Exercise 11
Molecular Dynamics and Cell Lists

High Performance Computing for Science and Engineering I

December 9, 2016

CSElab

Computational Science & Engineering Laboratory

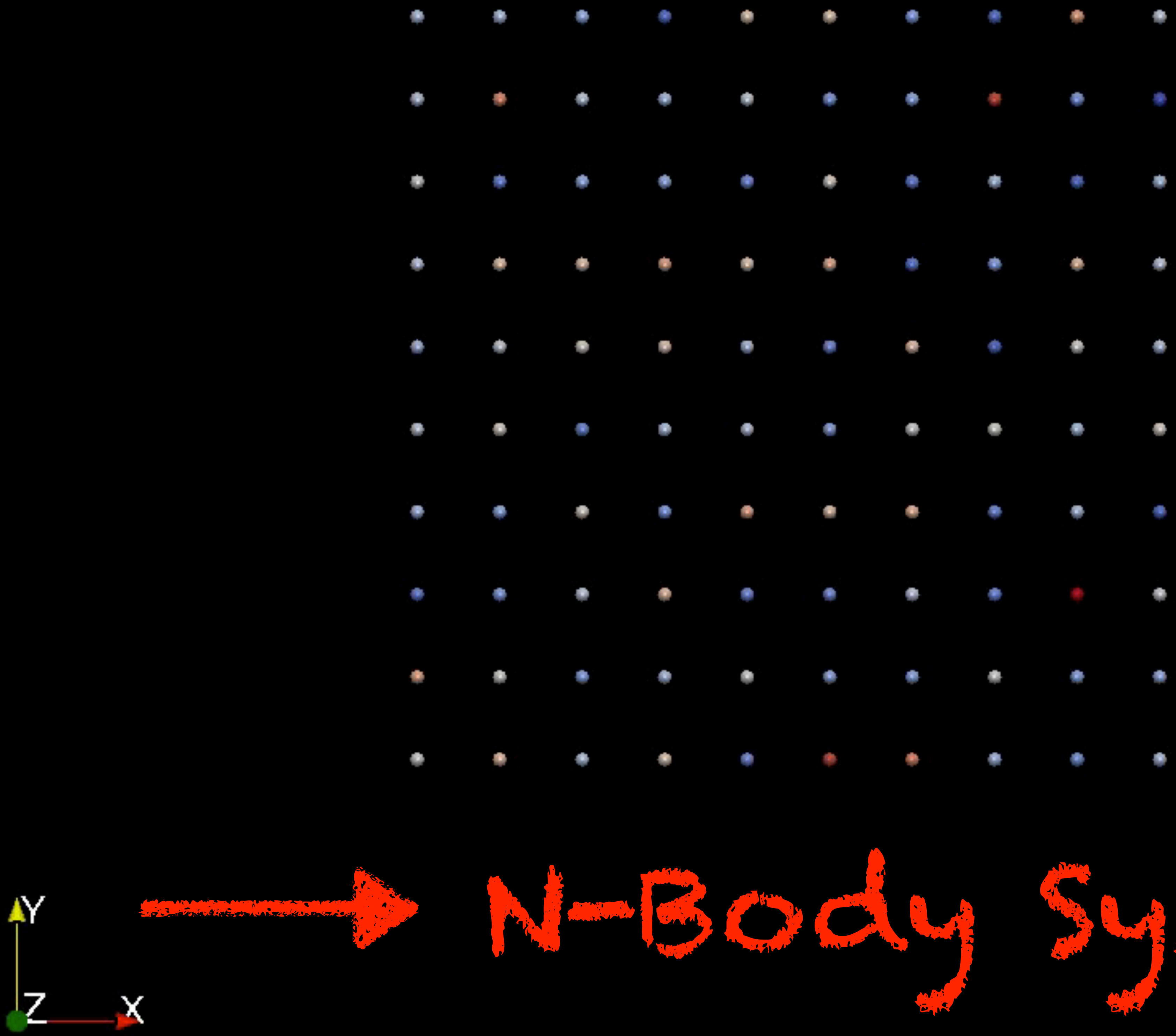
http://www.cse-lab.ethz.ch

