



MSE6701H Multiscale Materials Modeling and Simulation

Lecture 02

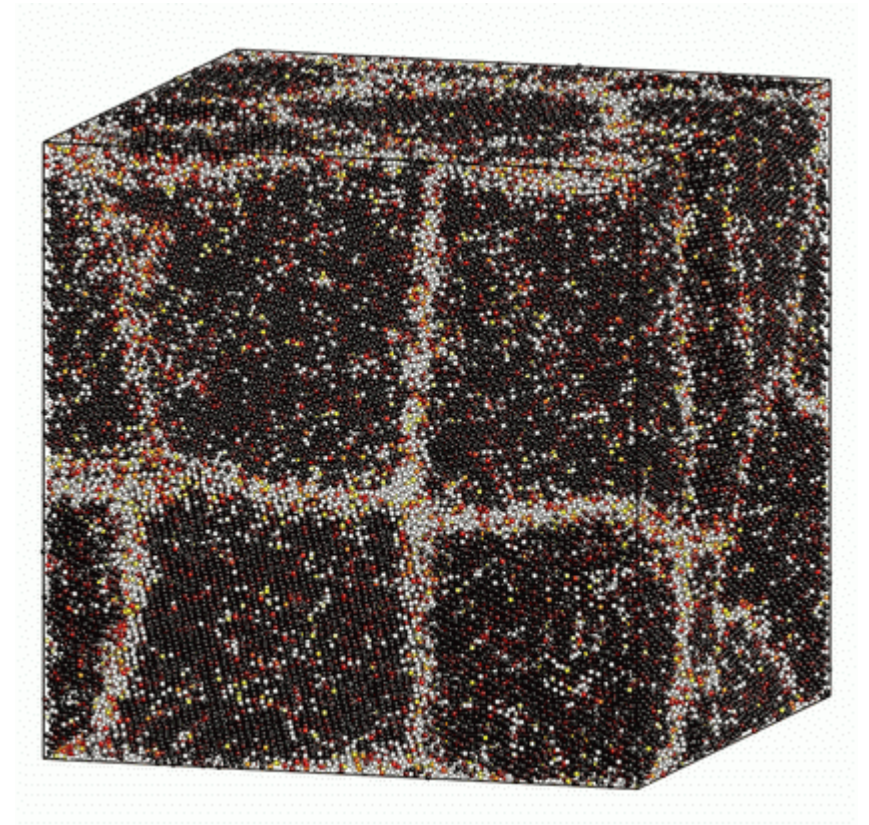
Molecular Dynamics Simulations

Lingt Kong (孔令体)

Shanghai Jiao Tong University

What is MD?

- **Dynamics:** a branch of mechanics that deals with forces and their relation primarily to the **motion** of bodies
- **MD:** Method that simulates the “real” dynamics of a collection of atoms, molecules, particles, or other objects.
- Method widely used in
 - Materials Science
 - Biological Sciences
 - Physics, Chemistry, Mechanics
- Method widely used for
 - Metal, ceramics, polymers, composites
 - Solids, liquids, gasses, and interfaces



MD: History

1953

- Metropolis et al.
- Metropolis Monte Carlo

1956

- Alder and Wainwright
- Molecular dynamics simulation of hard spheres

1964

- A Rahman
- MD for liquid Argon

1974

- Stillinger and Rahman,
- MD of liquid water

1980

- H. C. Andersen
- MD method for NPH, NVT, NPT ensembles

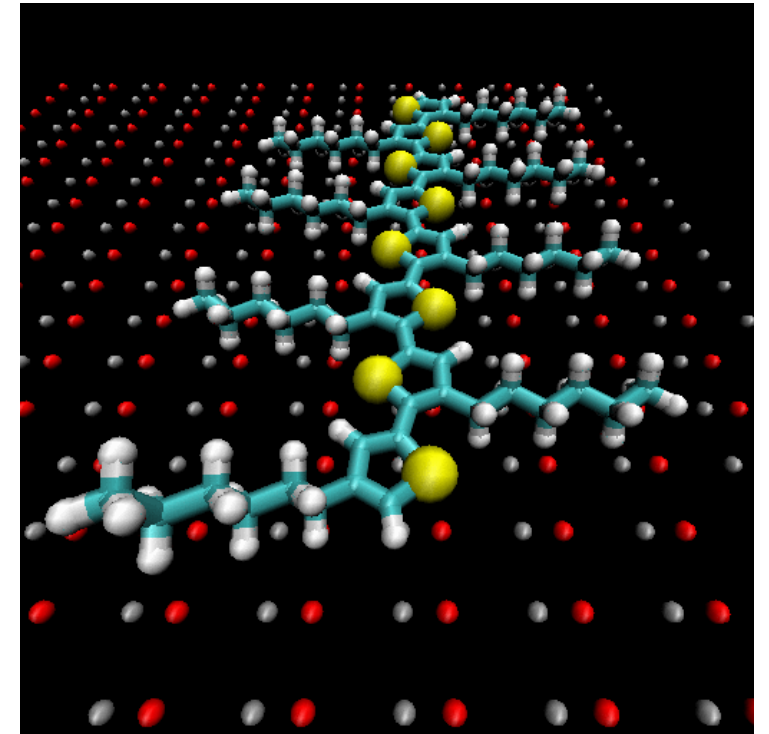
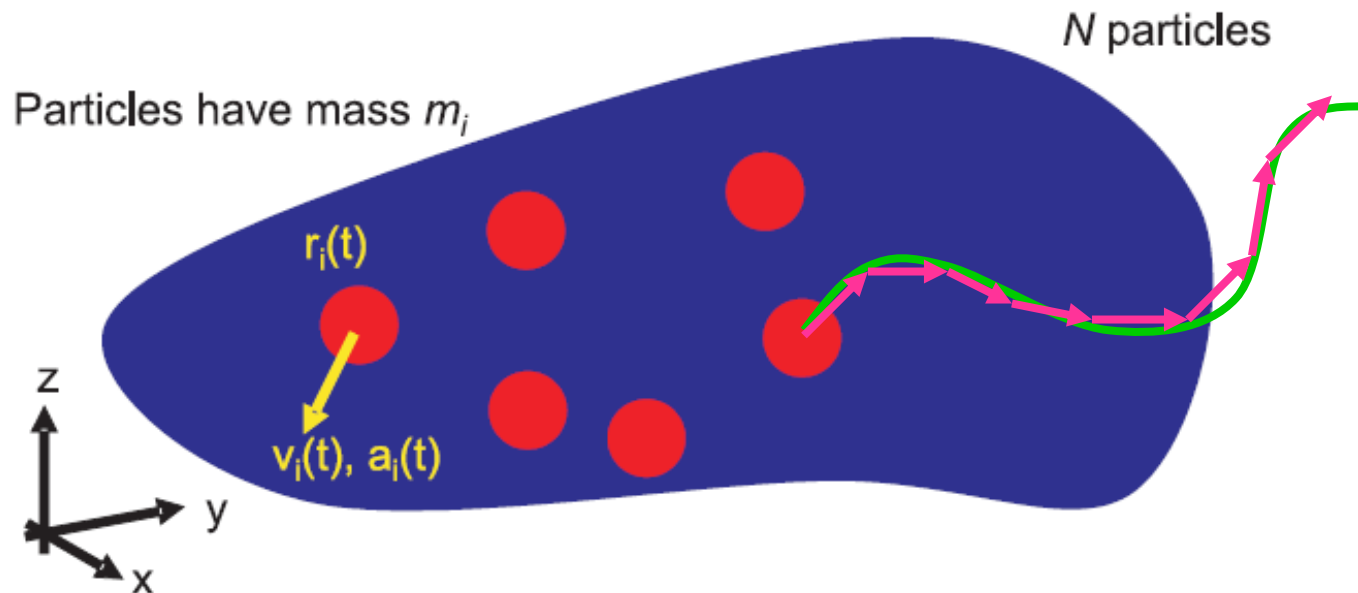
1986

