Molecular Dynamics

March 18, 2016

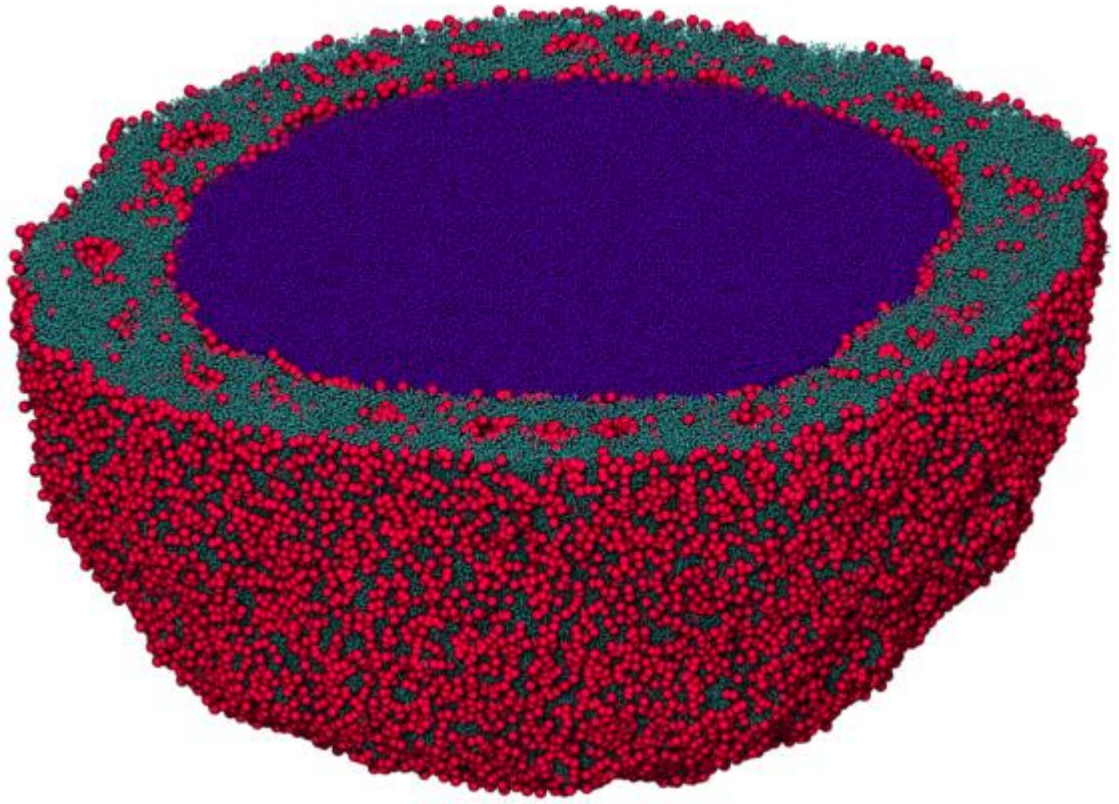
Instructor: Prof. Yeng-Long Chen

TA: 林子翔 (Tzyy-Shyang Lin)

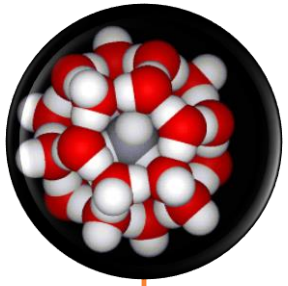
Email: ts.lin.92@gmail.com

What is Molecular Dynamics Simulation?

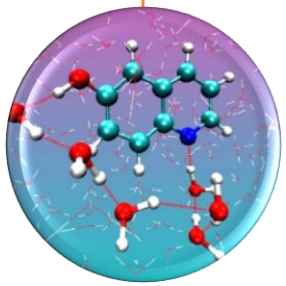
N-Body computer simulation that numerically integrate the EoM of particles to study the trajectory and behavior of atoms and molecules



