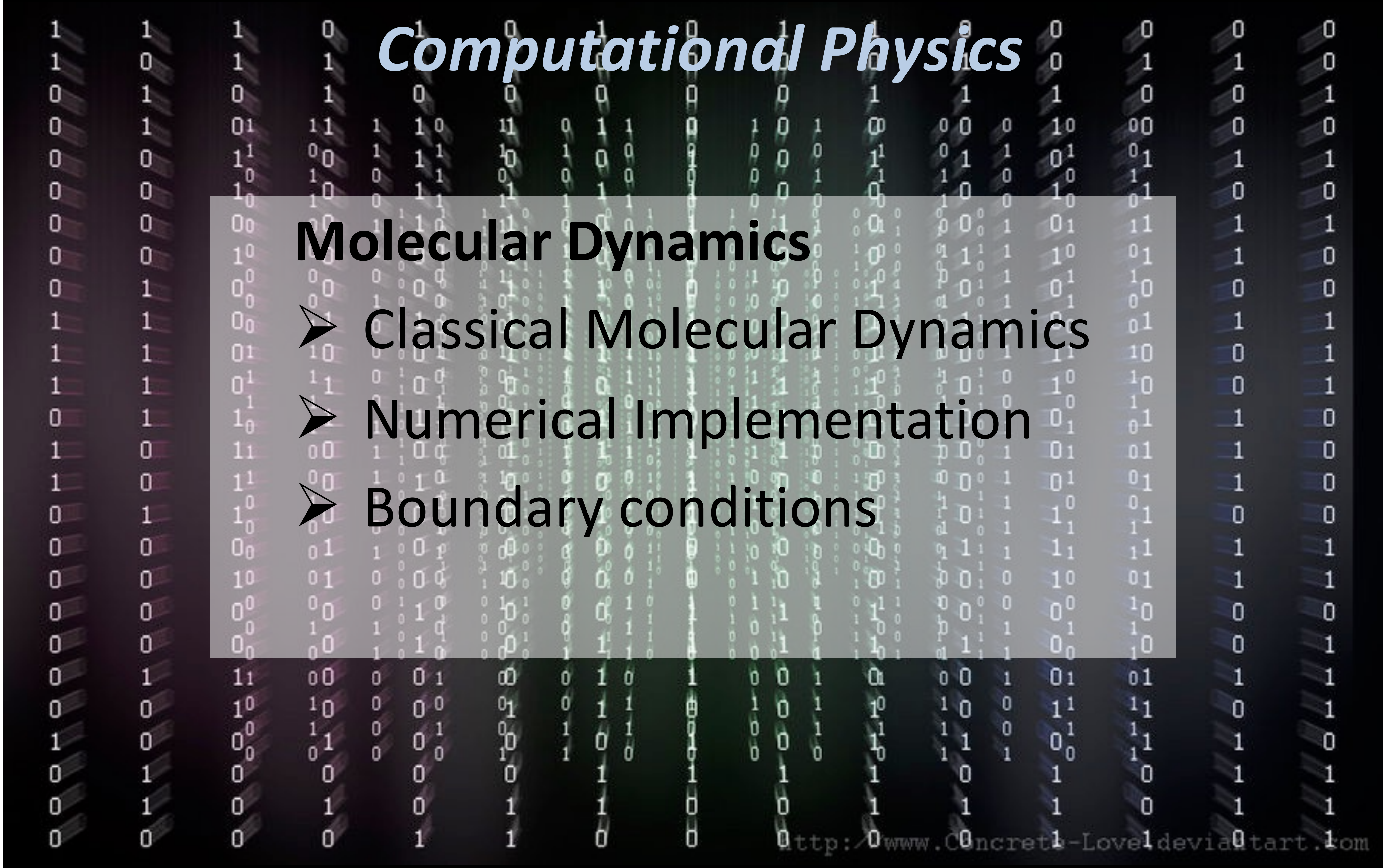


Computational Physics

Molecular Dynamics

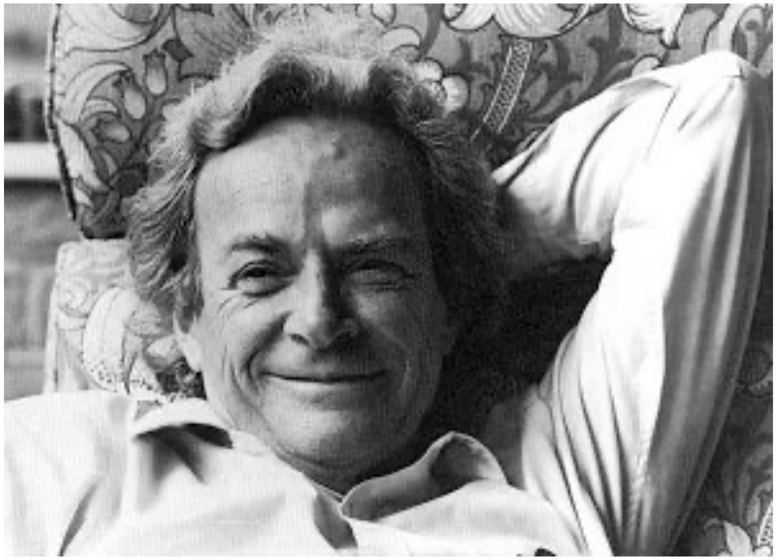
- ➢ Classical Molecular Dynamics
- ➢ Numerical Implementation
- ➢ Boundary conditions

What is Molecular Dynamics?

- “The science of simulating the motions of a system of particles” (Karplus & Petsko)
- Systems can be
  - as small as an atom
  - as large as a galaxy
- Equations of motion

↓

- Time evolution

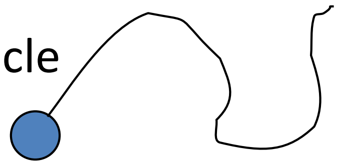


“everything that living things do can be understood in terms of the jigglings and wigglings of atoms.”

The Feynman Lectures in Physics vol. 1, 3-6 (1963)

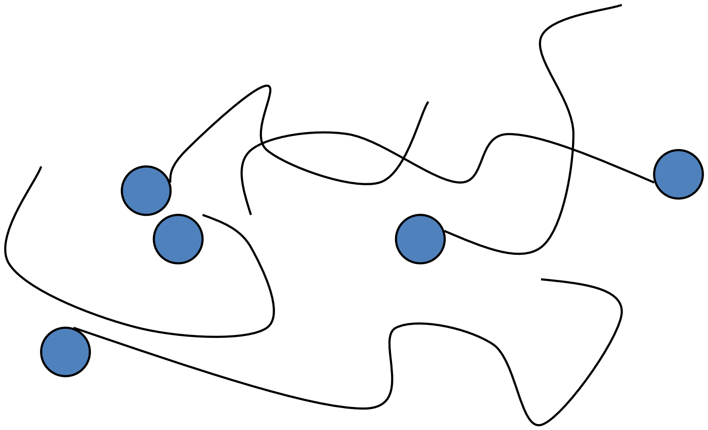
Molecular Dynamics (MD)

- Knowledge of the interaction potential for the particles ➔forces



One particle

easy analytically



Many particles impossible analytically

- Classical Newtonian equations of motion
- Many particle systems ➔simulation
- Maxwell-Boltzmann averaging process for thermodynamic properties: time averaging

MD simulations

- are computer N-body simulations of physical movements of atoms and molecules
- Their trajectories are determined by numerically solving the Newton's equations of motion for a system of interacting particles, where forces between the particles and potential energy are defined by *molecular mechanics force fields*. (also used for energy minimization in Monte Carlo simulations)
- Molecular mechanics uses classical mechanics to model molecular forces.

- Applications in **chemical physics**, **materials science,** and **bio physics**.