ETH

Eidgenössische Technische Hochschule Zürich

Swiss Federal Institute of Technology Zurich

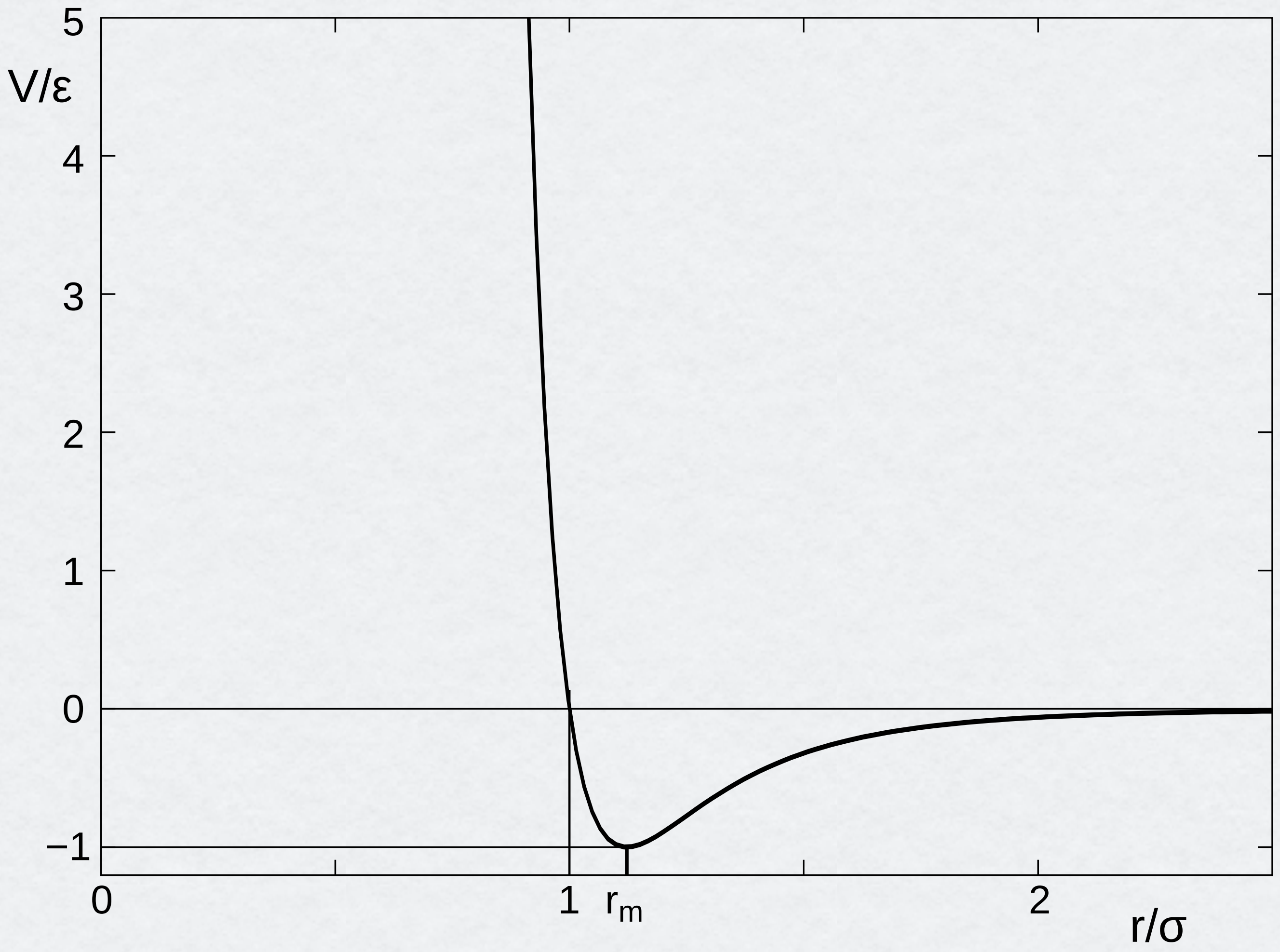
Goal: Motion of Interacting Molecules



→ N-Body System

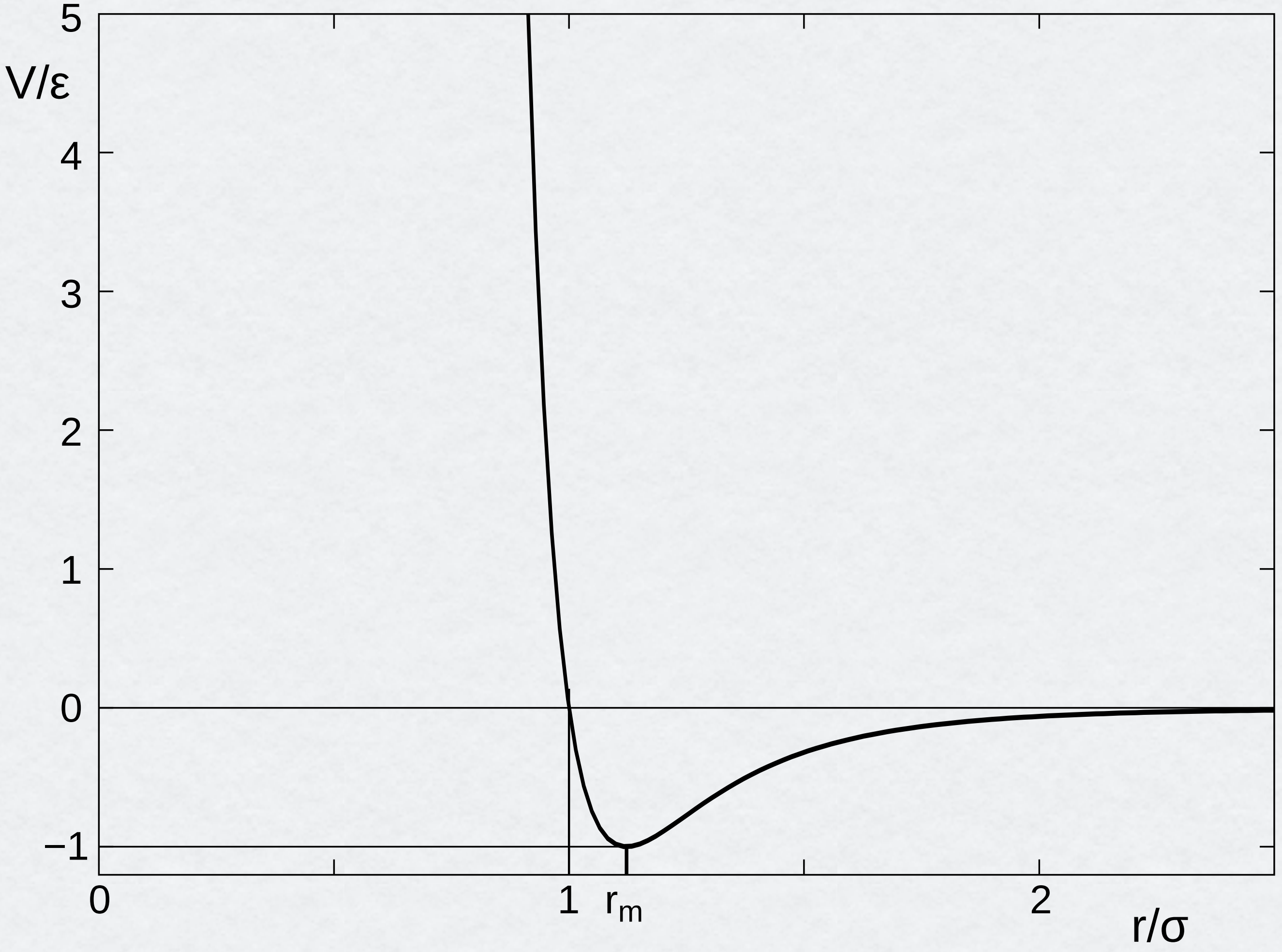
Lennard-Jones Potenial

$$V_{LJ}(r) = \varepsilon\left(\left(\frac{r_m}{r}\right)^{12} - 2\left(\frac{r_m}{r}\right)^{6}\right)$$



Lennard-Jones Potenial

$$V_{LJ}(r) = \varepsilon\left(\left(\frac{r_m}{r}\right)^{12} - 2\left(\frac{r_m}{r}\right)^{6}\right)$$