- R. Car and M. Parrinello
- Ab initio MD

MD: Basic Idea

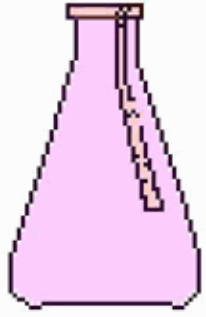
Solution of Newton's equations of motions for the particles

$$\mathbf{F} = m\mathbf{a} = m\ddot{\mathbf{r}}$$



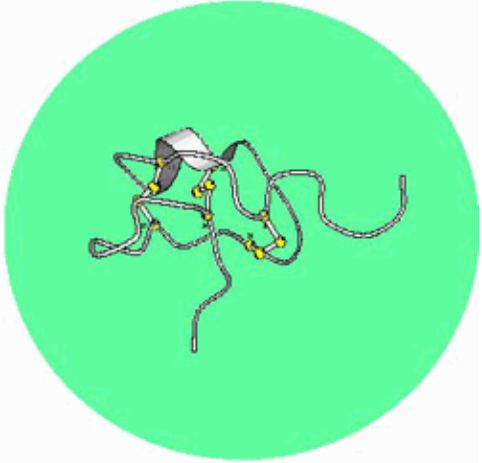
MD: Basic Idea

Experiment



Macroscopic

Molecular Simulation



Microscopic

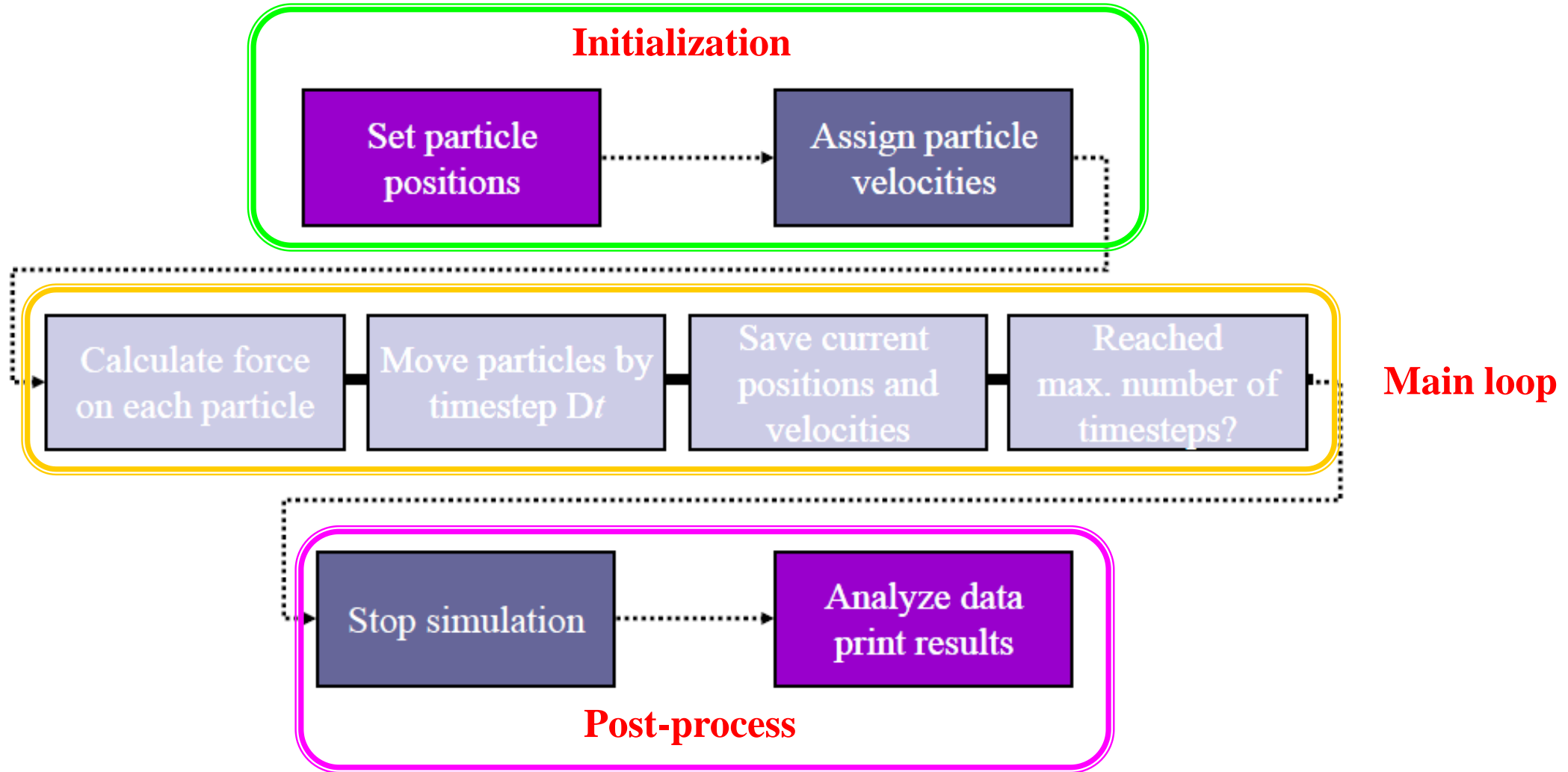
$$\langle A \rangle_{\text{ensemble}} = \int \int d\mathbf{p}^N d\mathbf{r}^N A(\mathbf{p}^N, \mathbf{r}^N) \rho(\mathbf{p}^N, \mathbf{r}^N)$$

$$\rho(\mathbf{p}^N, \mathbf{r}^N) = \frac{1}{Q} \exp \left[-\frac{H(\mathbf{p}^N, \mathbf{r}^N)}{k_B T} \right]$$

$$Q = \int \int d\mathbf{p}^N d\mathbf{r}^N \exp \left[-\frac{H(\mathbf{p}^N, \mathbf{r}^N)}{k_B T} \right]$$

$$\langle A \rangle_{\text{time}} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_{t=0}^{\tau} A(\mathbf{p}^N(t), \mathbf{r}^N(t)) dt = \frac{1}{M} \sum_{t=1}^M A(\mathbf{p}^N, \mathbf{r}^N)$$

MD: General procedure



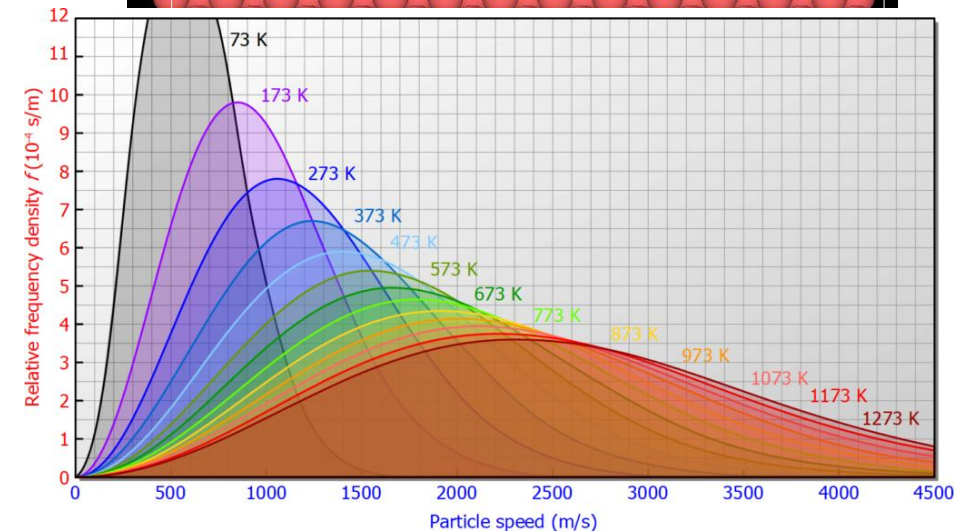
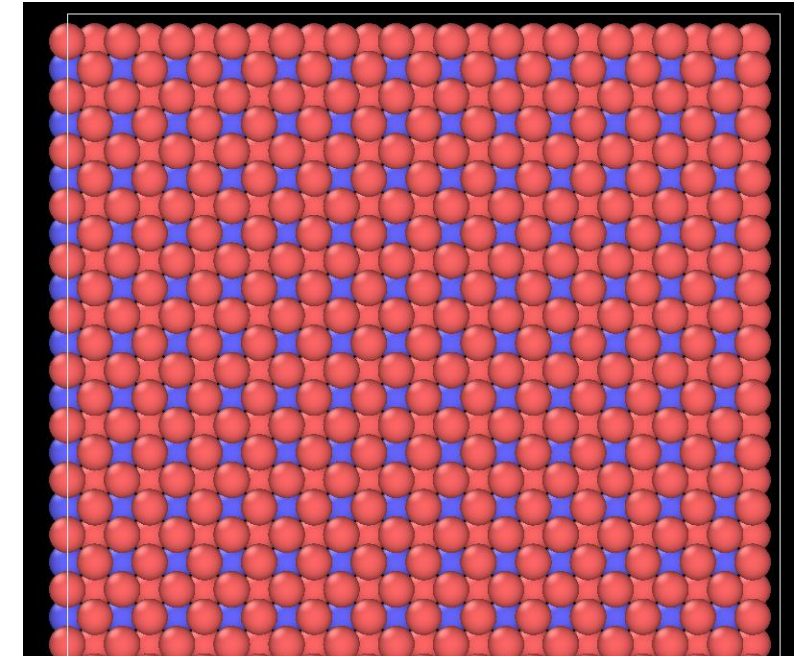
Initialization