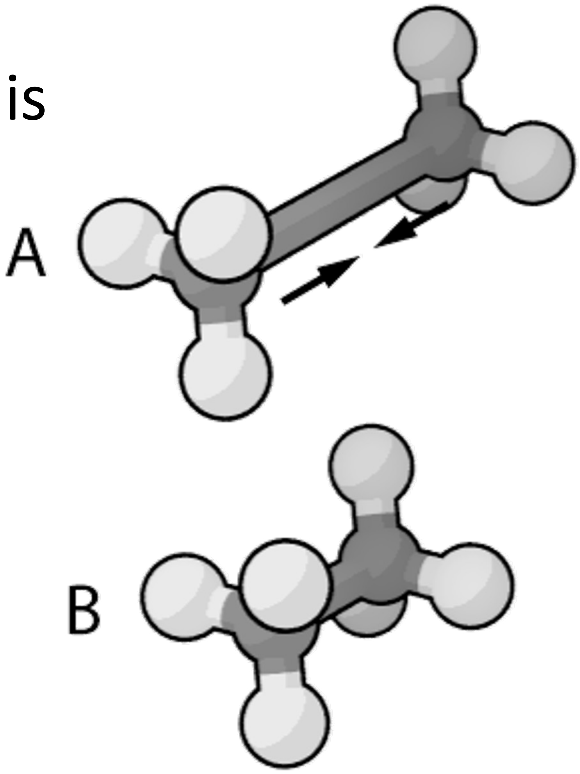
- Here only classical systems, not “quantum molecular dynamics”

All-atomistic MD simulations

Typical MD simulation

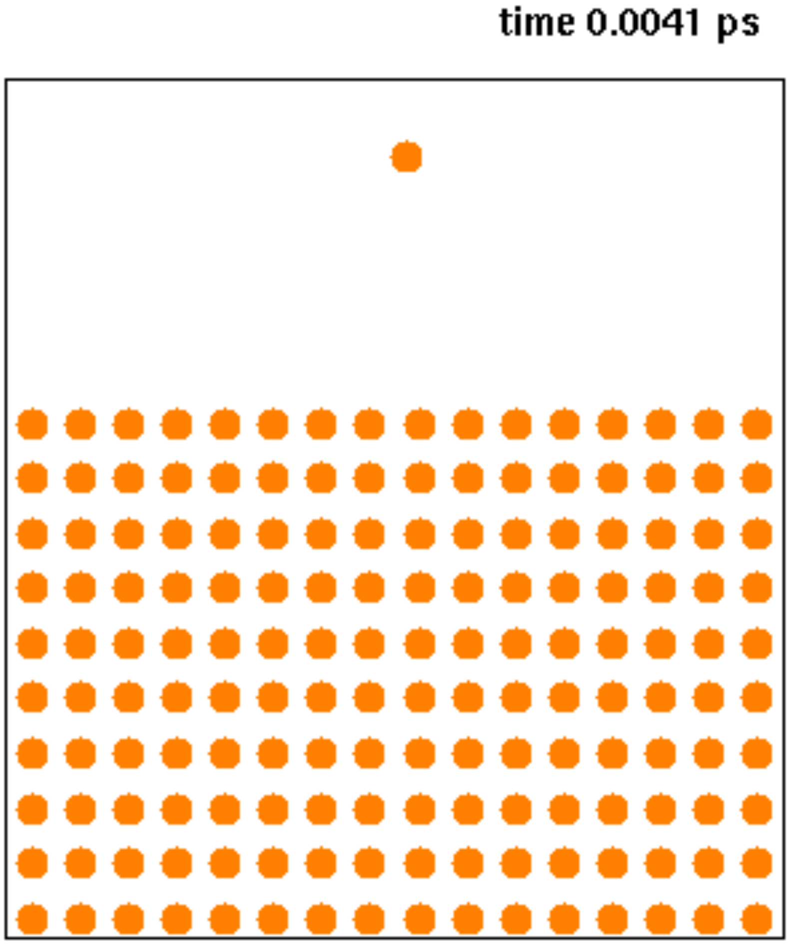
- Each atom is simulated as a single particle and is affected by the potential energy functions of every atom in the system
- Each particle is assigned a radius (typically the van der Waals radius), polarizability, and a constant net charge (generally derived from quantum calculations and/or experiment)
- Bonded interactions are treated as "springs" with an equilibrium distance equal to the experimental or calculated bond length

Variations are possible (e.g., consider dimers as "atoms")

