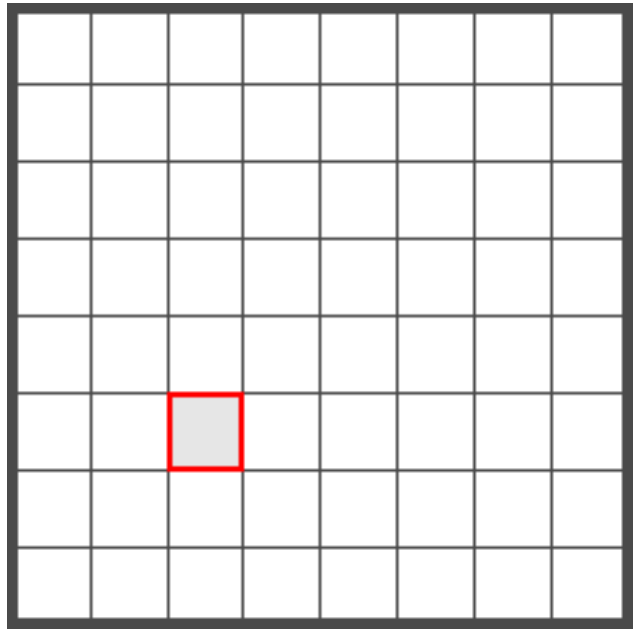
Vertical Operator

Shifts taylor-like expansions from level $L-1$ to L down the tree

$$\mu_{lm}(r-b) = \sum_{j=l}^{p} \sum_{k=-j}^{j} O_{j-l,k-m}(b)\mu_{jk}(r)$$

Interaction Sets

Near Field (*NF*) and Far Field (*FF*)

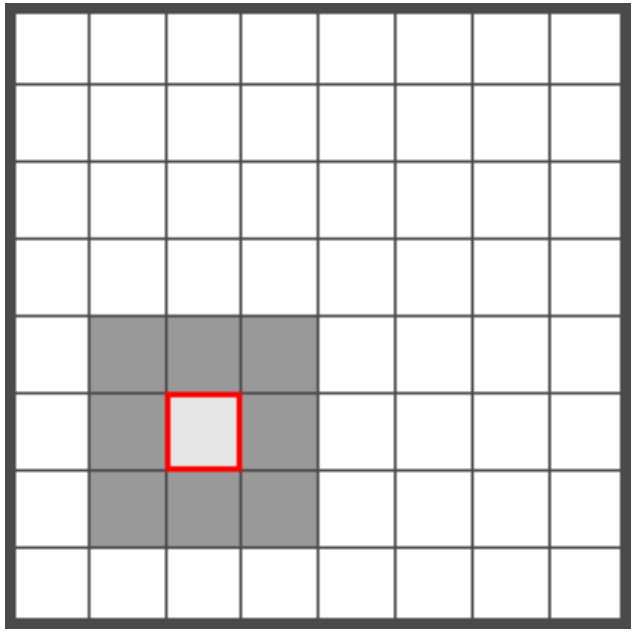


Interactions for box *A* on level *L*

The red box contains a multipole expansion of box *A*

Interaction Sets

Near Field (*NF*) and Far Field (*FF*)

