

Molecular Dynamics: Lennard-Jones liquid

Florian Dommert, Nadezhda Gribova*

11. 11. 2009

ICP, Uni Stuttgart

This tutorial aims on providing a basic knowledge about **Molecular Dynamics (MD)** simulations. In computational physics it is always essential to understand the underlying mechanisms of the algorithms, that provide the model of your system. To this end the tutorial is designed to guide you step by step through the source code. It starts with an examination of the system and dicusses the basic aspects of an MD simulation.

1 The system setup

Inspect the code `readdat.c`. This file contains all information required for the generation of our system for the following MD simulations. It is important to get familiar with it, because you will have to change the setup parameters in a later part of the exercises. A description of the variables is provided in the file `system.c`. To concentrate on the most important things the initialization of the setup is not discussed here, broadly. However you always have to know how your simulation system was created.

1.1 Task

1. Try to understand the initialization processes encoded in `init.c` and describe it for yourself .

2 Integration of Newton's equation of motion

One of the basic concepts of MD simulations is the integration of the classical equations of motion. Therefore a suitable algorithm is required. Since the equations of motion are **time-reversible**, we have to use an appropriate integration scheme. One of the most simple approach is the so called Verlet algorithm:

$$x(t + \Delta t) = 2x(t) - x(t - \Delta t) + a(t) \cdot \Delta t^2 + \mathcal{O}(\Delta t^4), \quad (1)$$

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

Positions, velocities and the acceleration at time t are denoted by $x(t)$, $v(t)$, and $a(t)$, respectively. However, the error for $v(t)$ is Δt^2 rather than Δt^4 for positions. To overcome this problem different methods exist. A widely used approach is named Velocity-Verlet algorithm, that includes the contributions of the velocities directly in every integration step:

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

$$v(t + \Delta t) = v(t) + \frac{a(t + \Delta t) + a(t)}{2} \Delta t + \mathcal{O}(\Delta t^4). \quad (4)$$

*gribova@icp.uni-stuttgart.de

2.1 Task

1. Implement the Velocity Verlet algorithm in the file `integrate.c`. All required variables are already declared, just fill in the correct lines.

2.2 Homework (1 point)

1. Derive the Verlet algorithm. *Hint: equation 1 consists of two Taylor expansions to include time reversibility.*

3 Implementation of the Force Field

So far the interaction of the particles is not defined. Since we want to model Lennard-Jones particles an appropriate potential is required. A possible approach would be a simple Lennard-Jones (LJ) potential, but it has drawbacks concerning the calculation of the forces, due to their pairwise character. Hence a modified version, the truncated LJ (tLJ) potential, should be used here with $r_{\text{cutoff}} = 2.5\sigma$:

$$V_{LJ}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] \quad (5)$$

$$V_{tLJ}(r) = \begin{cases} V_{LJ}(r) - V_{LJ}(r_{\text{cutoff}}) & r \leq r_{\text{cutoff}} \\ 0 & r > r_{\text{cutoff}} \end{cases} \quad (6)$$

Application of this potential decreases the CPU time for the computationally most expensive part, the calculation of the forces. Though this introduces errors in the force calculation correction mechanism allow to perform a reasonable MD simulation.

3.1 Task

1. Modify the file `force.c` to enable the calculation of the **FORCES** according to a **tLJ** potential (see 6) with $\epsilon = 1.0$ and $r_{\text{cutoff}} = 2.5$. *Pay attention: So far we have just talked about potentials and not FORCES!* Like in section 2 no variables need to be declared, only insertion of some lines is required.

4 Simulation details

In terms of statistical physics our code should simulate an thermodynamic NVE ensemble, but the force calculation and integration introduces artefacts. However they computations can be can be corrected via different approaches. A simple example to yield a constant temperature is to rescale the velocities. Though at first sight this approach looks simple and beautiful, it destroys the character of a thermodynamic ensemble a little bit.

4.1 Task

1. Think about why it is interesting to rescale the velocities, if we want to obtain a constant energy.
2. Try to find out which lines in `mdloop.c` are responsible for the velocity scaling. *Just provide the line numbers.*

5 Study and analysis of the results

After you inserted all the code type `make` in the path of the Makefile to produce an executable and input `./md_nve` in the same directory to perform a simulation. The output files `energy.dat`, `rdf.dat`, `vacf.dat`, and `msd.dat` will be created or **overwritten**. So take care, when performing the simulation study and backup the output after every simulation to an appropriate place. This study consists of four simulations with two different temperatures and two different particle densities. To change the temperature and density you will have to change `readdat.c` and recompile using `make`.

5.1 Homework(3 points)

1. Perform MD simulations with a temperature $T \approx 0.75$ and $T \approx 1.3$ at a density of $\rho \approx 0.8$ and two further according simulations at a density of $\rho \approx 1.2$.
2. Create **one** plot for all 4 simulations that contains all energies from `energy.dat`,(see 1). What is the reason for the change in the total energies in the different setups? *Hint: Think about the different contributions to the total energy.*

A possibility to obtain an idea about the structure of the system is the analysis of the radial distribution function $g(r)$ (RDF) of the particles:

$$g(r) = \frac{1}{\rho 4\pi r^2 dr} \sum_{ij} \langle \delta(r - |r_{ij}|) \rangle \quad (7)$$

The RDF describes by how many particles a particle is surrounded in comparison to the ideal gas phase, where we have $\rho \cdot V$ at every distance r .