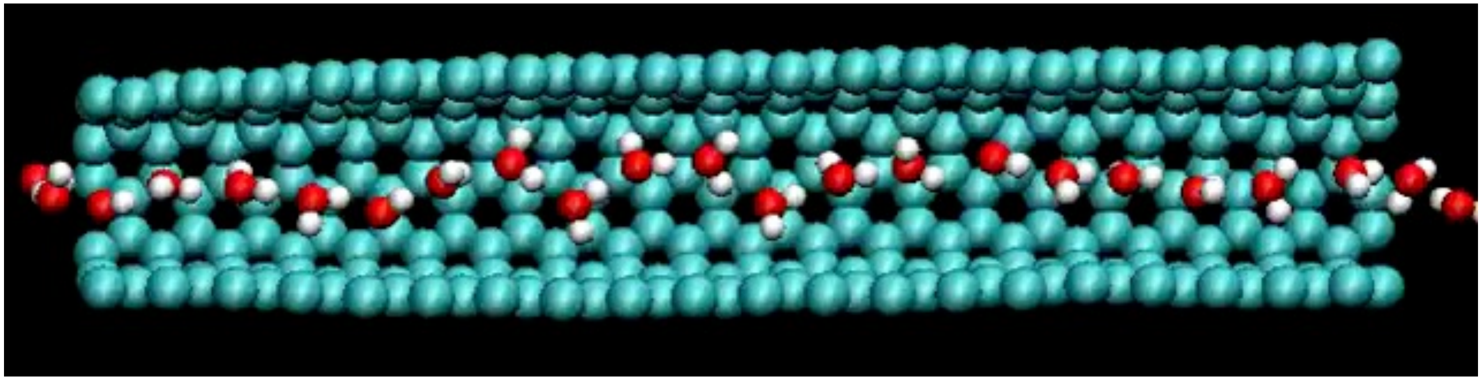


Example 1

Molecular dynamics simulation of the deposition of a single copper atom with a kinetic energy of 1 eV on a copper surface.



Example 2



Static charges cannot drive a continuous flow of water molecules through a carbon nanotube

Classical Molecular Dynamics

Here we consider a classical model system for molecular dynamics consisting of N particles with positions $\mathbf{r}_i \equiv \mathbf{r}_i(t)$, velocities $\mathbf{v}_i \equiv \mathbf{v}_i(t)$and masses m_i, where i =1, 2, … , N.

Newton's equations of motion:

$$m_i \ddot{r}_i = f_i(r_1, r_2, \ldots, r_N)$$

with forces: $f_i \equiv f_i(r_1, r_2, \ldots, r_N)$

$\boldsymbol{f}_i$ are vectors of same dimension as $\mathbf{r}_i$ & $\mathbf{v}_i$

We assume conservative forces, i.e.:

$$f_i(r_1, r_2, \ldots, r_N) = -\nabla_i U(r_1, r_2, \ldots, r_N)$$

Where the potential can be written as

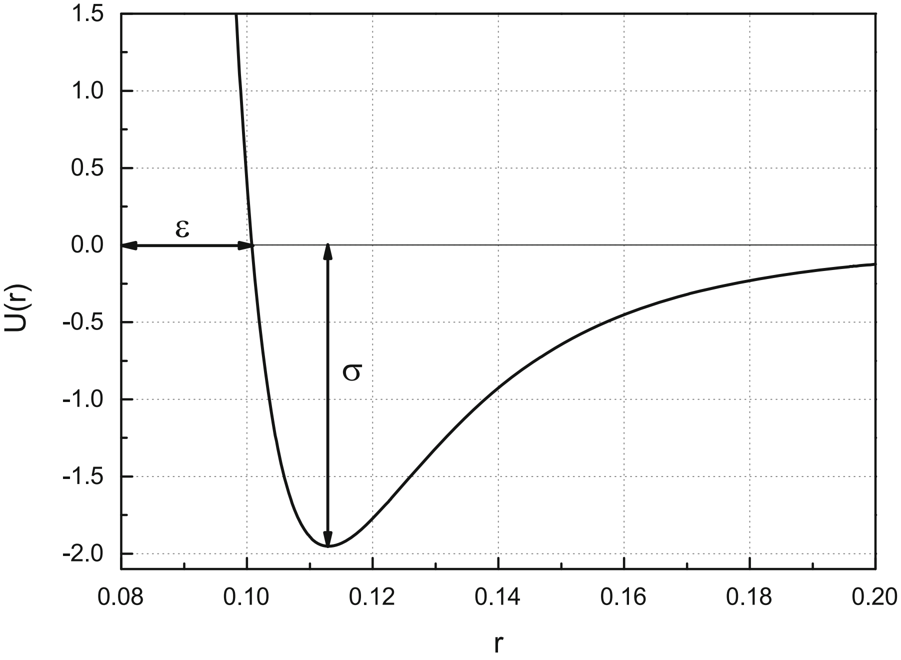
$$U = \frac{1}{2} \sum_i \sum_{j \neq i} U_{ij} + U_{\text{ext}}$$