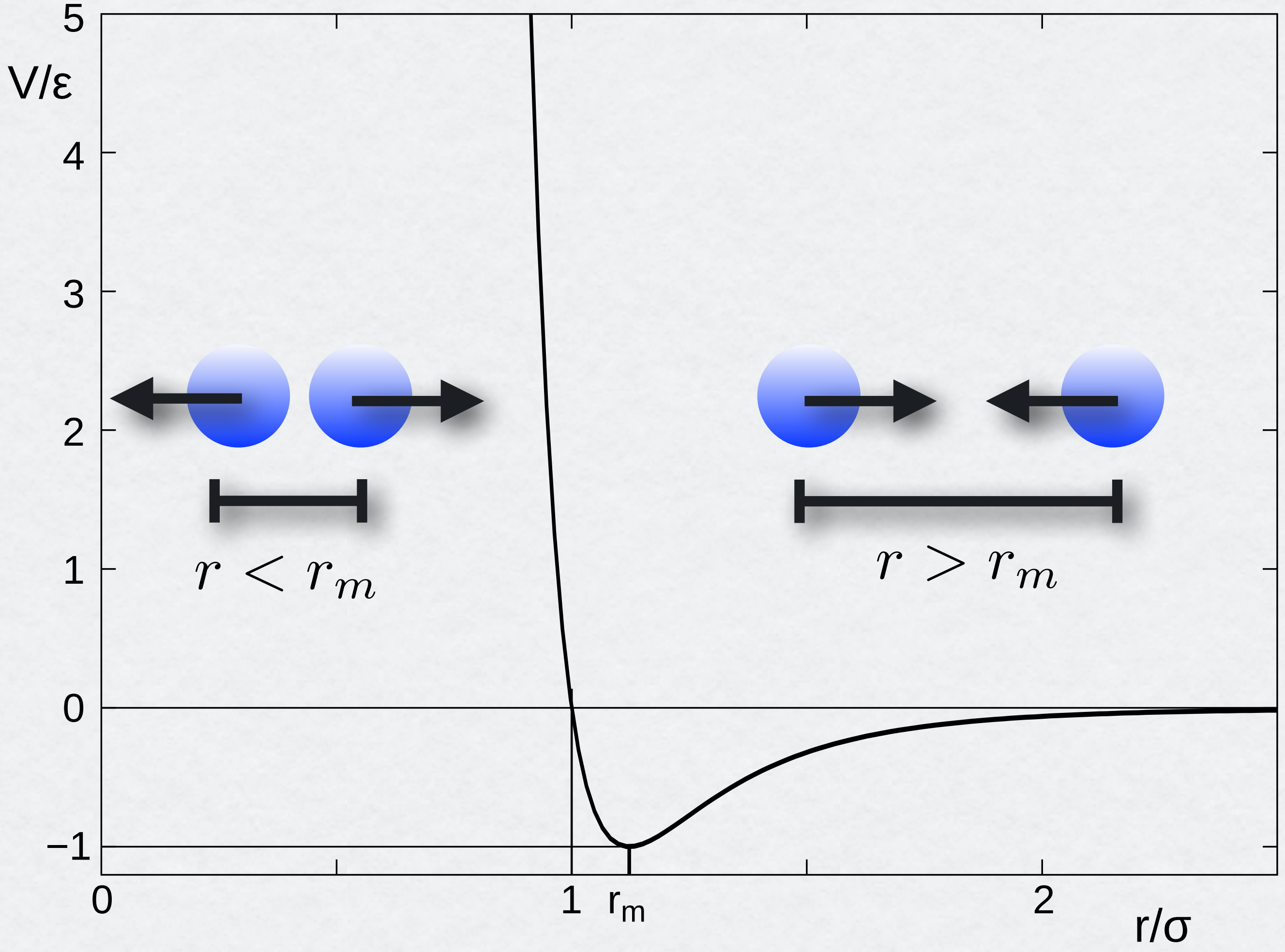
➡ force between molecules

$$\boxed{\vec{F}_{LJ}(\vec{x}_i, \vec{x}_j) = -\nabla V_{LJ}(\vec{x}_i, \vec{x}_j)}$$

$$r = \|\vec{r}\| = \|\vec{x}_i - \vec{x}_j\|$$

Lennard-Jones Potenial

$$V_{LJ}(r) = \varepsilon\left(\left(\frac{r_m}{r}\right)^{12} - 2\left(\frac{r_m}{r}\right)^{6}\right)$$



➡ force between molecules

$$\boxed{\vec{F}_{LJ}(\vec{x}_i, \vec{x}_j) = -\nabla V_{LJ}(\vec{x}_i, \vec{x}_j)}$$