- Initial atomic configuration
- Initial velocity (temperature)
- Boundary condition
- Controlling parameters

Maxwell-Boltzmann Distribution

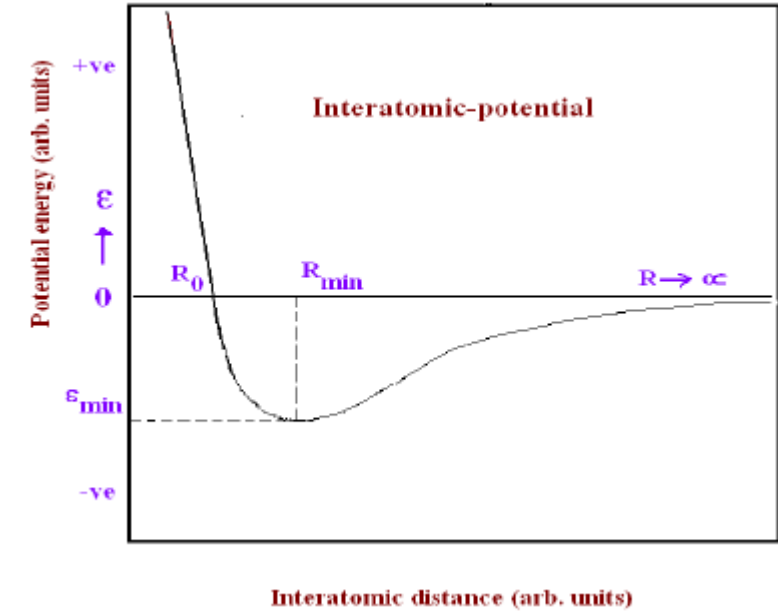
$$p(v) = \left(\frac{m}{2\pi k_B T} \right)^{1/2} \exp \left[-\frac{1}{2} \frac{mv^2}{k_B T} \right]$$

$$\sum_{i=1}^N m_i \mathbf{v}_i = 0 \quad T = \frac{1}{3Nk_B} \sum_{i=1}^N \frac{|\mathbf{p}_i|^2}{2m_i}$$

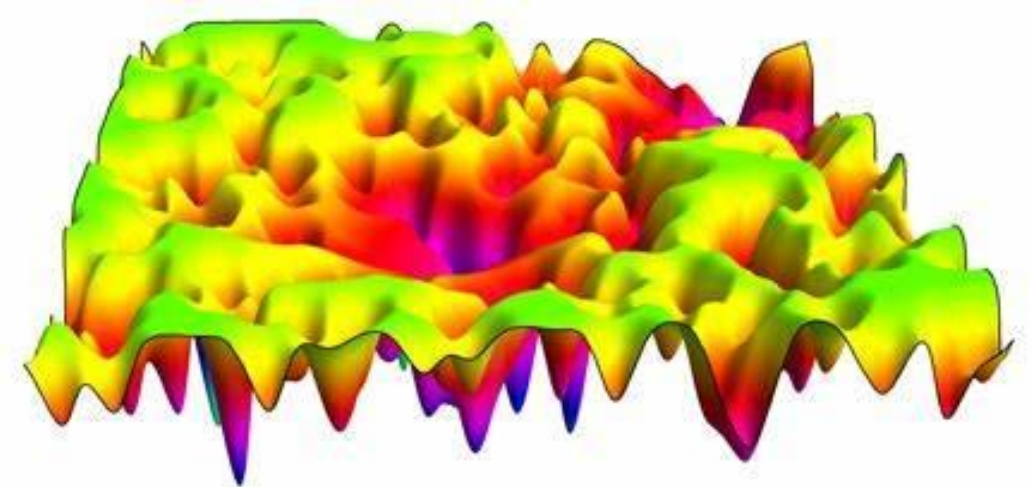


Force calculation

$$U(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$$



$$\begin{aligned}\vec{f}_i &= -\frac{\partial U}{\partial \vec{r}_i} \\ &= -\sum_{j \neq i} \frac{\partial U}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial \vec{r}_i}\end{aligned}$$



Integration algorithm: Verlet algorithm.

$$x = x_0 + v_0 t + \frac{1}{2} a t^2 + \frac{1}{6} b t^3 + \dots$$

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 + \frac{1}{6}b(t)\Delta t^3 + \mathcal{O}(\Delta t^4)$$

$$x(t - \Delta t) = x(t) - v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 - \frac{1}{6}b(t)\Delta t^3 + \mathcal{O}(\Delta t^4)$$

$$x(t + \Delta t) = 2x(t) - x(t - \Delta t) + a(t)\Delta t^2 + \mathcal{O}(\Delta t^4)$$

Integration algorithm: Verlet algorithm.

$$v(t) = \frac{x(t + \Delta t) - x(t - \Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2)$$

Disadvantages

- Error in velocity is $\mathcal{O}(\Delta t^2)$
- Need $x(t+\Delta t)$ to get $v(t)$
- Numerical imprecision in adding small and big numbers

$$v(t + \Delta t) = \frac{x(t + \Delta t) - x(t)}{\Delta t} + \mathcal{O}(\Delta t)$$

$$\overset{\mathcal{O}(\Delta t)}{x(t + \Delta t) - x(t)} = \overset{\mathcal{O}(\Delta t)}{x(t) - x(t - \Delta t)} + \overset{\mathcal{O}(\Delta t^2)}{a(t)\Delta t^2} + \mathcal{O}(\Delta t^4)$$

Integration algorithm: velocity Verlet

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2$$

$$a(t + \Delta t) = f(x(t + \Delta t))$$

$$v(t + \Delta t) = v(t) + \frac{1}{2}(a(t) + a(t + \Delta t))\Delta t$$

$$v(t + \frac{1}{2}\Delta t) = v(t) + \frac{1}{2}a(t)\Delta t$$

$$x(t + \Delta t) = x(t) + v(t + \frac{1}{2}\Delta t)\Delta t$$

$$a(t + \Delta t) = f(x(t + \Delta t))$$

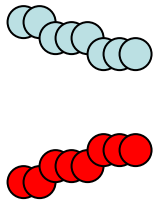
$$v(t + \Delta t) = v(t + \frac{1}{2}\Delta t) + \frac{1}{2}a(t + \Delta t)\Delta t$$

Other integration algorithms

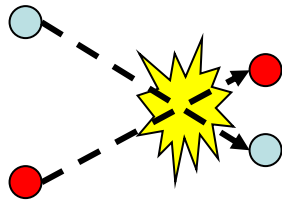
- Leap frog
- Beeman algorithm
- Gear algorithm
- Tuckerman, Berne and Martyna

Time step

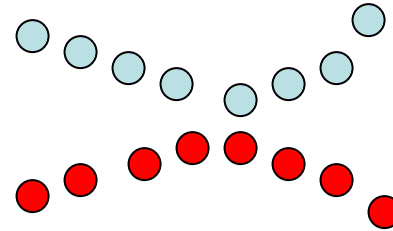
$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2$$