Potentials

For the two-body Kepler problem we used the central potential: ∝ -1/r
Most common the Lennard-Jones potential

$$U(|r|) = 4\sigma\left[\left(\frac{\epsilon}{|r|}\right)^{12} - \left(\frac{\epsilon}{|r|}\right)^{6}\right]$$

where σ and ϵ are real parameters:



developed to model the interaction between neutral atoms or molecules

- $|r|^{-12}$ describes Pauli repulsion
- $-|r|^{-6}$ van-der-Waals attraction

Distance between particles:

$$r_{ij} = |r_i - r_j| = |r_j - r_i| = r_{ji}$$

→ two-body potential: $U_{ij} \equiv U(r_{ij})$

Forces

This results in the force: $f(|r|) = -\nabla_r U(|r|) = \frac{24\sigma}{|r|^2}\left[2\left(\frac{\epsilon}{|r|}\right)^{12} - \left(\frac{\epsilon}{|r|}\right)^{6}\right] r$

In general, we can write the forces in the Newton equation:

$$\begin{aligned}
f_i &= -\nabla_i U \\
&= -\nabla_i \left(\frac{1}{2}\sum_k \sum_{l\neq k} U_{kl} + U_{\text{ext}}\right) \\
&= -\sum_{j\neq i} \nabla_i U_{ij} - \nabla_i U_{\text{ext}} \\
&= \sum_{j\neq i} f(r_{ij}) + f^i_{\text{ext}} \\
&= \sum_{j\neq i} f_{ij} + f^i_{\text{ext}}\ ,
\end{aligned}$$

Rewrite

We introduce: $R = (r_1, r_2, \ldots, r_N)^T \quad V = (v_1, v_2, \ldots, v_N)^T = \dot{R}$

$$F = (f_1/m_1, f_2/m_2, \ldots, f_N/m_N)^T$$

Which results in the Newton equation in form: $\boxed{\ddot{R} = F}$

Or in the coupled first order equations: $\boxed{\begin{pmatrix} \dot{R} \\ \dot{V} \end{pmatrix} = \begin{pmatrix} V \\ F \end{pmatrix}}$

Discretization

As usual, we discretize time as $t_k = k\Delta t$, $k \in \mathbb{N}$, and use the subscript k to indicate a dependence on t_k, e.g. $R_k = R(t_k)$

→ symplectic Euler

$$\begin{pmatrix} R_{k+1} \\ V_{k+1} \end{pmatrix} = \begin{pmatrix} R_k \\ V_k \end{pmatrix} + \begin{pmatrix} V_{k+1} \\ F_k \end{pmatrix} \Delta t$$

which combines to

$$R_{k+1} = R_k + V_k \Delta t + F_k \Delta t^2$$

Using the backward difference for V_k gives the recursion

$$\boxed{R_{k+1} = 2R_k - R_{k-1} + F_k \Delta t^2}$$

valid for k>1. for k=1 we use the Taylor expansion

$$\boxed{R_1 = R_0 + \Delta t V_0 + \frac{1}{2} F_0 \Delta t^2}$$

this is the Störmer-Verlet algorithm

follows also from central difference

$$\ddot{R}_k \approx \frac{R_{k+1} - 2R_k + R_{k-1}}{\Delta t^2} = F_k$$

Leap-frog method

using the central rectangular rule gives:

$$R_{k+1} = R_k + V_{k+\frac{1}{2}}\Delta t$$

and similarity we define

$$V_{k+\frac{1}{2}} = V_{k-\frac{1}{2}} + F_k\Delta t$$

This is the leap-frog algorithm together with the initialization (obtained by Taylor series)

$$V_{\frac{1}{2}} = V_0 + \frac{\Delta t}{2}F_0$$

velocity Verlet algorithm

We expand:

$$R_{k+1} = R_k + V_k \Delta t + \frac{1}{2} F_k \Delta t^2$$

and

$$V_{k+1} = V_k + \frac{1}{2}\left(F_k + F_{k+1}\right) \Delta t$$

where the remainder is approximated by the geometric mean at time t_k and t_{k+1}

I.e.: First calculate R_{k+1}, then using this F_{k+1}, and finally V_{k+1}

Remarks:

i. Symplectic Euler is time-reversal symmetric Δt→-Δt, position updates are highly accurate, but velocity updates not
ii. The latter is improved by leap-frog and velocity Verlet. Both are not time-reversal invariant.
iii. velocity Verlet is most popular

Numerical Implementation

Typical structure of a molecular dynamics code has three crucial steps:

- Initialization,
- start simulation and equilibrate,
- continue simulation and store results.