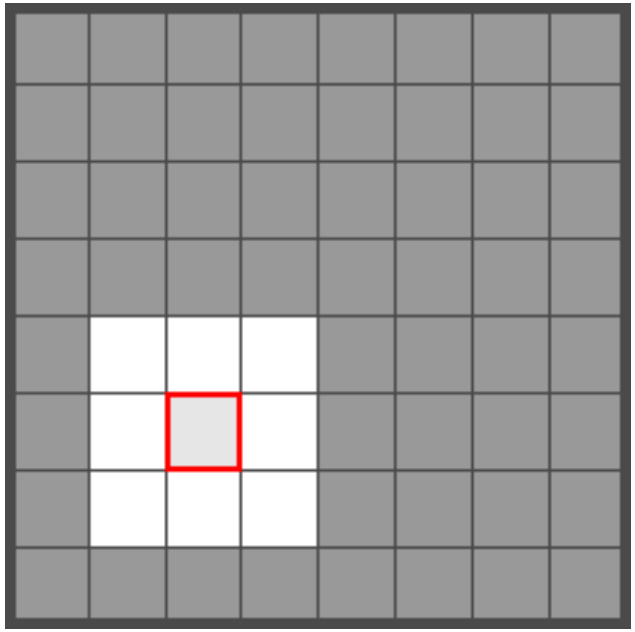


Interactions for box *A* on level *L*

Near field interactions are computed only on the lowest level

Interaction Sets

Near Field (*NF*) and Far Field (*FF*)

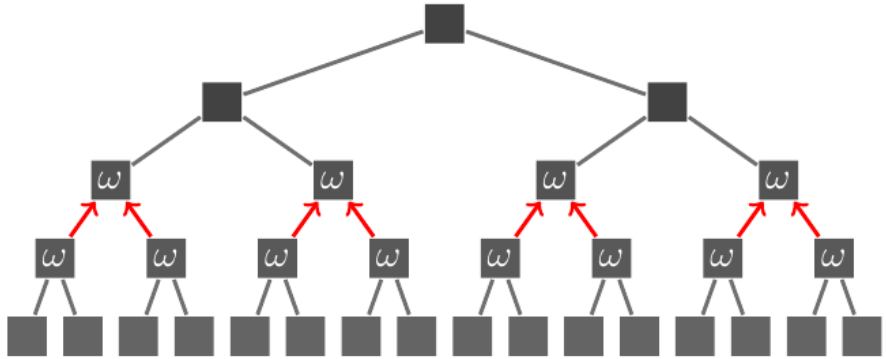


Interactions for box *A* on level *L*

Far field interactions are computed via multipoles

FMM Pass 1

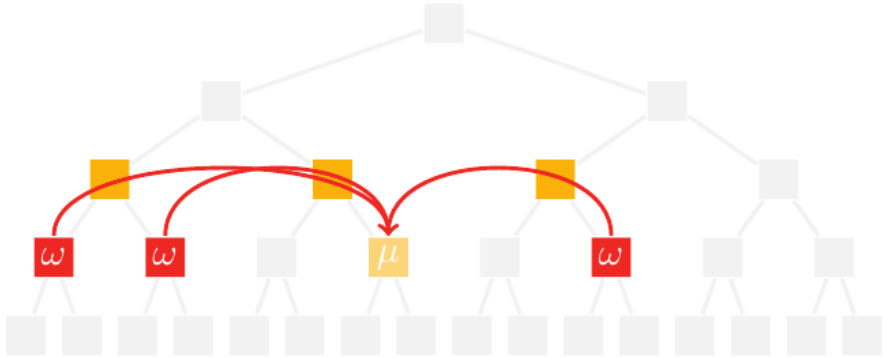
Form and shift Multipole expansions



- Sort particles into lowest level boxes
- Form multipole expansion on the lowest level
- Shift multipole expansion up the tree

FMM Pass 2

Transform distant multipole expansions

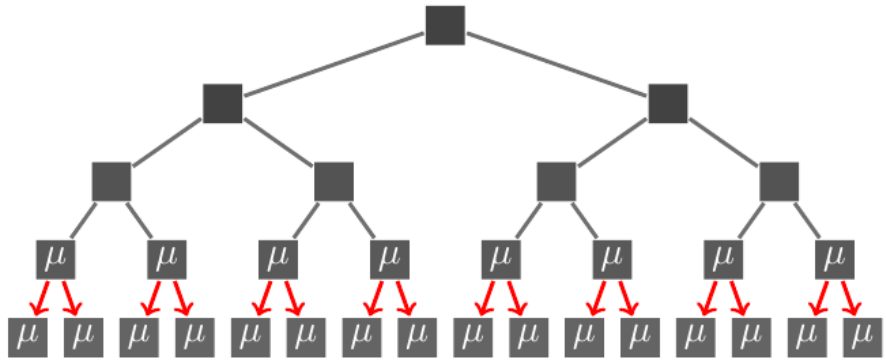


For each level:

- Find parent box
- Choose *ws*-neighbour parent boxes
- Find corresponding child boxes
- Find *ws*-separated child boxes
- Use M2L operator to perform interaction

FMM Pass 3

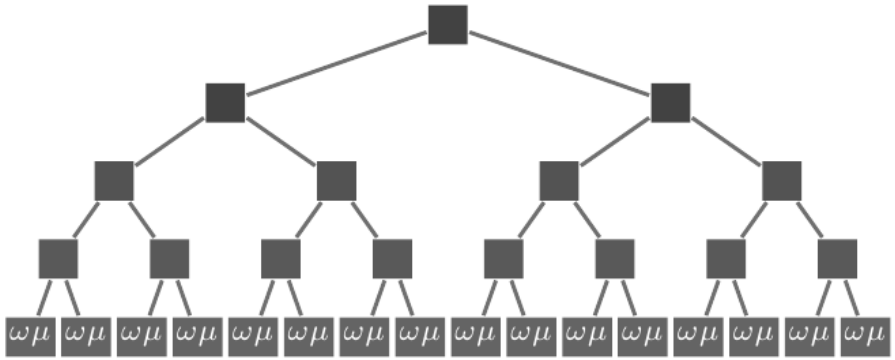
Shift Taylor-like expansions



- Find child boxes
- Shift Taylor-like coefficients down the tree, using L2L operator

FMM Pass 4

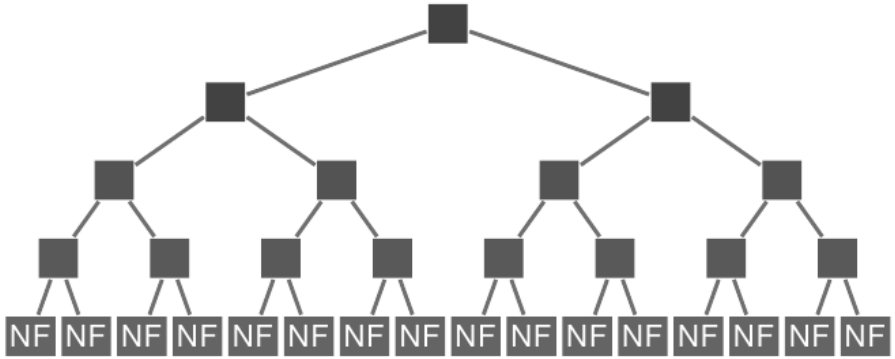
Calculate far field energy, forces and potentials



- All interactions are now present at the lowest level
- Compute far field interactions on the lowest level

FMM Pass 5

Calculate near field energy, forces and potentials



- Particles in the near field are limited $M = const$
- Compute near field interactions on the lowest level
- Add up far field and near field interactions

Total Coulomb Energy

- Far Field Part

$$E_{FF} = \sum_{ibox} \sum_{l=0}^{L} \sum_{m=-l}^{l} \omega_{lm} \mu_{lm}$$

- Near Field Part: Inbox

$$E_{NF_1} = \sum_{ibox} \sum_{i=1}^{N_{ibox}-1} \sum_{j=i+1}^{N_{ibox}} \frac{q_i q_j}{r_{ij}}$$

- Near Field Part: Nearest Neighbours

$$E_{NF_2} = \sum_{ibox} \sum_{jbox} \sum_{i=1}^{N_{ibox}} \sum_{j=1}^{N_{jbox}} \frac{q_i q_j}{r_{ij}}$$