Too small



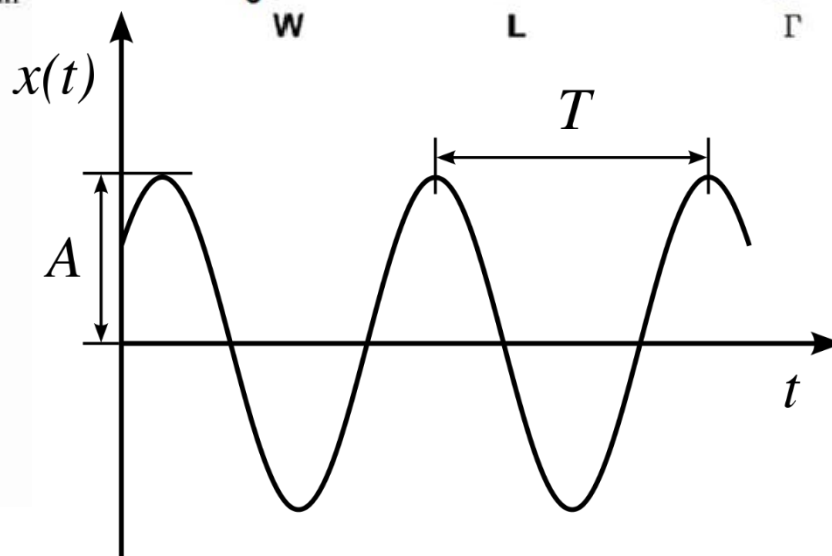
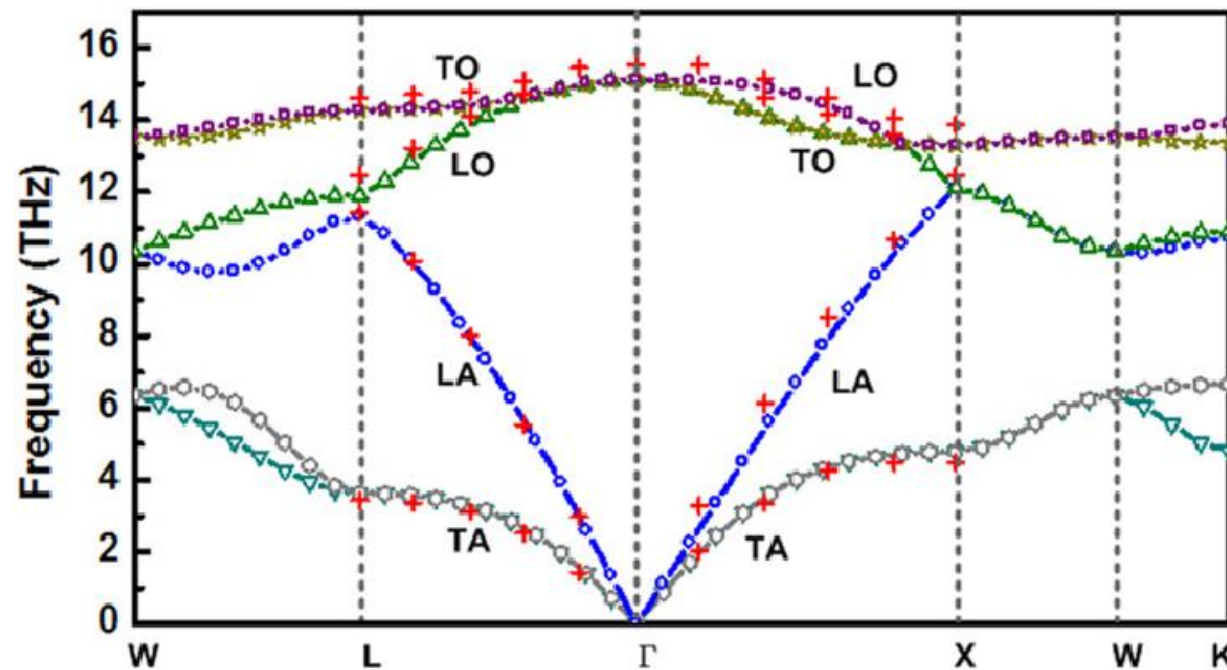
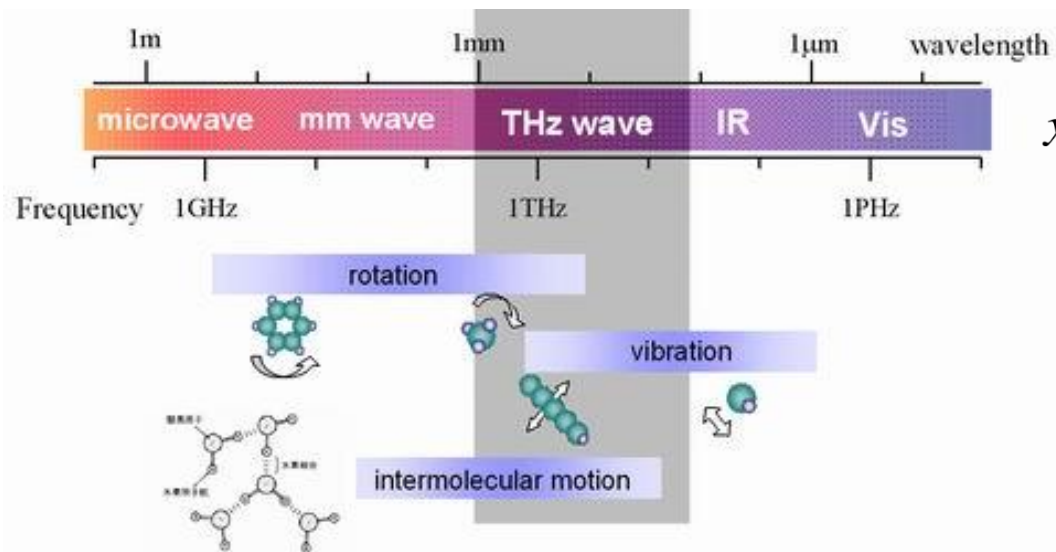
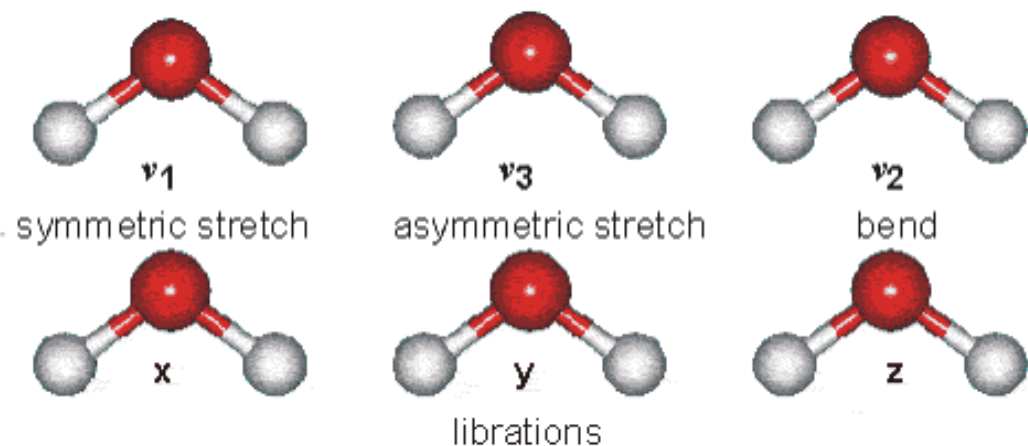
Too large



Just fine

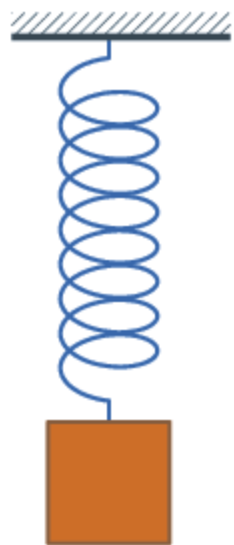
- ◆ As large as needed to generate trajectory sufficient long for problem
- ◆ Can't be too large, otherwise inaccurate or unstable
- ◆ Must be smaller than the smallest time modeled