Important are appropriate boundary conditions

Boundary conditions

Two possibilities:

(i) finite system, implementation of boundary conditions might be straightforward. E.g. N particles within a finite box of reflecting boundaries: simply propagate particle-coordinates in time and if a particle tries to leave the box, correct its trajectory according to a reflection law. The velocity is adjusted accordingly.

Example:

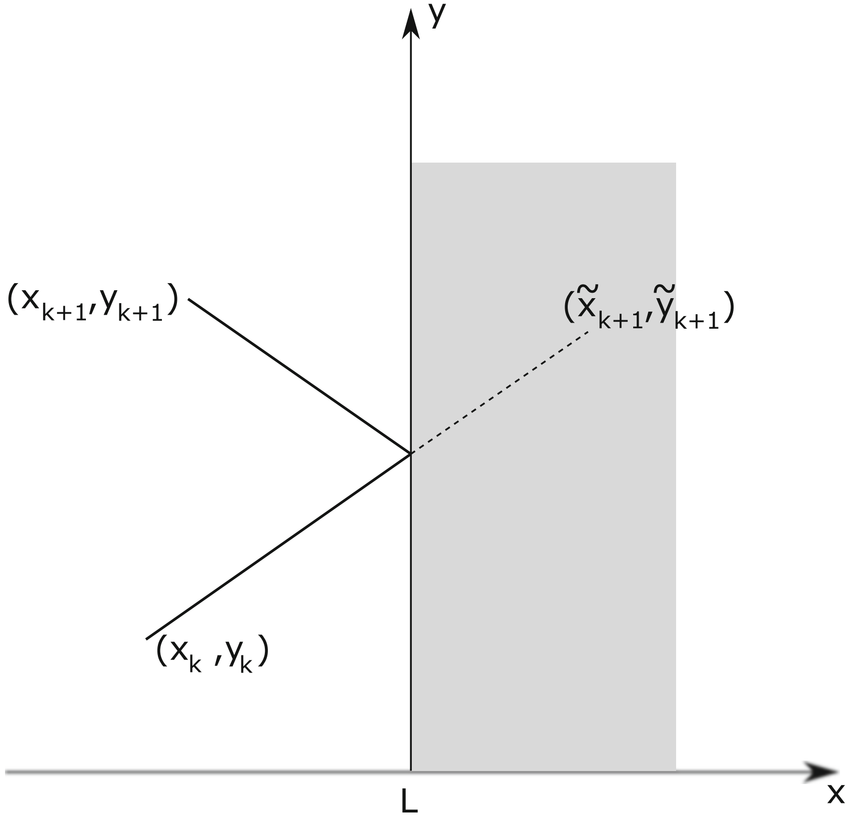
$$r_{k+1} = \begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} L - (\tilde{x}_{k+1} - L) \\ \tilde{y}_{k+1} \end{pmatrix}$$

$$v_{k+1} = \begin{pmatrix} v_{k+1,x} \\ v_{k+1,y} \end{pmatrix} = \begin{pmatrix} -\tilde{v}_{k+1,x} \\ \tilde{v}_{k+1,y} \end{pmatrix}$$

where L is the size of the box

$$\tilde{x}_{k+1},\ \tilde{y}_{k+1},\ \tilde{v}_{k+1,x} \text{ and } \tilde{v}_{k+1,y}$$