- Total Coulomb Energy

$$E^c_{tot} = E_{FF} + E_{NF_1} + E_{NF_2}$$

FMM Parameter I

Number of Poles P

Infinite Expansion

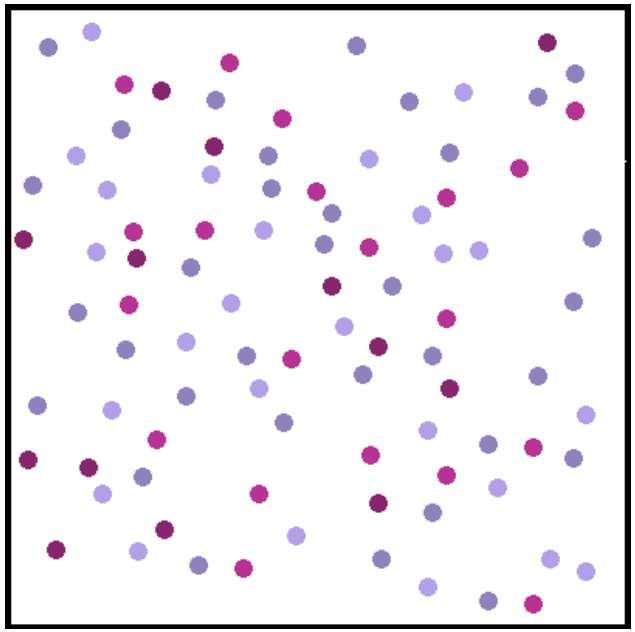
$$\frac{1}{d} = \frac{1}{|r-a|} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \dots$$