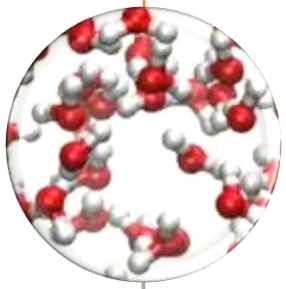
Classification of MD



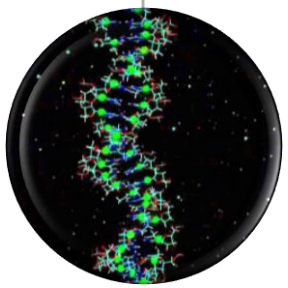
Molecular Dynamics



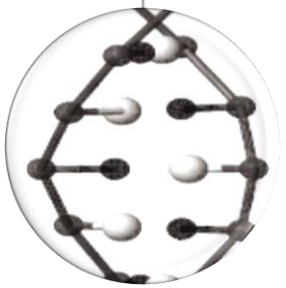
Ab-Initio MD



Classical MD

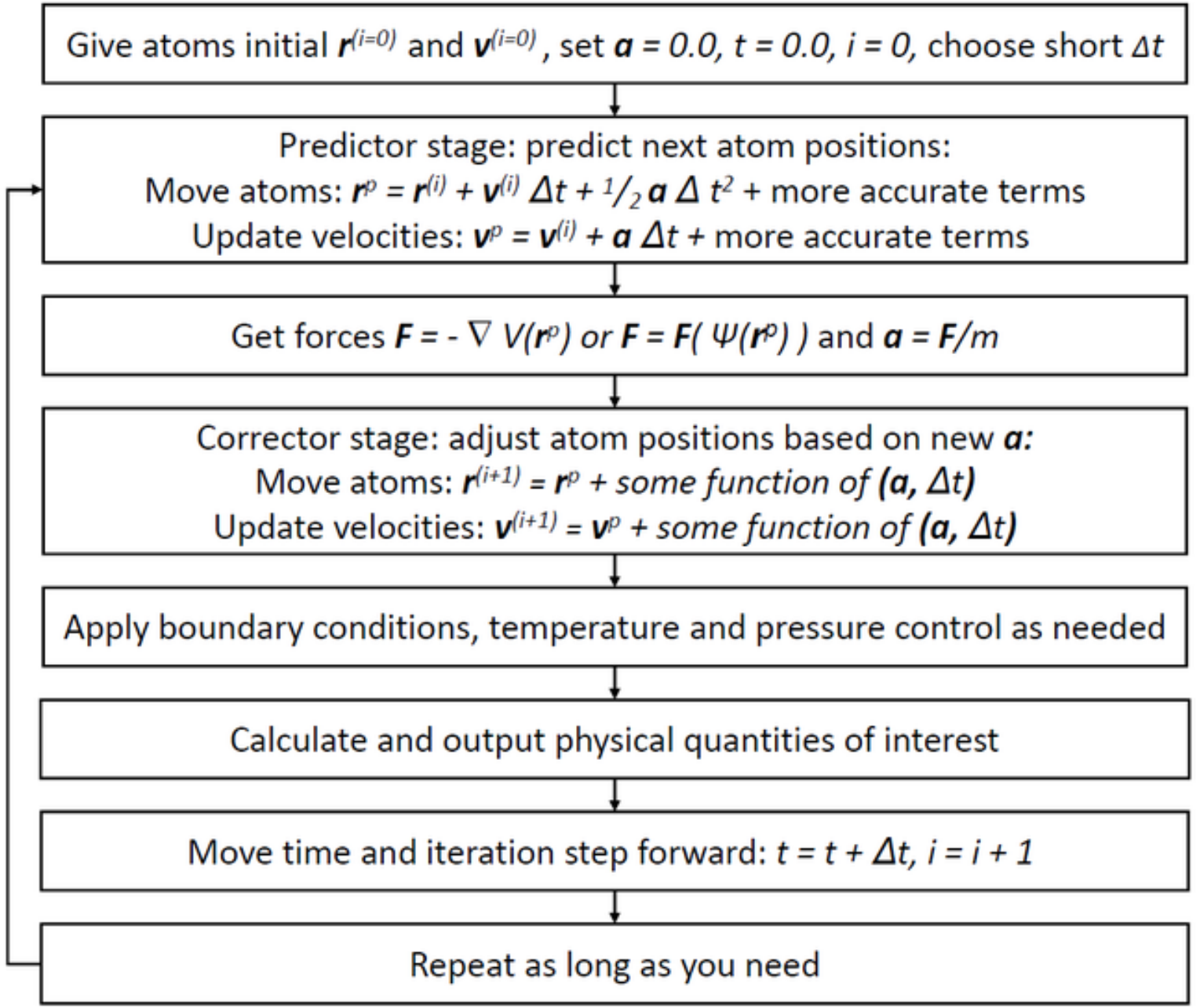


All-Atom (AA)



Coarse Grained (CG)

Recipe of MD


Give atoms initial $\mathbf{r}^{(i=0)}$ and $\mathbf{v}^{(i=0)}$, set $\mathbf{a} = 0.0$, $t = 0.0$, $i = 0$, choose short Δt
Predictor stage: predict next atom positions:
Move atoms: $\mathbf{r}^p = \mathbf{r}^{(i)} + \mathbf{v}^{(i)}\,\Delta t + {}^1/_2\,\mathbf{a}\,\Delta\,t^2$ + more accurate terms
Update velocities: $\mathbf{v}^p = \mathbf{v}^{(i)} + \mathbf{a}\,\Delta t$ + more accurate terms
Get forces $\mathbf{F} = -\nabla V(\mathbf{r}^p)$ or $\mathbf{F} = \mathbf{F}(\Psi(\mathbf{r}^p))$ and $\mathbf{a} = \mathbf{F}/m$
Corrector stage: adjust atom positions based on new $\mathbf{a}$:
Move atoms: $\mathbf{r}^{(i+1)} = \mathbf{r}^p$ + some function of $(\mathbf{a}, \Delta t)$
Update velocities: $\mathbf{v}^{(i+1)} = \mathbf{v}^p$ + some function of $(\mathbf{a}, \Delta t)$
Apply boundary conditions, temperature and pressure control as needed
Calculate and output physical quantities of interest
Move time and iteration step forward: $t = t + \Delta t$, $i = i + 1$
Repeat as long as you need