Time step



$$\Delta t \sim \frac{T_{\min}}{20 \sim 50}$$

First MD: harmonic oscillator



```
def MD_Harmonic():
    mass = 1.
    k = 1.
    x = 0.
    f = -k * x
    v = 0.5
    a = f / mass

    period = 2.*np.pi*np.sqrt(mass/k)
    dt = 0.05
    half_dt = dt * 0.5
    nsteps = int(2.*period/dt)
    ioutput = 5

    thermo = list()
    Ep = 0.5*k*x*x
    Ek = 0.5*mass*v*v
    Et = Ep + Ek
    thermo.append([0, 0*dt, x, v, Ep, Ek, Et])

    for istep in range(1,nsteps+1):
        v = v + a * half_dt
        x = x + v * dt
        f = -k * x
        a = f / mass
        v = v + a * half_dt

        if istep%ioutput == 0:
            Ep = 0.5*k*x*x
            Ek = 0.5*mass*v*v
            Et = Ep + Ek

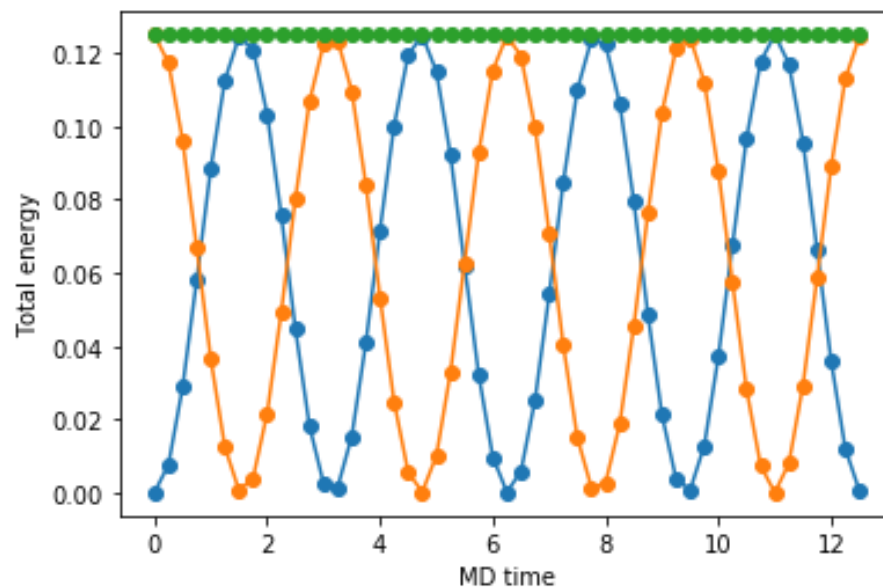
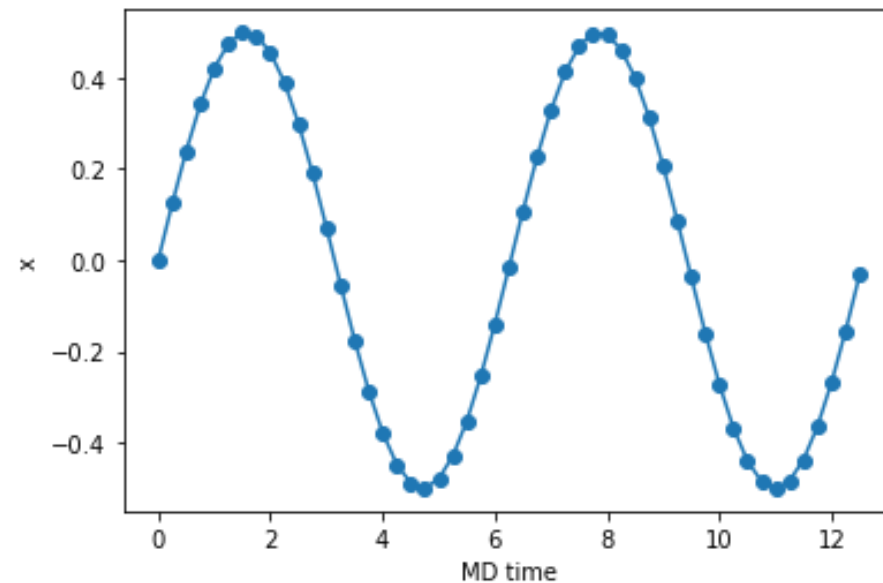
            thermo.append([istep, istep*dt, x, v, Ep, Ek, Et])
    return thermo
```

Initialization

Setup

Verlocity-verlet

Property



Boundary Conditions

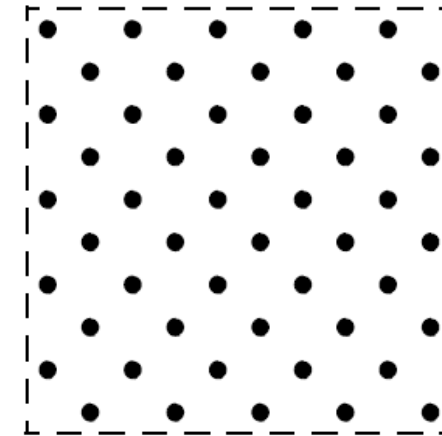
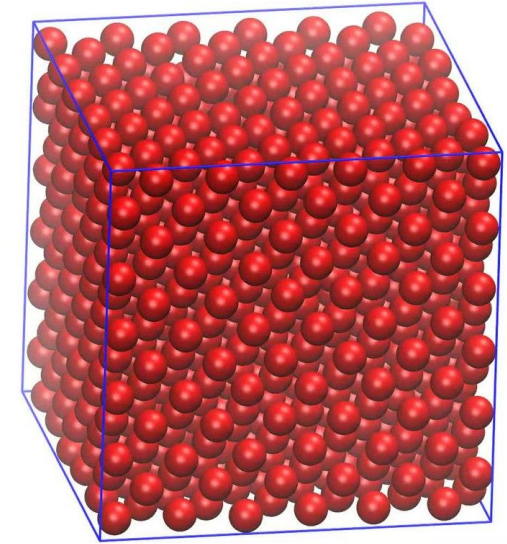
Simulation cell: **domain within which the simulation is carried out.**

- Frequently orthogonal
- Non-orthogonal also possible

Problem: What to do with the atoms at the borders?

Option 1: free boundary

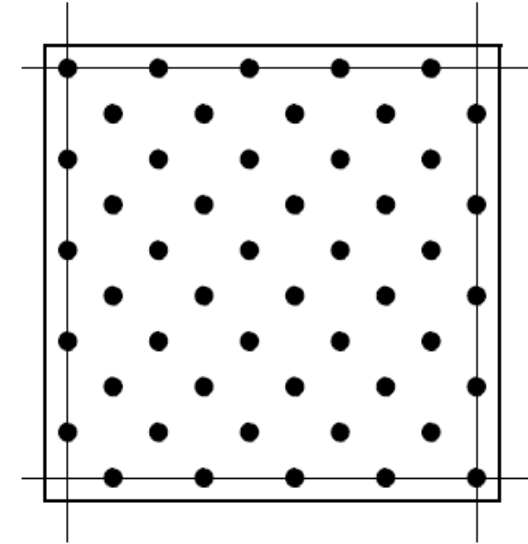
- ✓ Suitable for single molecule, cluster, nanoparticle
- Does not work for bulk materials: dangling bonds