Finite Expansion

$$\frac{1}{d} = \frac{1}{|r-a|} \approx \sum_{l=0}^{p} \sum_{m=-l}^{l} \dots$$

FMM Parameter II

Depth d of the FMM Tree

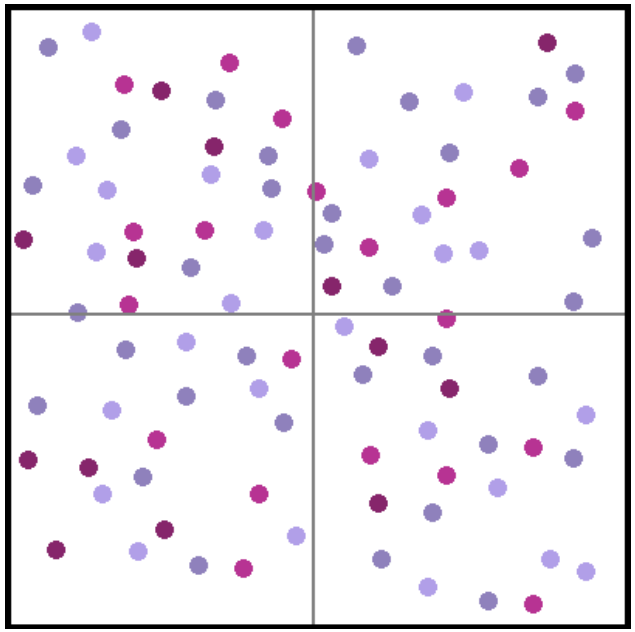


Tree depth d, Level $L = d + 1$

Simulation box divided into 8^d subboxes

FMM Parameter II

Depth d of the FMM Tree

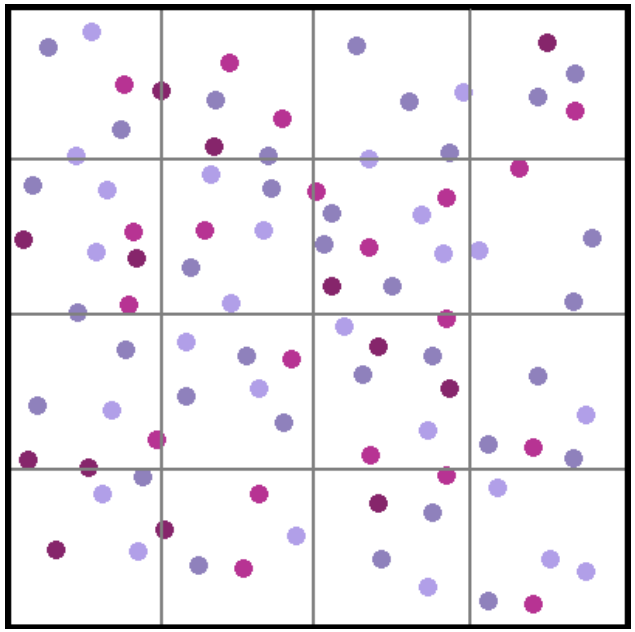


Tree depth d, Level $L = d + 1$

Simulation box divided into 8^d subboxes

FMM Parameter II

Depth d of the FMM Tree

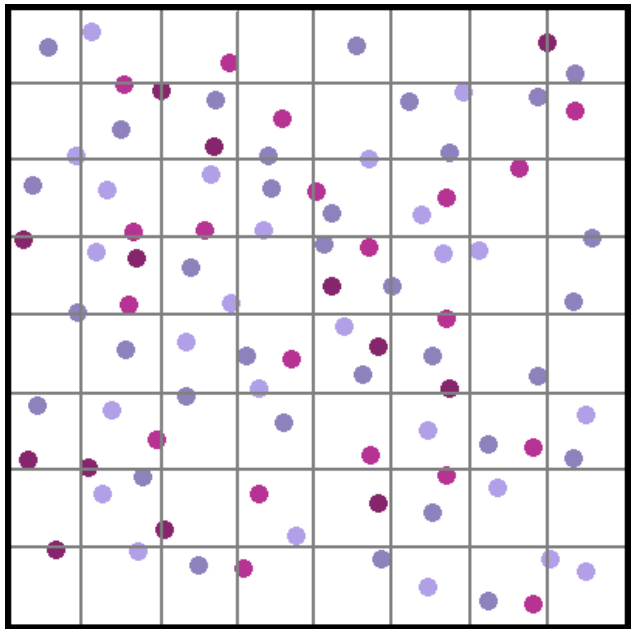


Tree depth d, Level $L = d + 1$

Simulation box divided into 8^d subboxes