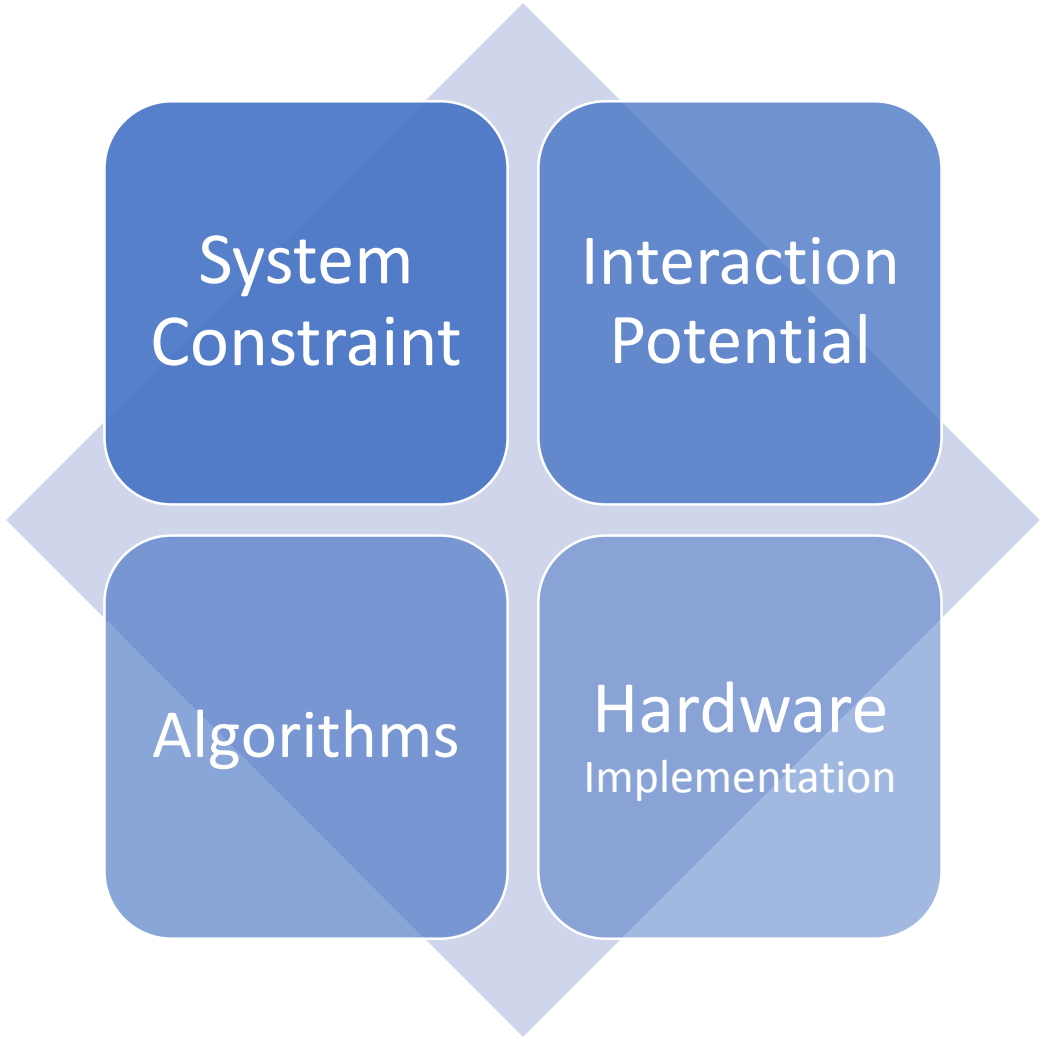
Limitation of MD

- Integration error can never be fully eliminated
- Trajectories should not be perceived as directly comparable to experiments, the statistical behavior should be compared
- Many results may be artificial and due to the limitation of models
- For ergodic systems, statistical behavior can be sampled from a single simulation run, but this is not true for non-ergodic systems

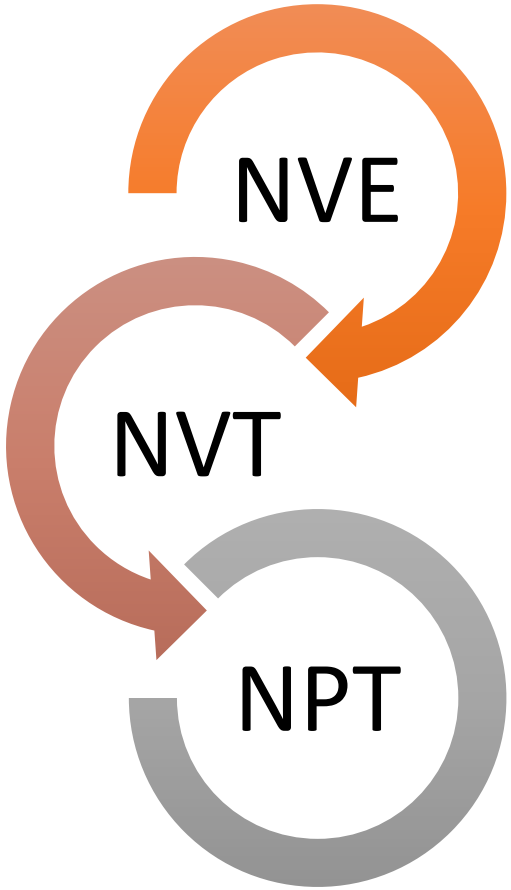
Nondimensionalization

- Molecular dynamics is often carried out in nondimensional form
- Equations of motion and potentials are nondimensionalized prior to simulation
- Different systems of different length scales can be simulated with one simulation
- Often the equations are nondimensionalized with the characteristic length σ, thermal energy $k_B T$ and characteristic time τ