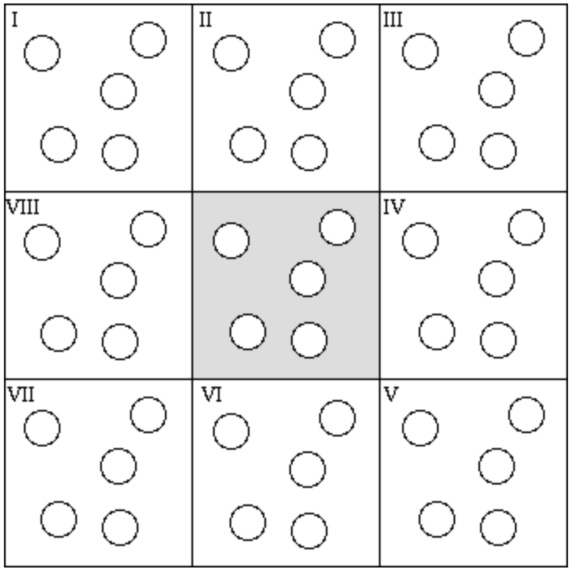
would be the coordinates/velocities in absence of border



…

(ii) The system is not confined. This situation is entirely different.

- could be approximated by finite, but large system → finite volume effects
- use periodic boundary conditions: If a particle leaves the box, it enters the box at the same time on the opposite side. This means that a finite system is surrounded by an infinite number of completely identical replicas of the system, where the forces are allowed to act across boundaries → calculating the force on one particle requires the evaluation of an infinite sum. Numerically not manageable and we have to find ways to truncate the sum, e.g., only nearest neighbor cells, but depends on the range of interaction forces.



If total velocity is non-zero (adds to kinetic energy)

$$v_{\text{tot}} = \sum_{i=1}^{N} v_i \neq 0$$

one can introduce a shift: (so the system is at rest, as it should be for closed systems)

$$v_i' = v_i - \frac{1}{N} v_{\text{tot}}$$

Initialization and Equilibration

The equipartition theorem states that every degree of freedom contributes $k_B T/2$ to the total kinetic energy. E.g. for N particles in d dimensions we have d(N-1) degrees of freedom if the total velocity is zero (closed system):

$$E_{\mathrm{kin}} = \frac{1}{2}\sum_{i=1}^{N} m_i v_i^2 = \frac{d(N-1)}{2} k_{\mathrm{B}} T$$

which we can use to define the temperature of a system:

$$k_{\mathrm{B}} T = \frac{1}{d(N-1)} \sum_{i=1}^{N} m_i v_i^2$$

Often T is an input parameter, not an observable. This means we should rescale the velocities to get the correct temperature (might need to be done several times to get a constant temperature; does not change the total velocity):

$$v_i' = \lambda v_i$$

…

The “rescaled” temperature is then

$$k_B T' = \frac{\lambda^2}{d(N-1)} \sum_{i=1}^{N} m_i v_i^2$$

which we can use to define the rescaling parameter λ:

$$\lambda = \sqrt{\frac{d(N-1)k_B T'}{2E_{\text{kin}}}}$$

The choice of initial conditions influence the equilibration behavior. If a temperature is defined, initial velocities should be chosen according to the Maxwell-Boltzmann distribution, i.e., with velocities distributed according to pdf

$$p(|v|) \propto |v|^2 \exp\left(-\frac{m|v|^2}{2k_B T}\right)$$

(read appendix E on pdfs)

…

Question: How do we check if we reached thermal equilibrium?

In statistical mechanics one typically has time-dependent observables O(t). Its expectation value is defined as:

$$\langle O \rangle = \lim_{\tau \to \infty} \frac{1}{\tau} \int_0^{\tau} \mathrm{d}t O(t)$$

Since we do not know O(t) analytically and cannot wait indefinitely, we use

$$\langle O \rangle \approx \overline{O} = \frac{1}{n} \sum_{j=k+1}^{k+n} O(t_j)$$

If n and k are sufficiently large, we can assume that this average has converged. This means we need to find the k for which <O> does not change anymore for k'>k (for sufficiently large n).