$$r = \|\vec{r}\| = \|\vec{x}_i - \vec{x}_j\|$$

N-Body Solver

- Newton's second law

$$\frac{\mathrm{d}\vec{x}_i}{\mathrm{d}t} = \vec{v}_i$$

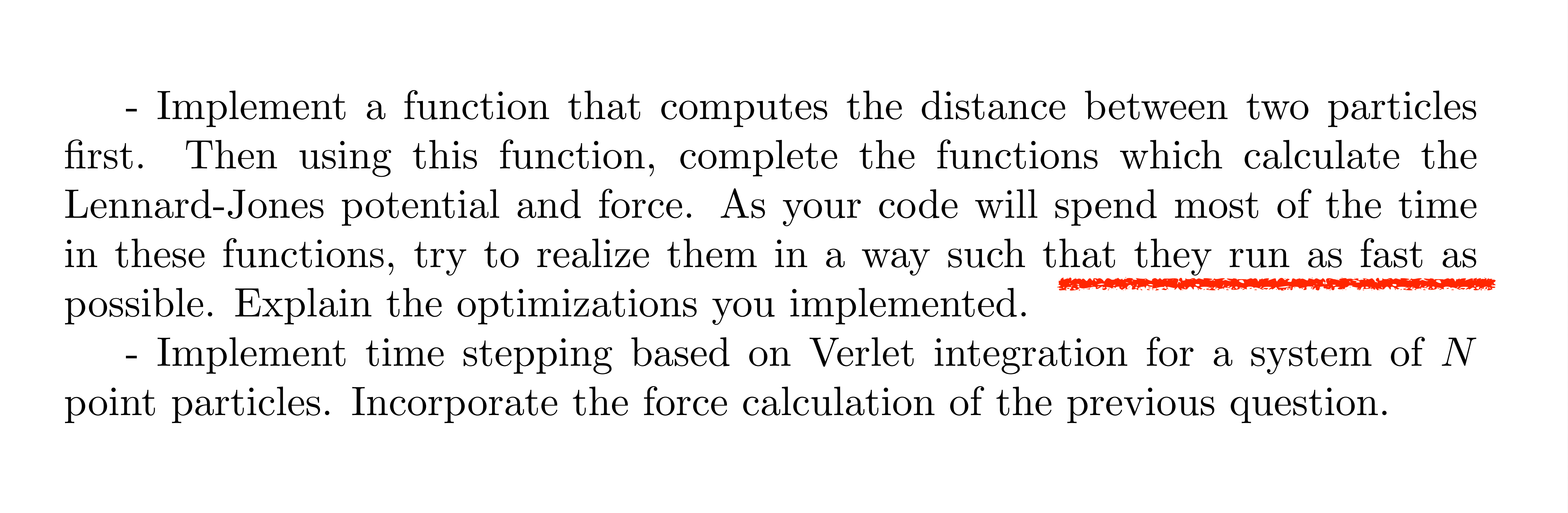
$$\frac{\mathrm{d}\vec{v}_i}{\mathrm{d}t} = \vec{a}_i = \frac{1}{m_i}\vec{F}_i$$

- Verlet integration

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

$$\vec{v}_i(t+\Delta t) = \vec{v}_i(t) + \frac{\Delta t}{2}\left(\vec{a}_i(t) + \vec{a}_i(t+\Delta t)\right)$$