Important Aspects in MD Simulations



System Constraints



Microcanonical Ensemble

Canonical Ensemble

Isothermal–isobaric Ensemble

NVE

- $F(X) = -\nabla U(X) = M\dot{V}(t)$
- $V(t) = \dot{X}(t)$
- For every step, the position and velocity is integrated with a symplectic method

NVT

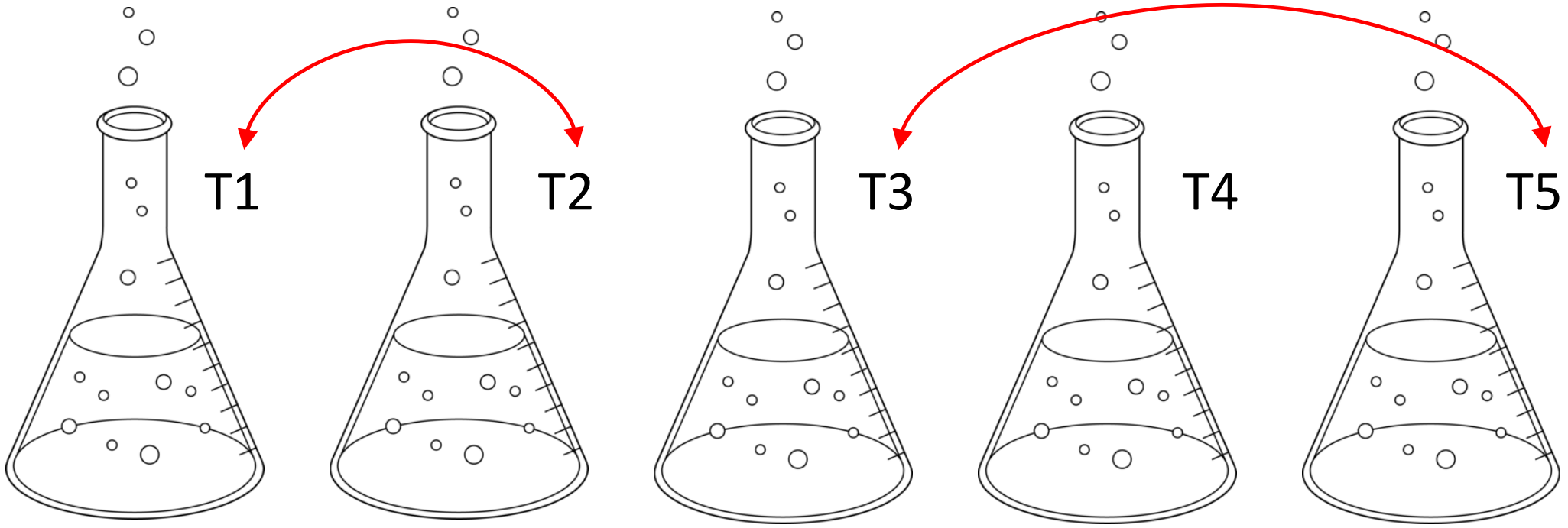
- The total energy of the system is not conserved
- $k_B T = m\langle v^2 \rangle$
- $P(p) = \left(\frac{\beta}{2\pi m}\right)^{3/2} \exp\left(-\frac{\beta p^2}{2m}\right)$
- We need a thermostat to control the temperature

Thermostats

- Rescaling of velocity
- Anderson thermostat
- Berendsen thermostat
- Nose-Hoover thermostat
- Nose-Hoover chains

Other Ensembles

- Replica exchange MD (REMD)
  - For exploring the phase space at low temperature
  - Avoid trapping due to local energy minimum