Units

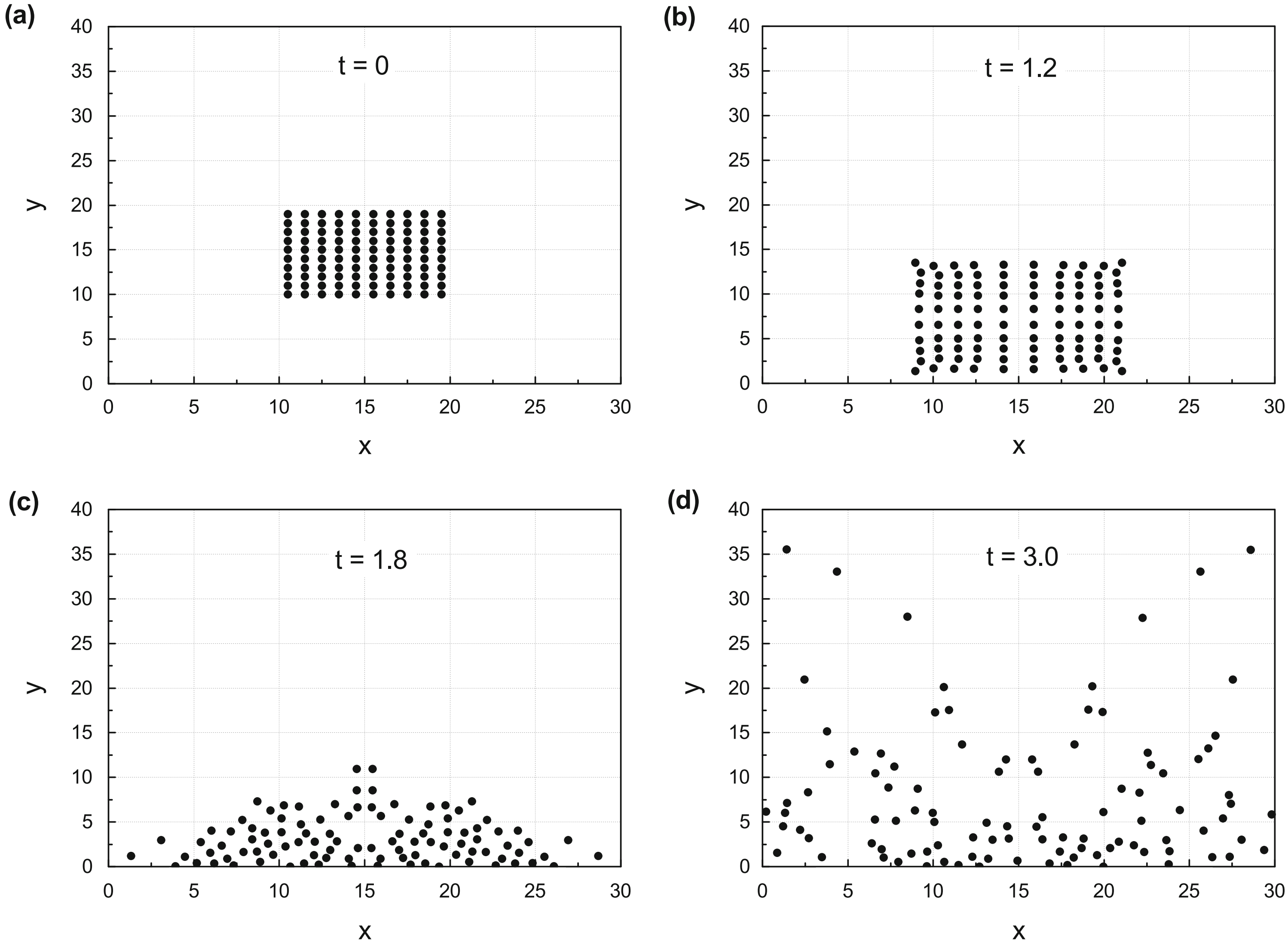
As mentioned before it is often useful to use rescaled, dimensionless units instead of *natural* units to avoid numerical instabilities. E.g.

$$r' = \frac{r}{L}$$

which would result in positions being in the interval [0; 1].

One has to be careful that not all units can be rescaled independently.
(e.g. the energy scale and time scale are related)

Example



Lab today & Thursday

Implement the molecular dynamics of N particles in a box:

- N=100, L=30
- Initially arranged in square (lower left corner at (10.5,10) and lattice spacing 1)
- gravity acts on each particle
- particles interact by Lennard-Jones (LJ) potential

use leap-frog or velocity Verlet algorithms

Parameters: $\sigma=\epsilon=1$, m=1, g=9.81, $\Delta t=10^{-3}$, $N_t=5000$ (requires ~$3\cdot 10^7$ LJ potential calculations)