Boundary Conditions

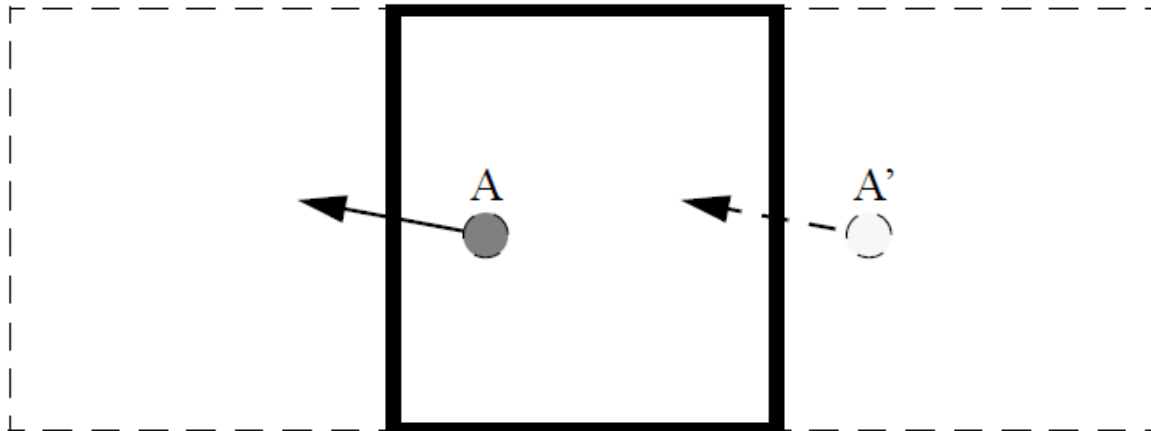
Option 2: fix the boundary atoms

- ✓ Completely unphysical, generally avoided
- ✓ Sometimes needed

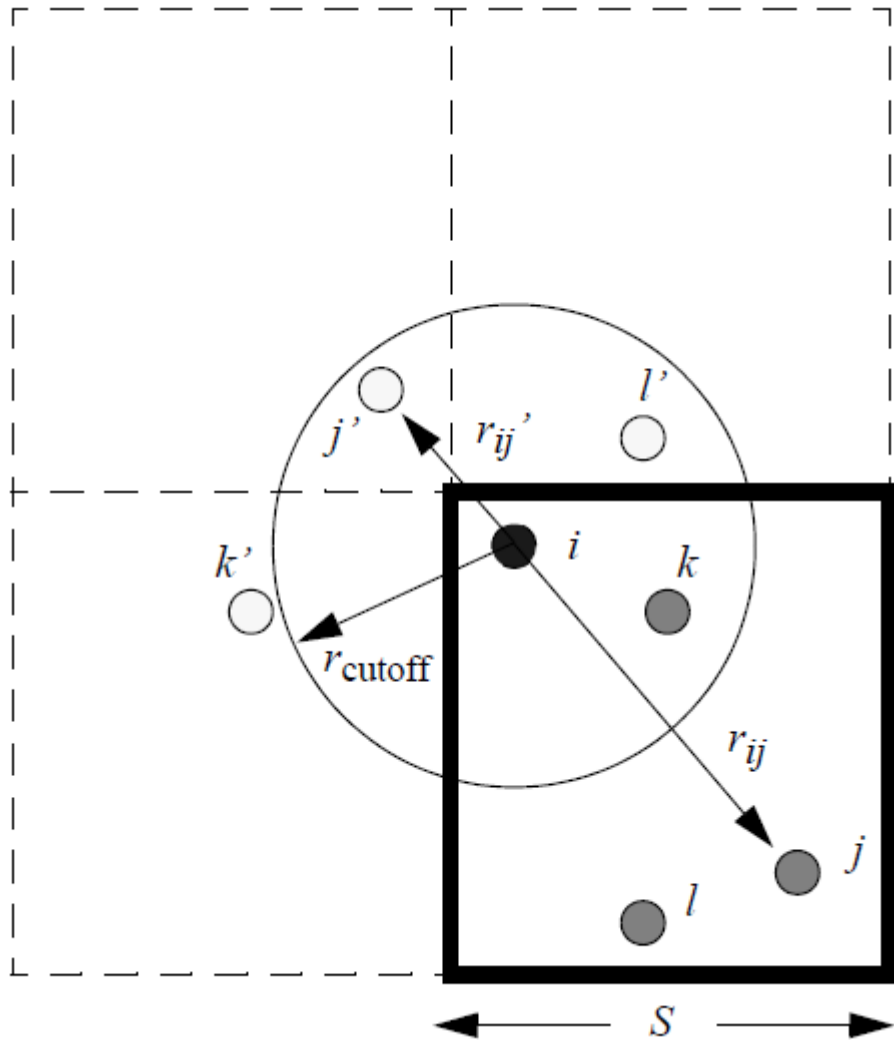


Option 3: Periodic boundary condition

- ✓ Atoms passing the boundary comes back on the other side
- ✓ PBC must be considered when evaluating distances



Periodic boundary condition

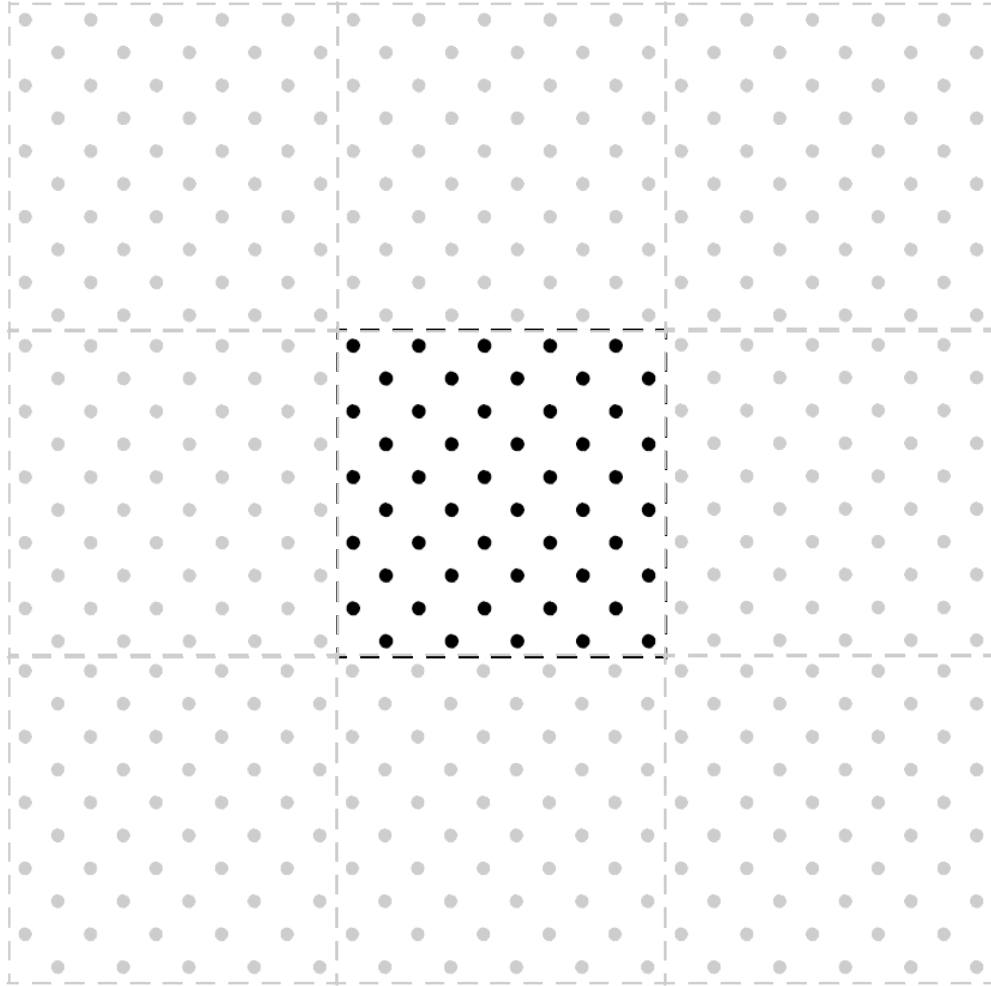


$$\vec{r}_{ij} = x_{ij}\vec{x} + y_{ij}\vec{y} + z_{ij}\vec{z}$$

$$x_{ij} = x_j - x_i$$

```
if pbc_x:
    while xij > 0.5*xbox:
        xij = xij - xbox
    while xij <= -0.5*xbox:
        xij = xij + xbox
```


Periodic boundary condition



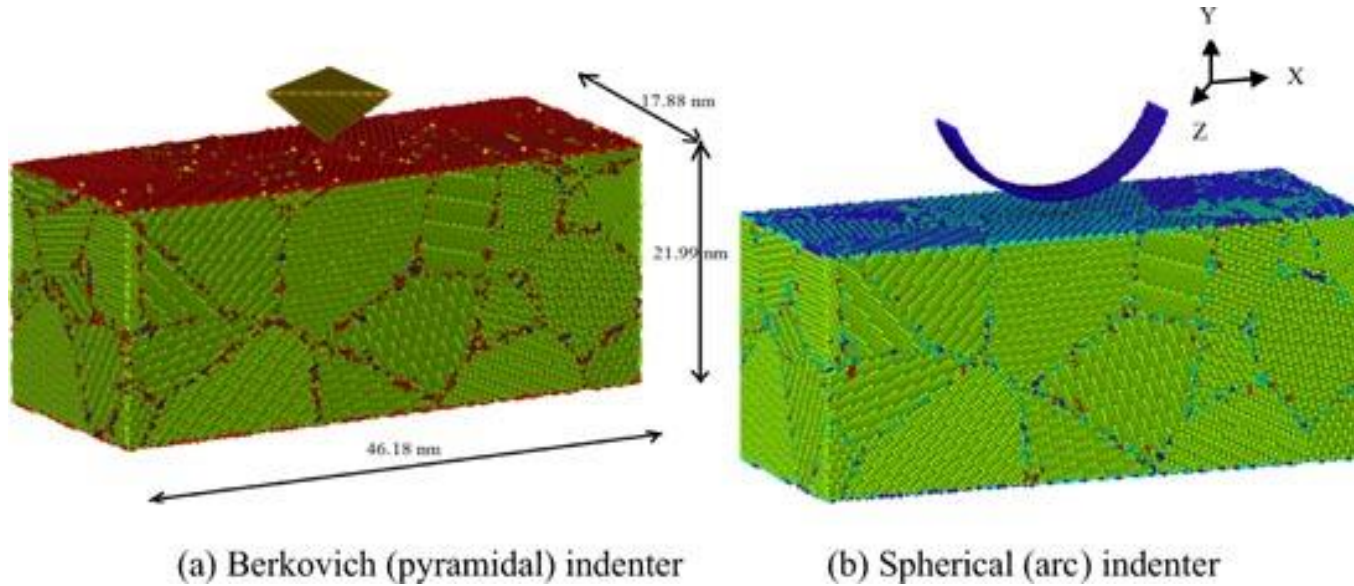
Pro:

- ✓ Eliminate free surface
- ✓ Model infinite system

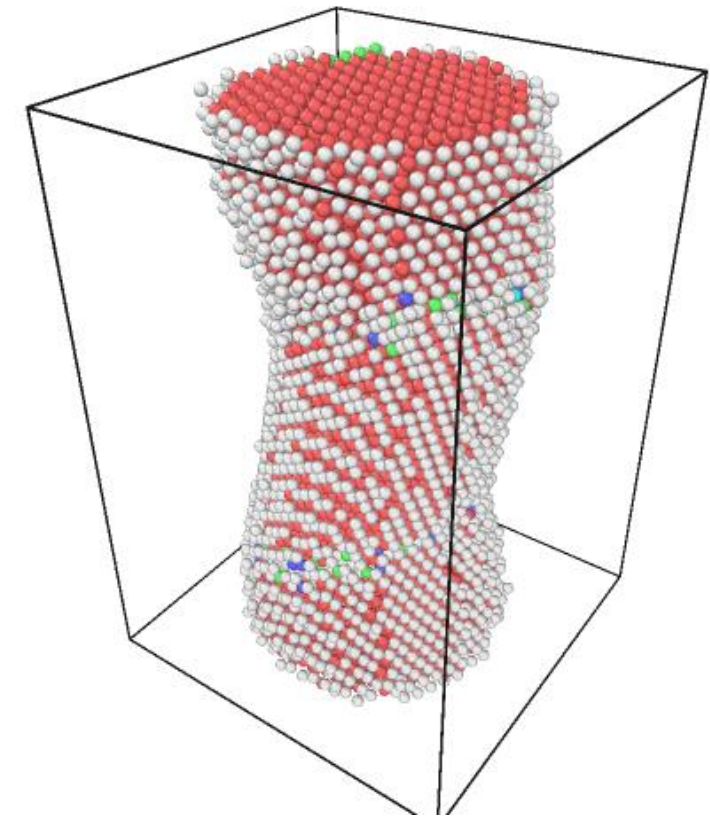
Con:

- ❑ Artificial interaction across boundary
- ❑ Upper limit for phonon wavelength
- ❑ Size scaling test generally required

Mixed boundary conditions

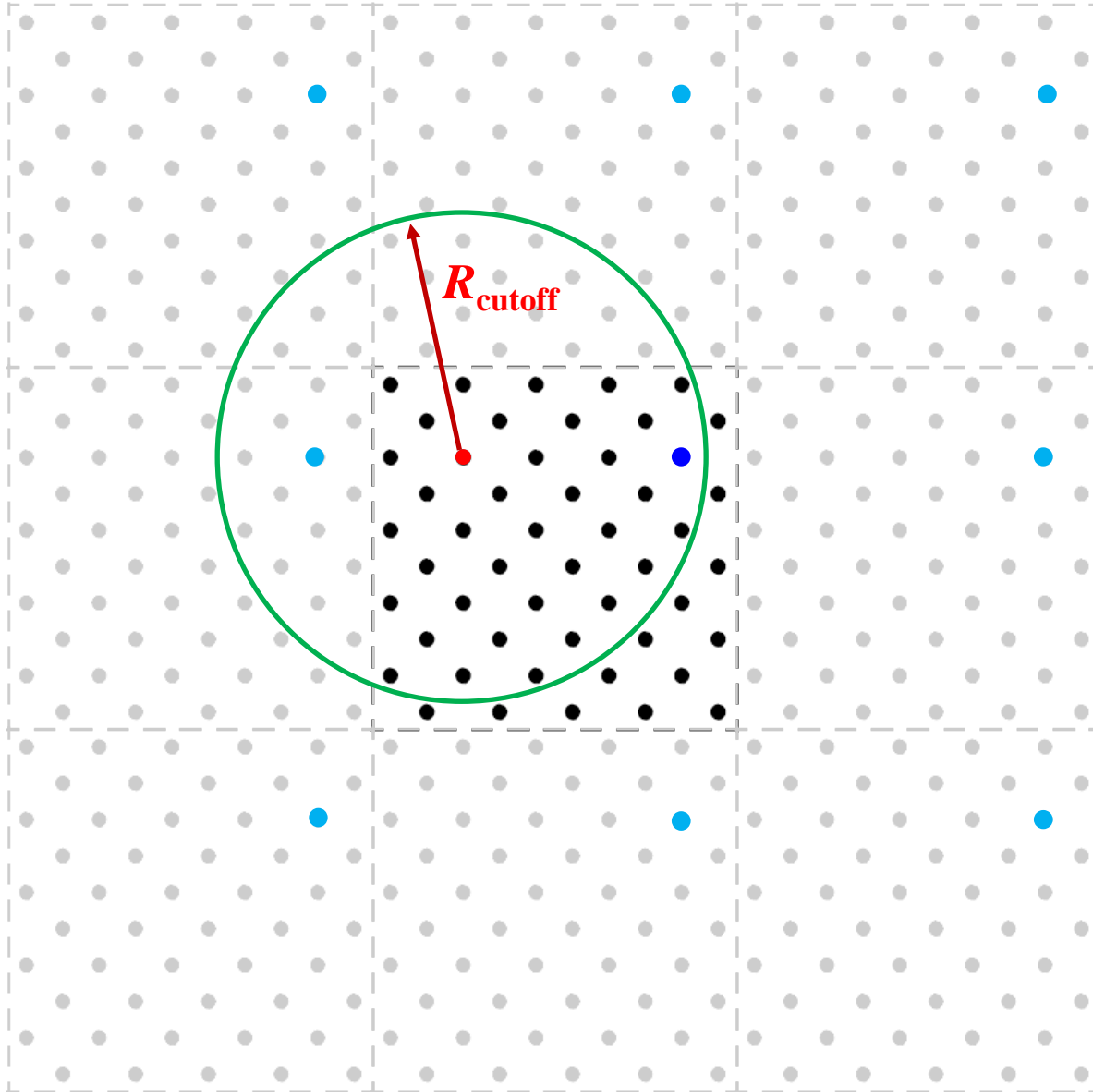


Side	BC
Y+	Free
Y-	Fixed
X	Periodic
Z	Periodic



Side	BC
Axial	Periodic
Transverse	Free

Minimum Image Convention

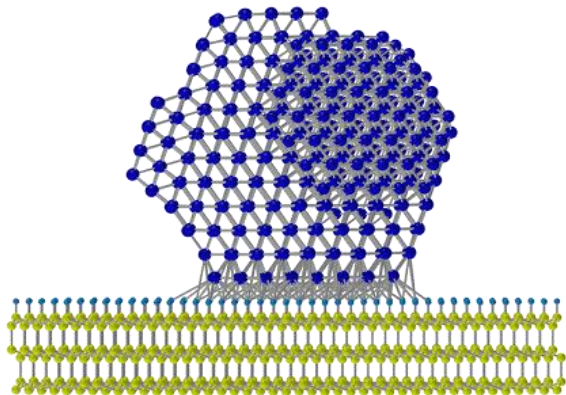


$$L > 2 * R_{\text{cutoff}}$$

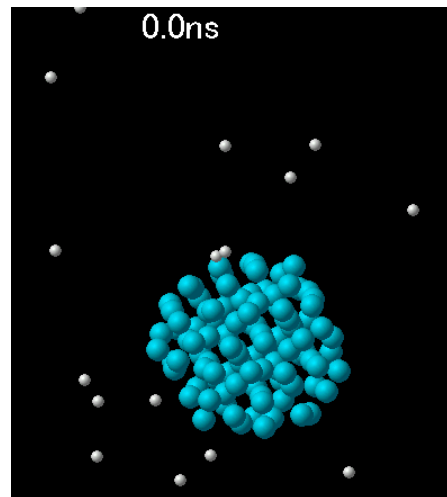
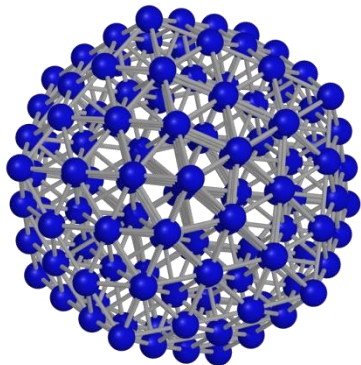
Guarantees that an atom interacts with at most one of the other atoms and its images.