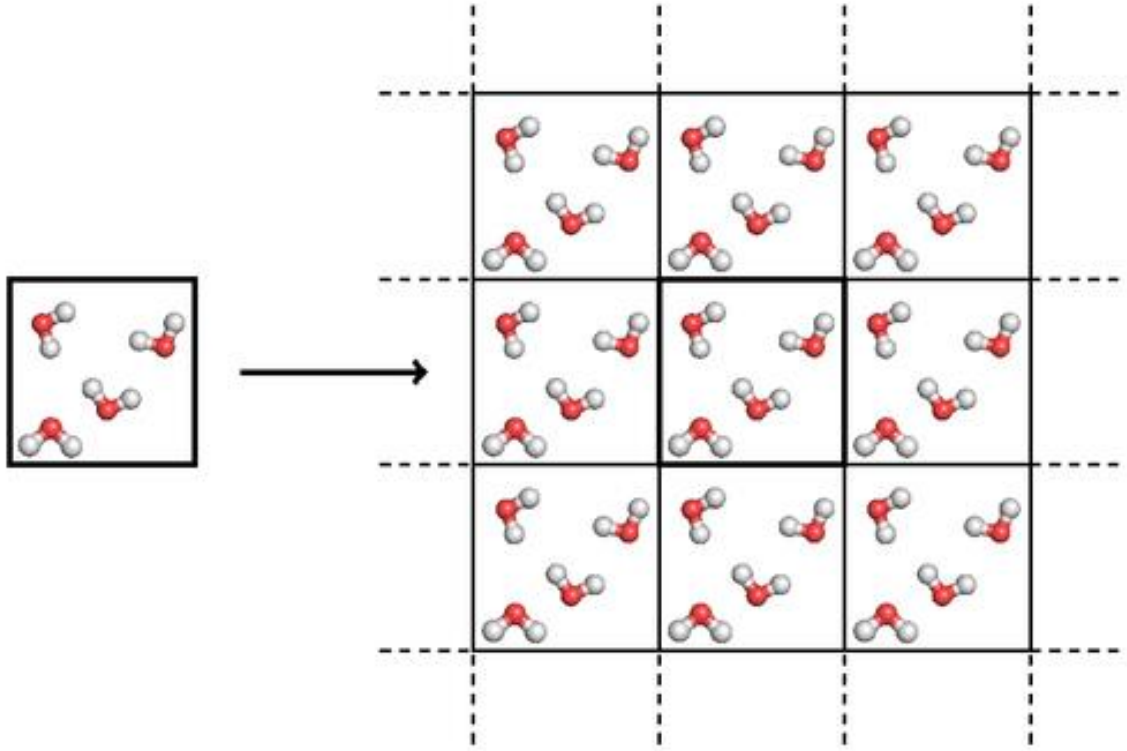


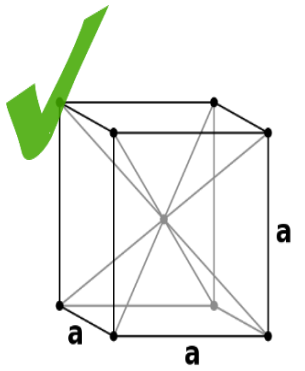
Other Simulation Methods

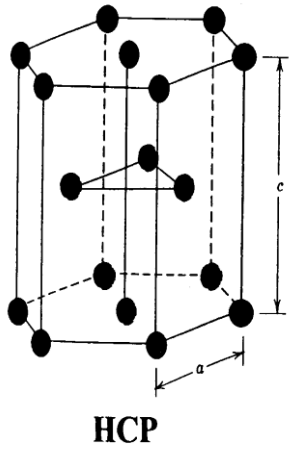
- Langevin dynamics
  - $M\ddot{X} = -\nabla \mathrm{U}(\mathrm{X}) - \zeta \dot{X} + \sqrt{2\zeta k_B T} R(t)$
  - ζ is damping coefficient
  - $R(t)$ is zero-mean delta-correlated stationary Gaussian noise originated from thermal fluctuation
    - $\langle R(t) \rangle = 0$
    - $\langle R(t) R(t') \rangle = \delta(t - t')$
- Brownian dynamics
  - Overdamped case of Langevin dynamics
  - $0 = -\nabla \mathrm{U}(\mathrm{X}) - \zeta \dot{X} + \sqrt{2\zeta k_B T} R(t) \Rightarrow V = -\frac{\nabla \mathrm{U}}{\zeta} + \sqrt{2D} R(t)$
  - D is diffusion coefficient obtained with Einstein relation $\zeta D = k_B T$

Boundary Conditions

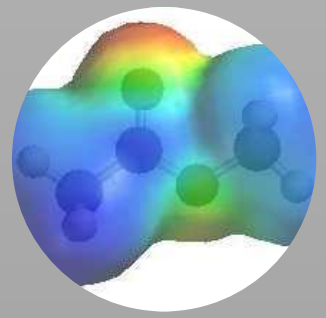
- Period BC
  - Box size should be twice larger than interparticle interaction range
  - Finite size effect: system property with characteristic length comparable to box size cannot be obtained
  - Shape of box might introduce artifacts like crystal formation
- Wall
  - Reflecting
  - Wall potential