FMM Parameter II

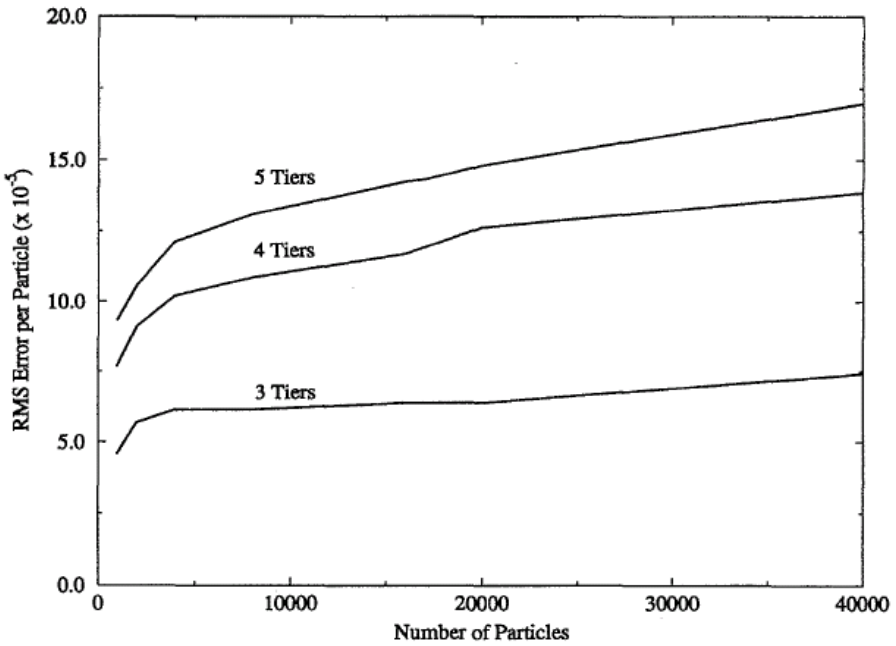
Depth d of the FMM Tree



Tree depth d, Level $L = d + 1$

Simulation box divided into 8^d subboxes

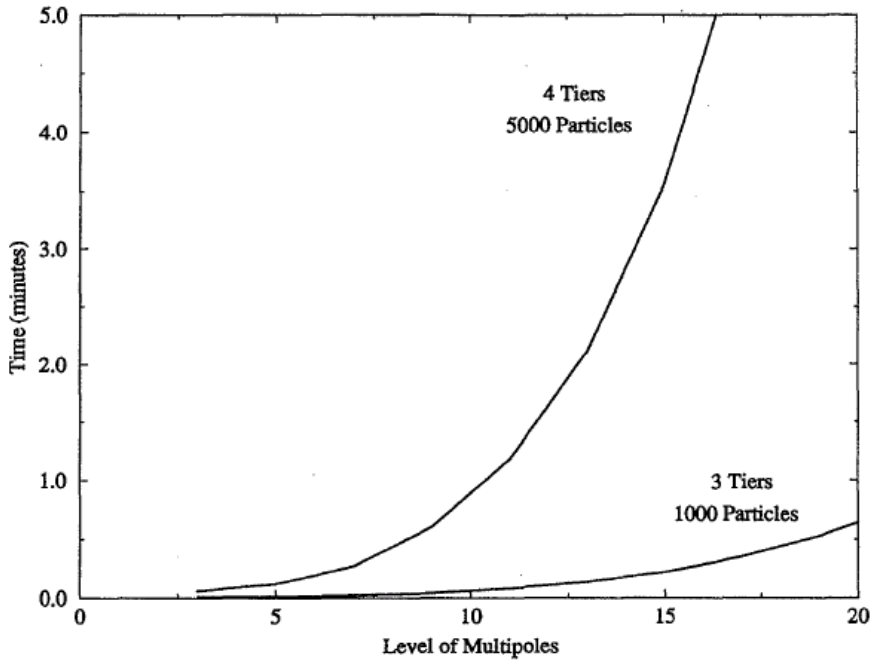
Discussion 1



[J. Chem. Phys. **101** (8), 15 October 1994]

Dependance on particle number of errors in potential per particle calculated by the FMM

Discussion 2

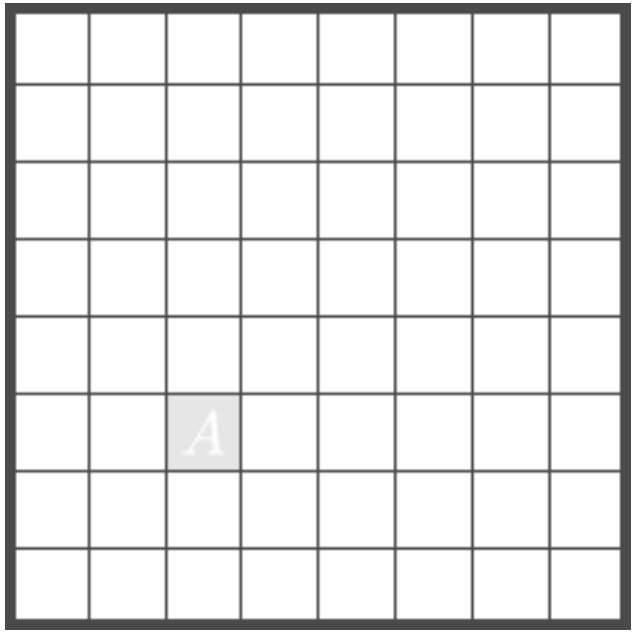


[J. Chem. Phys. **101** (8), 15 October 1994]

Timings showing the way in which the computational complexity of the FMM depends on the numbers of poles, p