Capability of MD

- ✓ **Structural:** crystal structure, defects (vacancies, interstitials, dislocations, grain boundaries, precipitates)
- ✓ **Thermodynamic:** equation of state, heat capacities, thermal expansion, free energies
- ✓ **Mechanical:** elastic constants, cohesive and shear strength, elastic and plastic deformation, fracture toughness
- ✓ **Vibrational :** phonon dispersion, vibration frequencies, molecular spectroscopy
- ✓ **Transport :** diffusion, viscous flow, thermal conduction



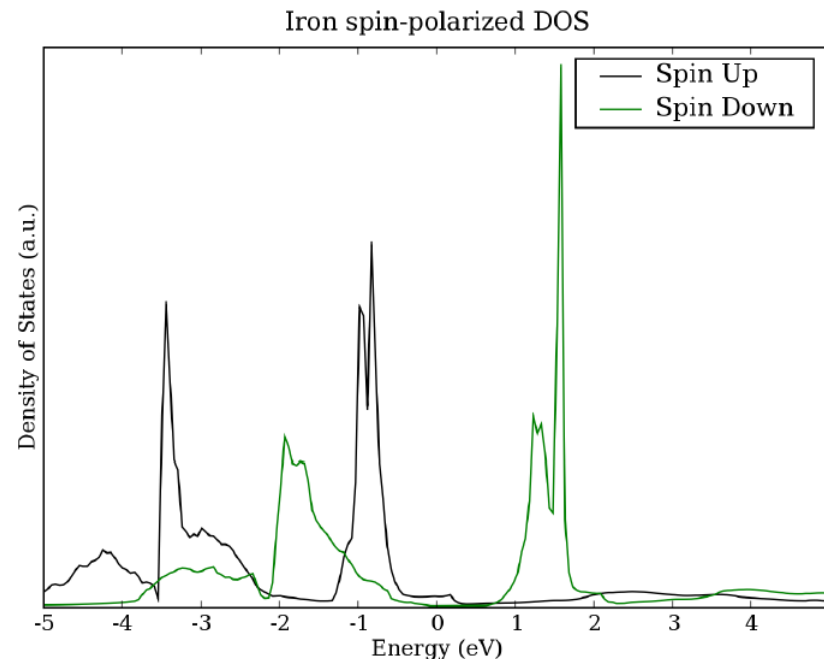
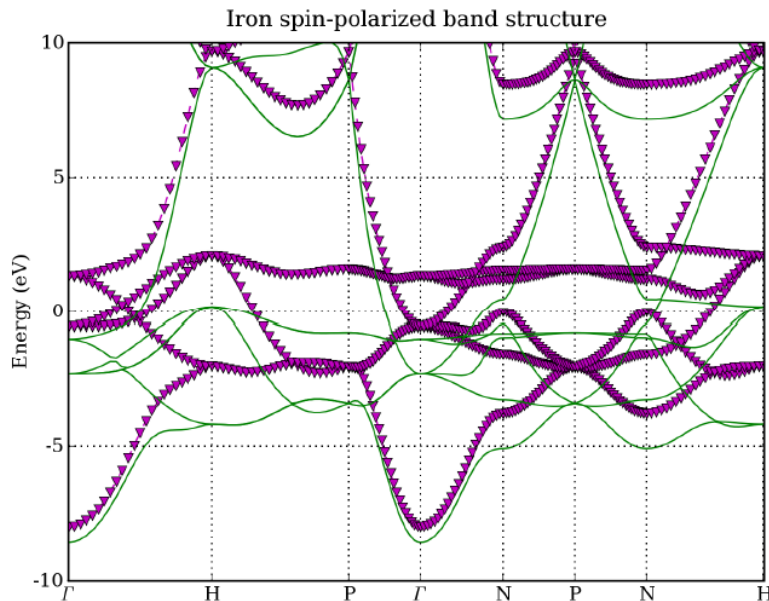
Ag on H-Si(111) vs free standing Ag particle



Growth of SWCNT via CVD

Limitation of MD

- There are properties which classical MD cannot calculate because electrons are involved.
- To treat electrons properly one needs quantum mechanics. In addition to electronic properties, optical and magnetic properties also require quantum mechanical (first principles or *ab initio*) treatments.



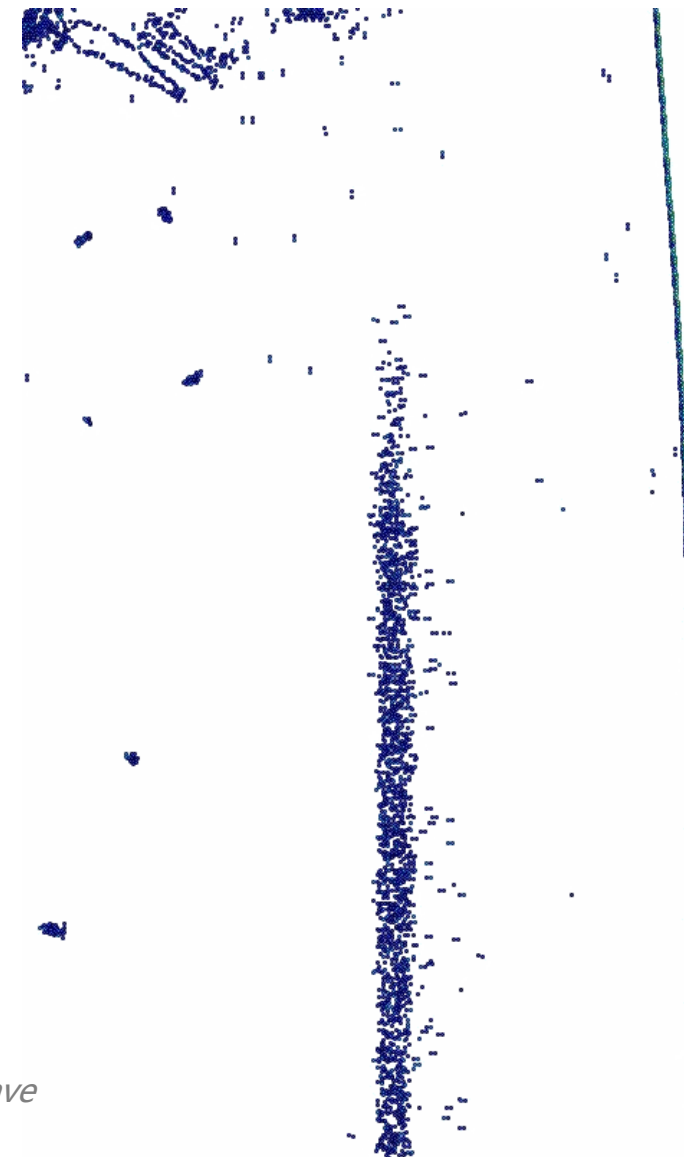
■ Time scale: *ns*

■ Length scale: *nm*

Yet MD is still a unique tool ...

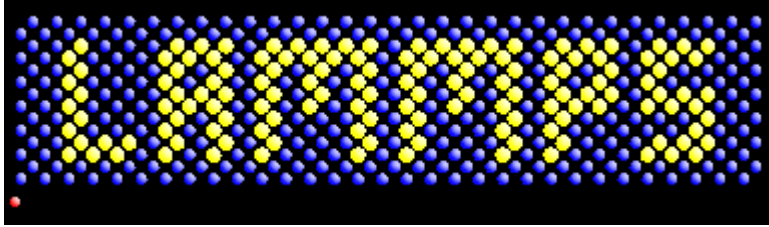
- **Unified** study of all physical properties
- Several **hundred** particles are **sufficient** to simulate **bulk** matter
- **Direct link** between potential model and physical properties
- **Complete control** over input, initial and boundary conditions
- **Detailed atomic trajectories**

(adapted from S. Yip, Nuclear Engrg./MIT.)



Video from: V. V. Stegailov and A. V. Yanilkin, *Structural transformations in single-crystal iron during shock-wave compression and tension: Molecular dynamics simulation*, J of Expt and Theoretical Physics, 104, 928-935 (2007).

Popular codes



GROMACS FAST.
FLEXIBLE.
FREE.



LatGen



Accelrys Discovery Studio Visualizer 3.0
Discovery Studio Visualization tools you can to view molecular data



Atomeye

VMD
Visual Molecular Dynamics

Home work



Extend the MD_Harmonic_Oscillator code to simulate a one-dimension periodic array of particles, where particles interact with their nearest neighbors only via a spring.

- You can use any programming language; (python, c++, fortran, perl, java, julia...)
- You would need to use arrays;
- You would need to write the thermal info to a file;
- You could visualize the output with gnuplot, matplotlib, xmgrace, origin...
- Write a report: including your source code and simulation results
- Due: Oct 13th, 2022

Next Lecture:

Interatomic Potentials