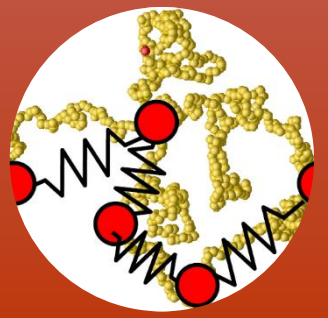


Interaction Potentials



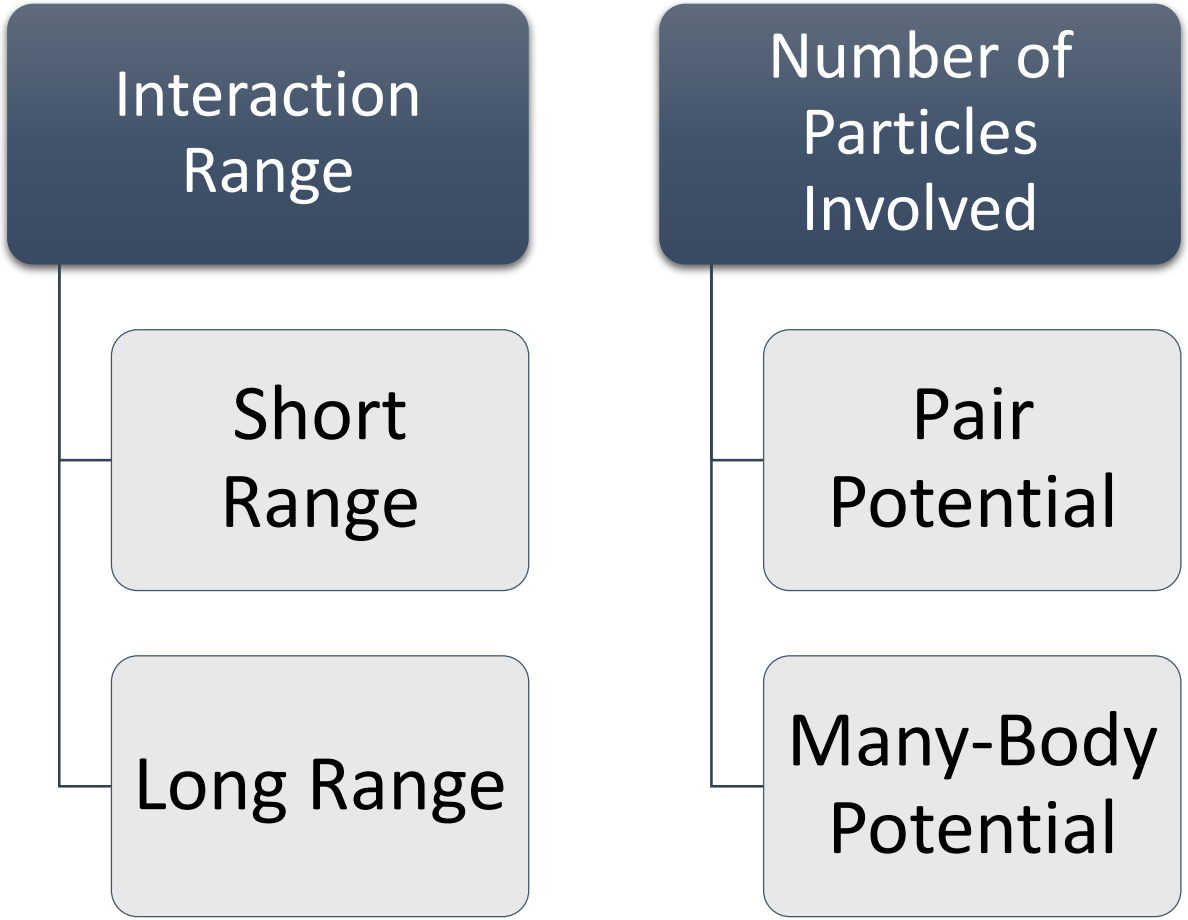
QM Potentials

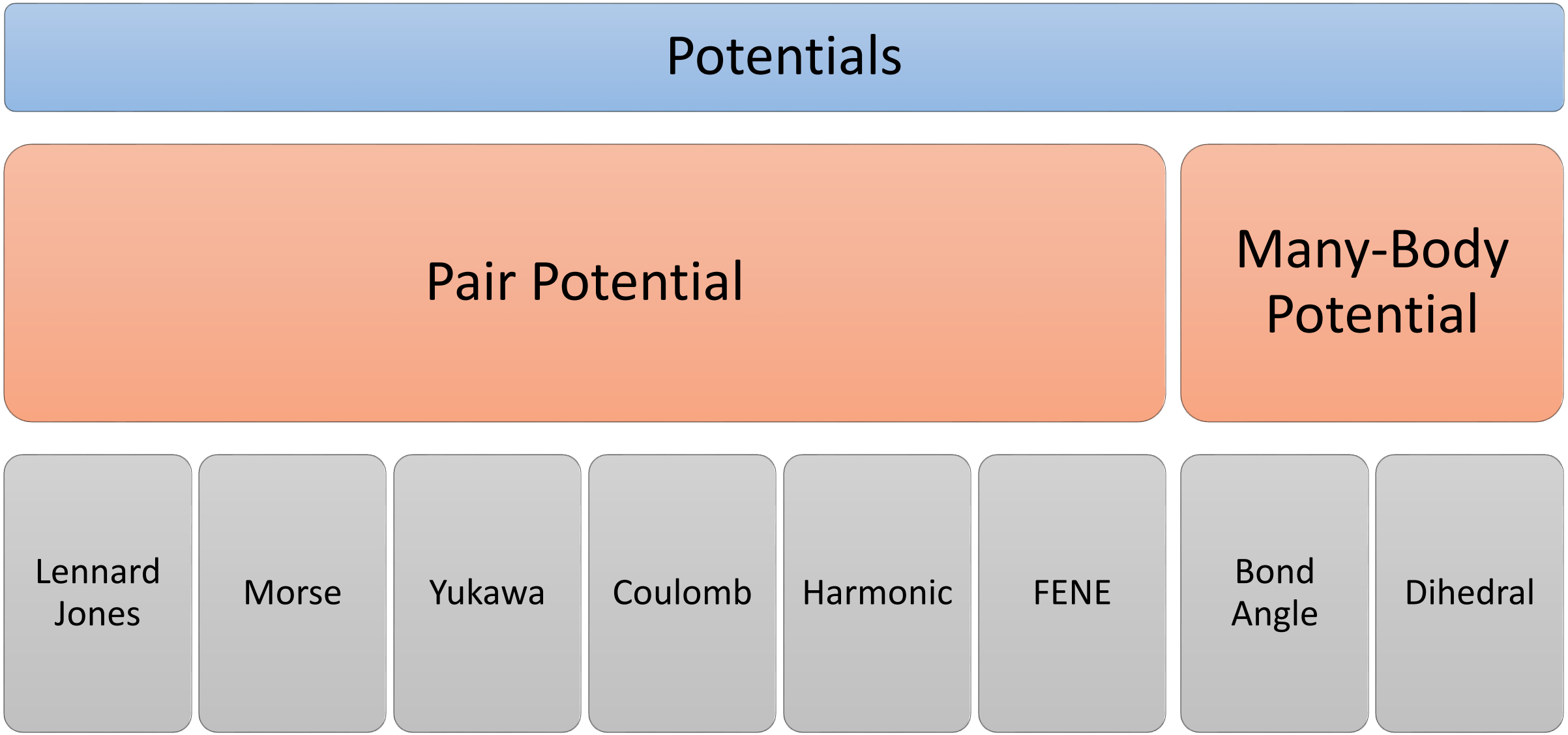
All-Atom Force Fields