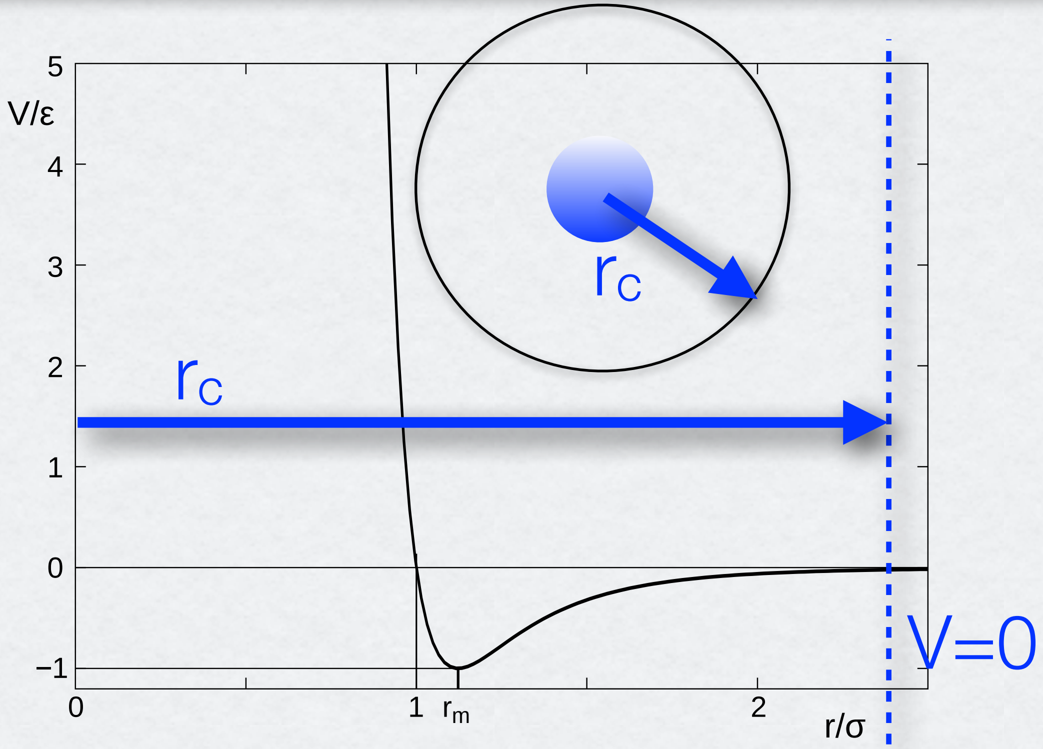
Question 1a and 1b



- Implement a function that computes the distance between two particles first. Then using this function, complete the functions which calculate the Lennard-Jones potential and force. As your code will spend most of the time in these functions, try to realize them in a way such that they run as fast as possible. Explain the optimizations you implemented.

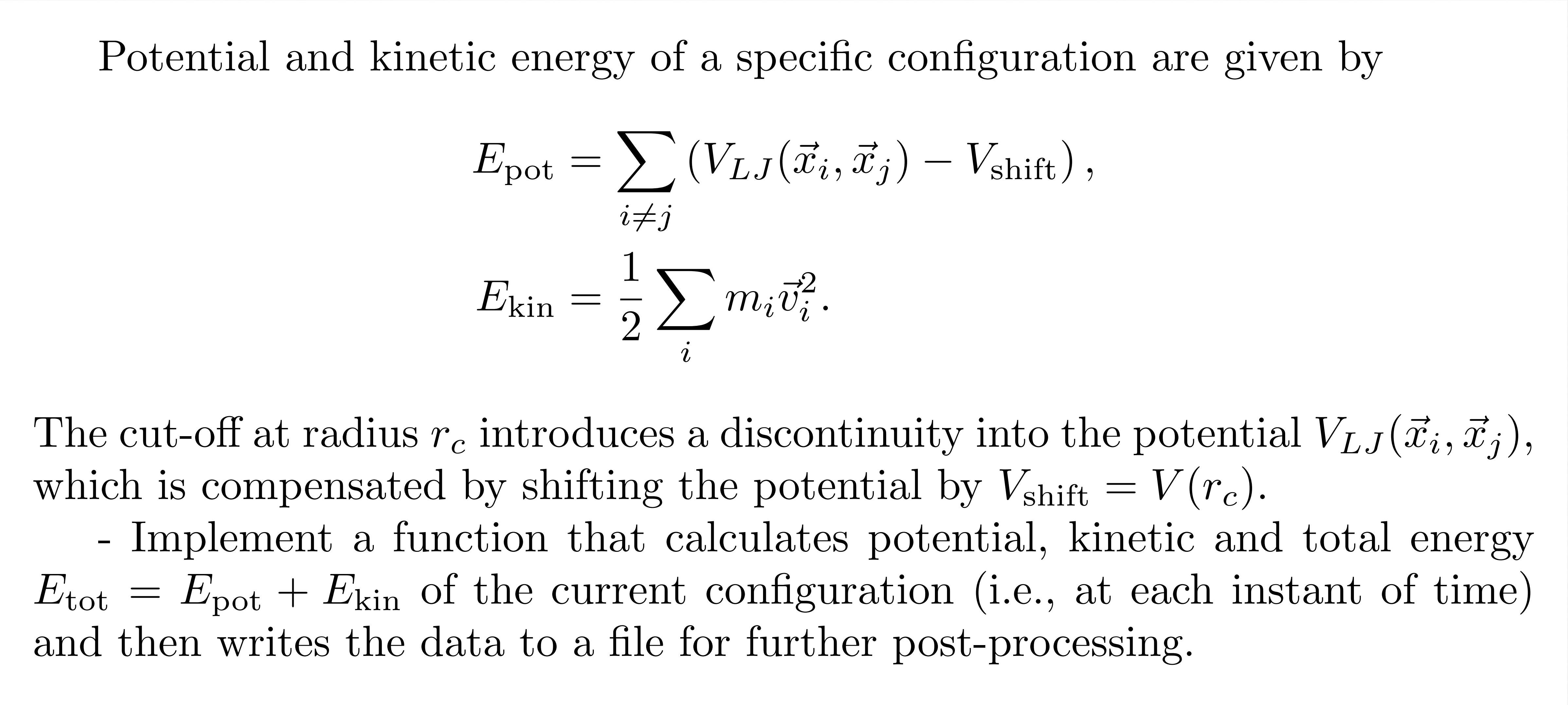
- Implement time stepping based on Verlet integration for a system of N point particles. Incorporate the force calculation of the previous question.