FMM Parameter III

Separation criterion *ws*

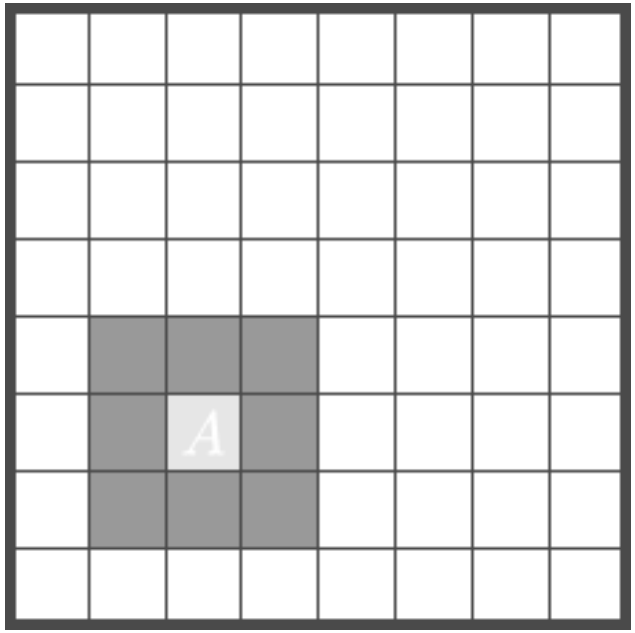


Separation Criterion *ws*

Near field contains $(2 \cdot ws + 1)^3$ boxes

FMM Parameter III

Separation criterion *ws*

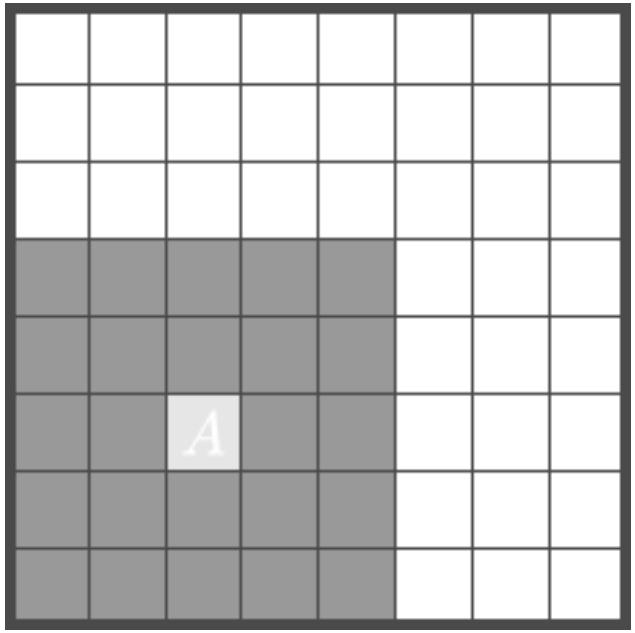


Separation Criterion *ws*

Near field contains $(2 \cdot ws + 1)^3$ boxes

FMM Parameter III

Separation criterion *ws*

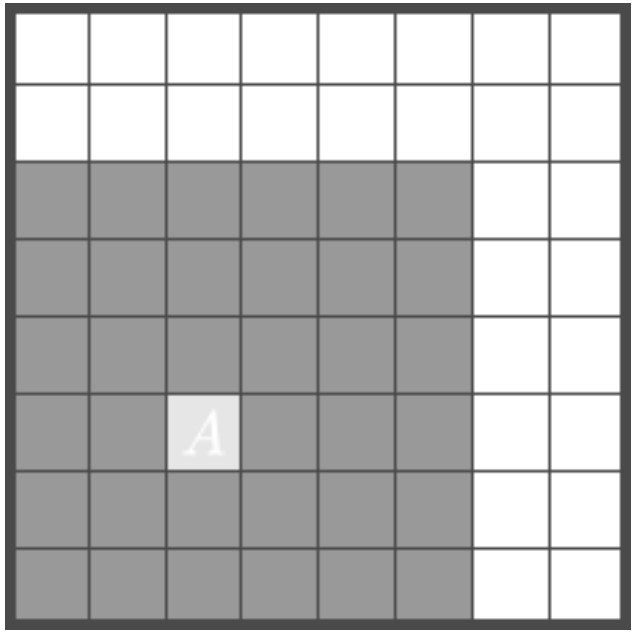


Separation Criterion *ws*

Near field contains $(2 \cdot ws + 1)^3$ boxes

FMM Parameter III

Separation criterion *ws*

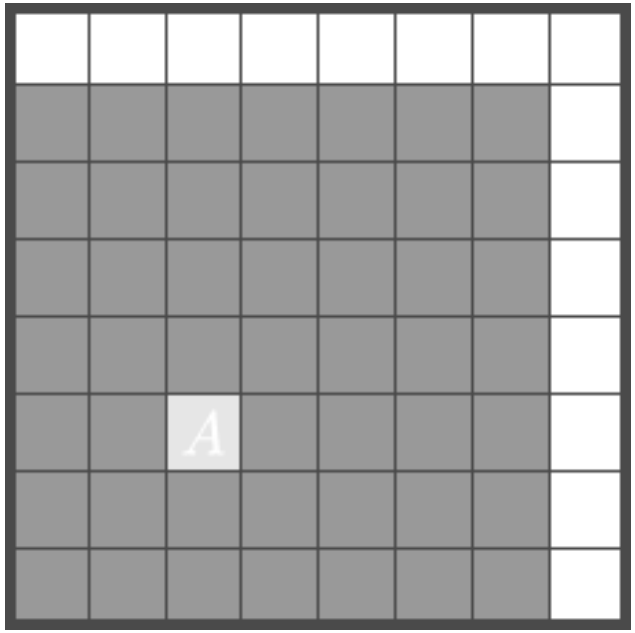


Separation Criterion *ws*

Near field contains $(2 \cdot ws + 1)^3$ boxes

FMM Parameter III

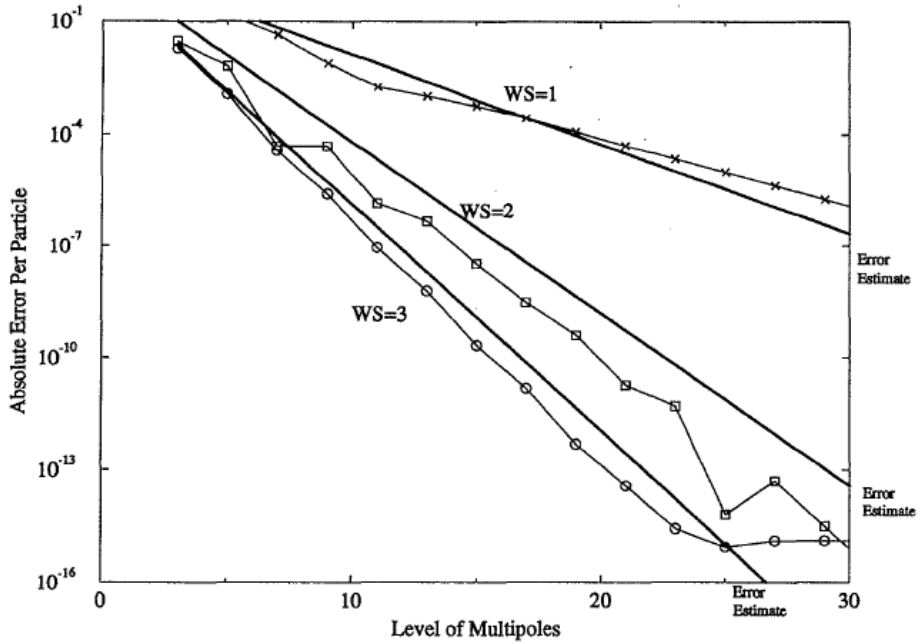
Separation criterion *ws*



Separation Criterion *ws*

Near field contains $(2 \cdot ws + 1)^3$ boxes

Discussion 3



[J. Chem. Phys. **101** (8), 15 October 1994]

The error in the potential per particle as a function of the level, p

JÜLICH
FORSCHUNGSZENTRUM

Implementation and evaluation of parallel integrators for the FMM

Part IV: Conclusion

September 27, 2010 | Valentina Banciu Advisers: Oliver Bücker Ivo Kabadshow

Conclusion

- We have implemented and evaluated different order integrators
- We have implemented a Low-Storage Runge-Kutta scheme which only requires half storage
- We have discussed the FMM operators
- We have shown the FMM passes
- We have analysed the dependancy between FMM parameters

Thank you for your attention

Valentina Banciu
Faculty of Physics, University Of Bucharest
405 Atomistilor Str., Magurele, Romania
vbanciu@live.com