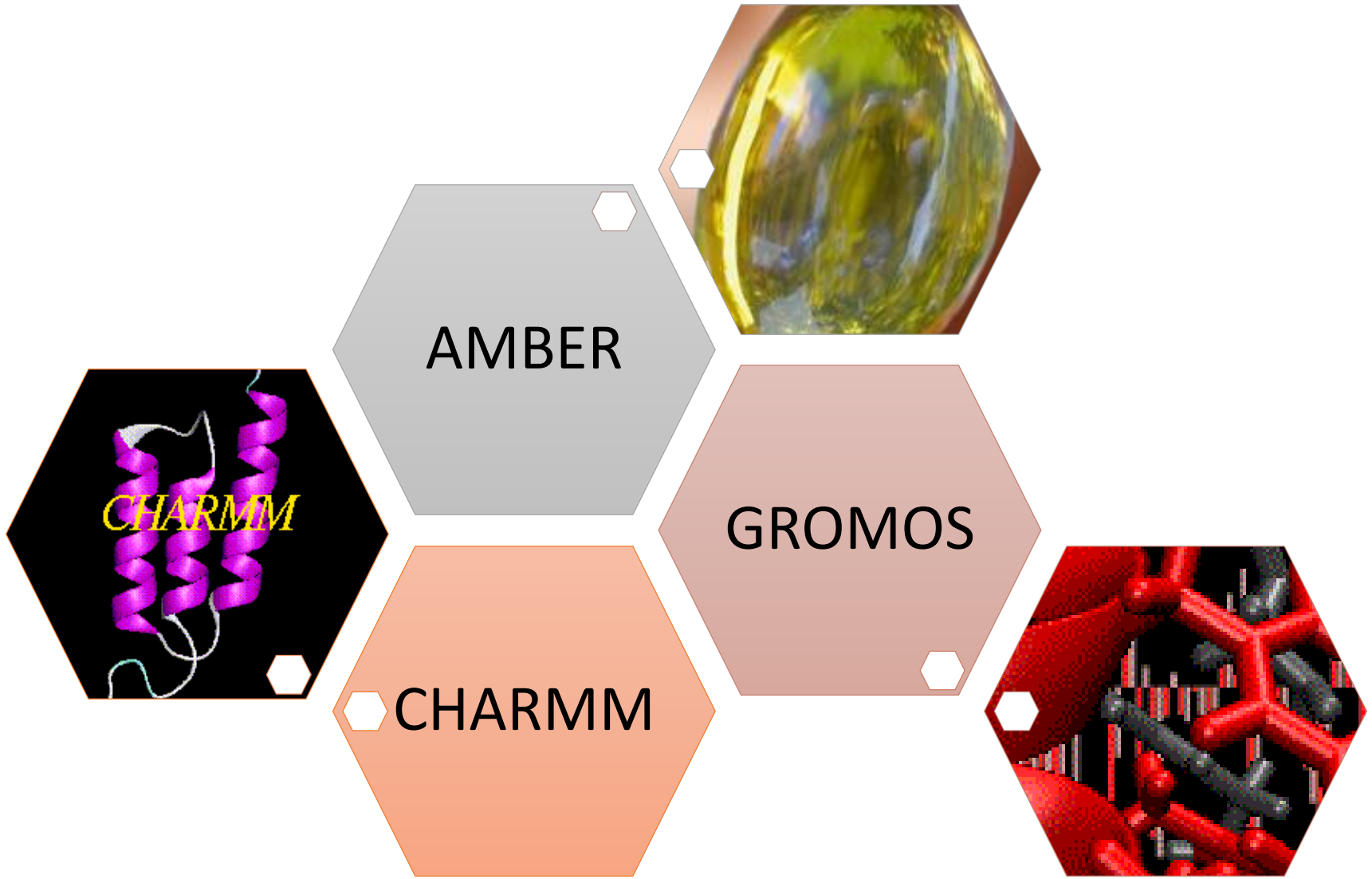
Coarse Grained Potentials

Interparticle Potentials




Potentials
Pair Potential
Many-Body Potential
Lennard Jones
Morse
Yukawa
Coulomb
Harmonic
FENE
Bond Angle
Dihedral