Question 1a and 1b - Hints



- truncation of Lennard-Jones potential at r_C
- fast code: avoid sqrt() and pow(), reduce divisions
- periodic boundaries

Question 1c



Potential and kinetic energy of a specific configuration are given by

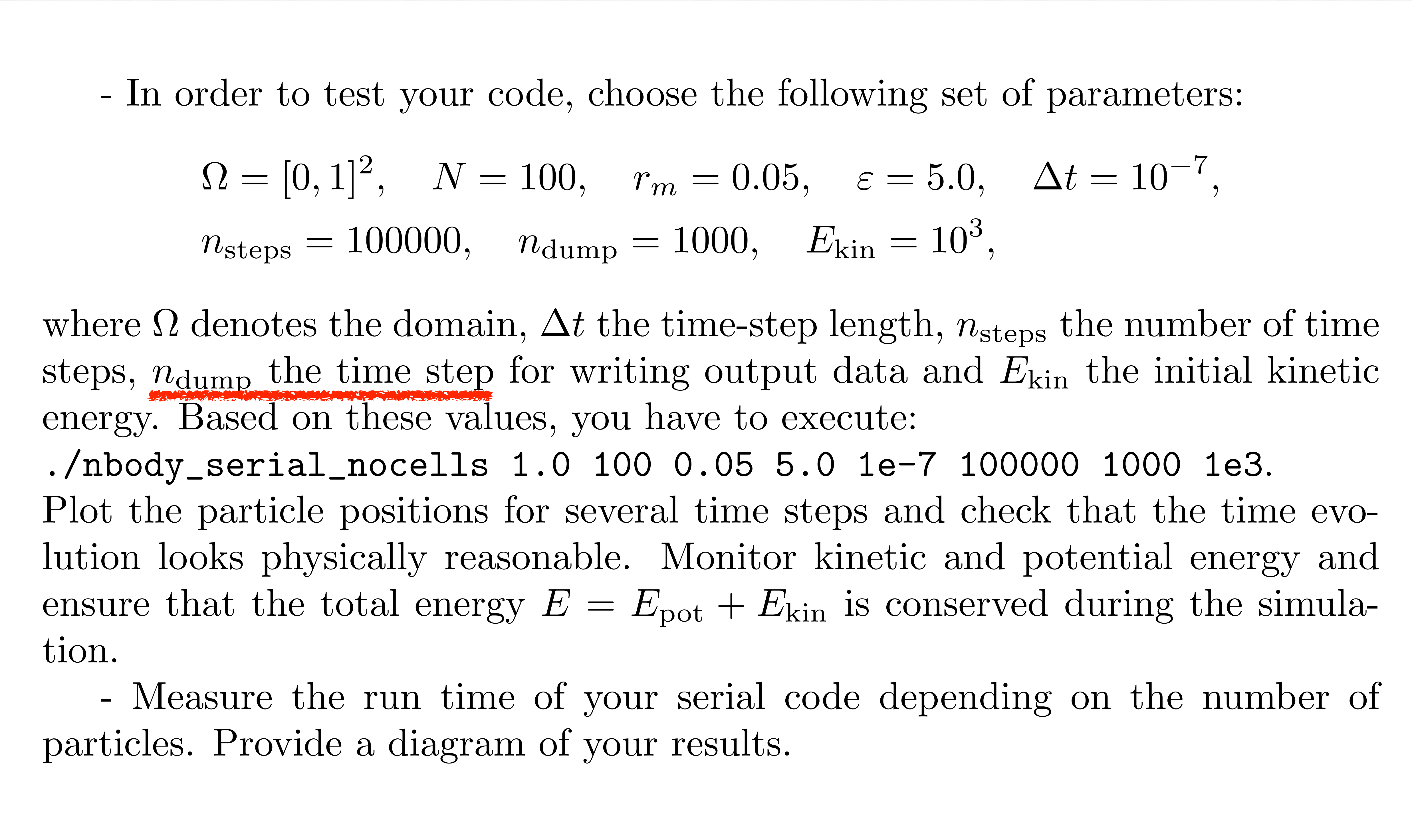
$$E_{\text{pot}} = \sum_{i \neq j} \left(V_{LJ}(\vec{x}_i, \vec{x}_j) - V_{\text{shift}} \right),$$

$$E_{\text{kin}} = \frac{1}{2} \sum_i m_i \vec{v}_i^2 .$$

The cut-off at radius r_c introduces a discontinuity into the potential $V_{LJ}(\vec{x}_i, \vec{x}_j)$, which is compensated by shifting the potential by $V_{\text{shift}} = V(r_c)$.