Common Force Fields



Implicit Solvent Methods

- Explicit solvent simulation can be computation intensive
- Some methods to account for electric interaction with solvent
  - Accessible surface area (ASA)
  - Poisson-Boltzmann
  - Generalized Born (GB)
- Viscous effect can be included by using Langevin or Brownian dynamics
- More complex hydrodynamic interaction requires CFD calculation:
  - Lattice Boltzmann
  - Immersed boundary method

Algorithms

- Integrators
- Short-range interaction algorithms
- Long-range interaction algorithms

Integrators

- Different integration schemes have different performance
- Some performance criteria
  - Time reversibility (Phase space area preserving ability)
  - Long-term energy drift
  - Short-term energy drift
  - Accuracy for large steps
  - Required Memory
  - Computation speed

Common Integration Schemes

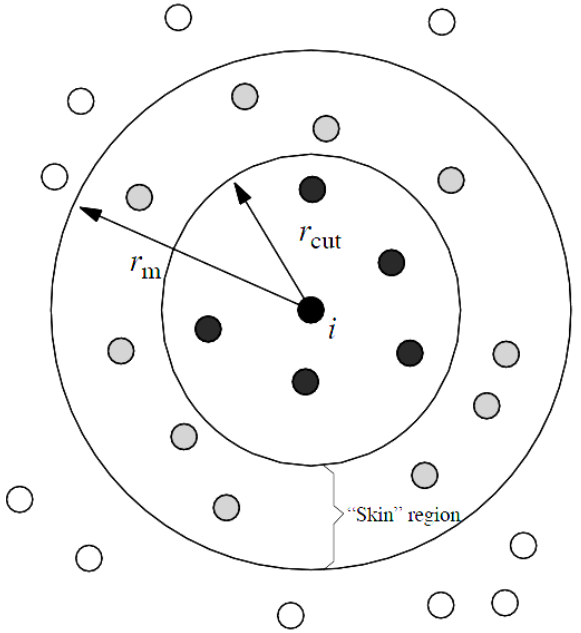
- Euler
- Runge-Kutta
- Verlet-Stoermer
- Velocity Verlet
- Beeman's algorithm

Short-range interaction algorithms

- Computing all pair interactions is a $O(N^2)$ task
- Some algorithms exist that aim to speed up the calculation of short range interactions
- Common algorithms include
  - Verlet list
  - Cell list
  - Combined list

Verlet list

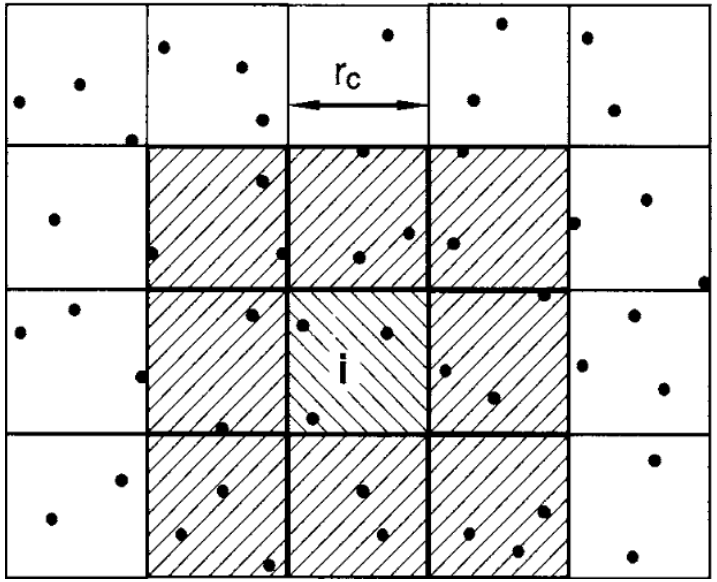
- A list of all particles interacting with a particle is recorded and periodically updated for every particle
- Complexity of tasks
  - Creating neighbor list $\sim O(N^2)$
  - Lists are updated every n_u steps
  - The CPU time to calculate energy using Verlet list is $t_v = cn_vN + c_vN^2/n_u$
- After optimization, the overall complexity is $\sim O(N^{1.5})$



n_v: number of particles in a list
n_u: period of list updates
N: total number of particles
c: coefficient for calculating energy
c_u: coefficient for creating list

Cell List

- The box is divided into cells, and only particles in adjacent cells are considered
- Complexity of tasks
  - Creating cell list $\sim O(N)$
  - Calculation with neighboring cells (8 neighbors in 2D, 26 in 3D)
  - The CPU time to calculate energy using cell list is $t_l = cn_lN + c_lN$



n_l: number of particles in a cell list
N: total number of particles
c: coefficient for calculating energy
c_l: coefficient for creating list

Long-range interaction algorithms

- Potentials that decays with r^{-n} $(n < 3)$ are long range interactions
- A common long range force is the electrostatic interaction
- Some methods to speed up
  - Ewald summation
  - Particle mesh Ewald
  - P3M
  - Shifted force method

Hardware

- MD simulations can be sped up with specialized hardware
- Another option is using parallel programming and utilize GPU acceleration

Any Question?

Programming Exercise #2

- Programming Exercises can be found on the website of this course
- https://softphys.wordpress.com/

- Programming Exercise #2
- https://softphys.wordpress.com/2016/03/11/programming-exercise-2/

In future sessions

- More detailed introduction to reference code
- VMD (visual molecular dynamics)