- Implement a function that calculates potential, kinetic and total energy $E_{\text{tot}} = E_{\text{pot}} + E_{\text{kin}}$ of the current configuration (i.e., at each instant of time) and then writes the data to a file for further post-processing.

Question 1d and 1e



- In order to test your code, choose the following set of parameters:

$$\Omega = [0,1]^2, \quad N = 100, \quad r_m = 0.05, \quad \varepsilon = 5.0, \quad \Delta t = 10^{-7},$$
$$n_{\text{steps}} = 100000, \quad n_{\text{dump}} = 1000, \quad E_{\text{kin}} = 10^3,$$

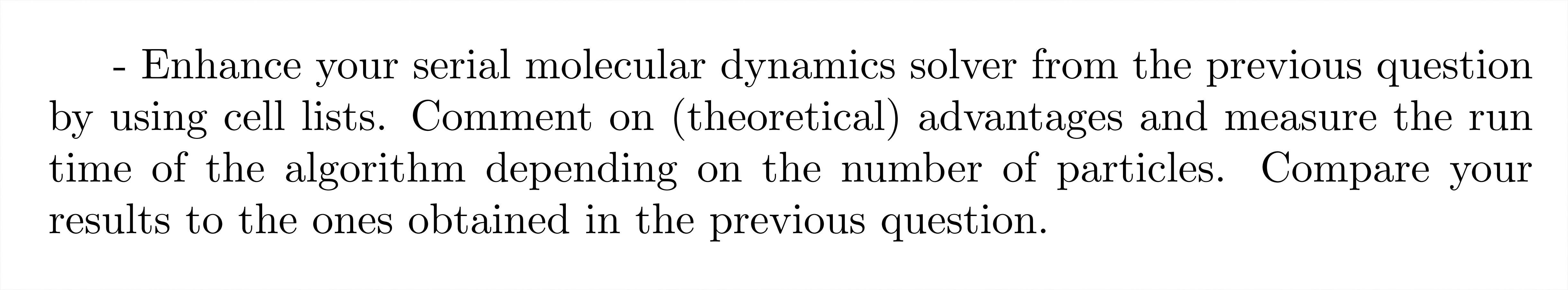
where Ω denotes the domain, Δt the time-step length, n_{steps} the number of time steps, n_{dump} the time step for writing output data and E_{kin} the initial kinetic energy. Based on these values, you have to execute:

```
./nbody_serial_nocells 1.0 100 0.05 5.0 1e-7 100000 1000 1e3.
```

Plot the particle positions for several time steps and check that the time evolution looks physically reasonable. Monitor kinetic and potential energy and ensure that the total energy $E = E_{\text{pot}} + E_{\text{kin}}$ is conserved during the simulation.

- Measure the run time of your serial code depending on the number of particles. Provide a diagram of your results.

Question 2a



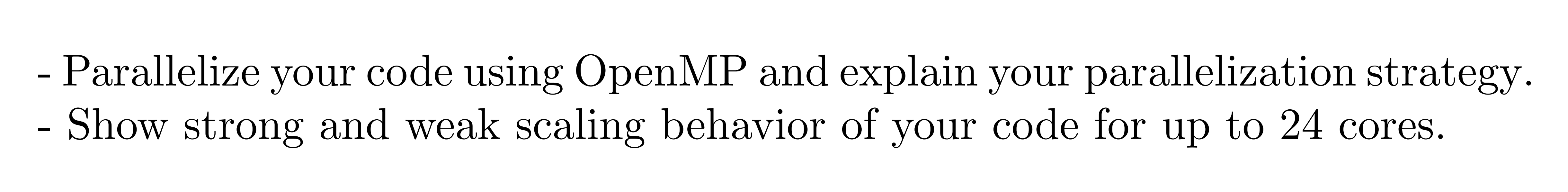
- Enhance your serial molecular dynamics solver from the previous question by using cell lists. Comment on (theoretical) advantages and measure the run time of the algorithm depending on the number of particles. Compare your results to the ones obtained in the previous question.

Question 2a - Hints

- extend “typedef std::vector< position> configuration;” to a vector (representing cells) of a vector (containing respective particles)
- consider periodic boundaries
- run time study: consider particle density

See the lecture for hints on cell lists!

Question 2b and 2c



- Parallelize your code using OpenMP and explain your parallelization strategy.
- Show strong and weak scaling behavior of your code for up to 24 cores.

Question 2b and 2c - Hints

- parallelization over cells
- consider force computation as well as update of positions and velocities
- not required: parallelization of assignment of particles to cells

Summary