5.2 Homework (2 points)

1. Plot all RDFs from `rdf.dat` in one graph like figure 2 and describe the different structures by examining the trend of the different curves. *Hint: Look at the locations and values of the maxima and minima.*

Since you have performed MD simulations, you gathered information about the dynamics of the system. Now you have to analyse the velocity autocorrelation function $C(t)$ (VACF) and the mean square displacement Δr^2 (MSD):

$$C(t) = \langle v(t) \cdot v(0) \rangle \quad (8)$$

$$\Delta r^2 = \langle |\vec{r}(t) - \vec{r}(0)|^2 \rangle \quad (9)$$

The VACF shows the timespan of the correlation between the velocities of a particle. This aspect gets interesting for calculating the MSD, since we have to average by shifting the starting point for the calculation from the beginning towards the end of the simulation. For this we want uncorrelated starting points and the VACF helps us to decide.

5.3 Homework (4 points)

1. Plot the four different MSDs and VACFs from `msd.dat` and `vacf.dat`. (As an example see figures 3, 4).
2. Discuss the change of the first minimum of the VACFs.
3. Explain how the slopes of the MSDs are different for the different simulations.

The deadline for handing in the homework is 25.11.09

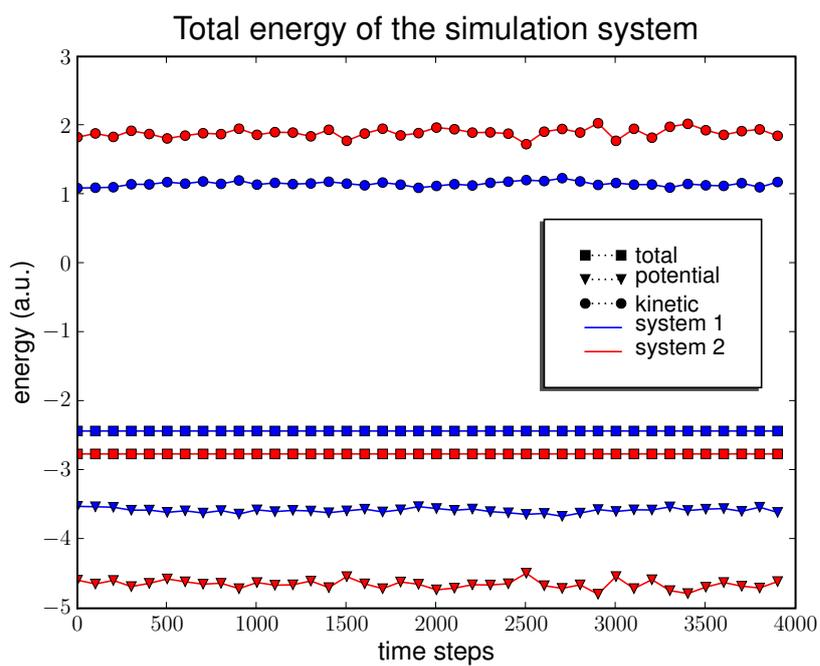


Figure 1: Example for the trend of the total energy.

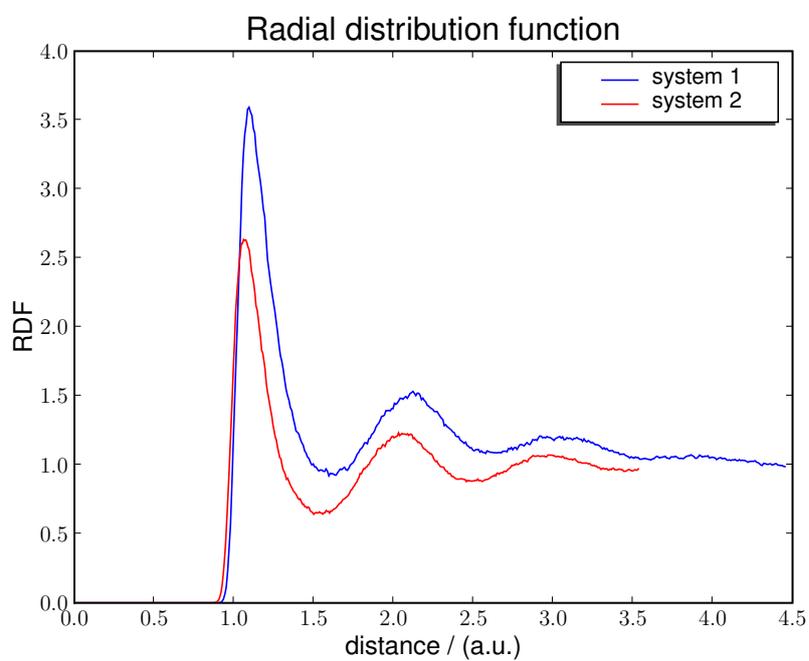


Figure 2: Example for the trend of the radial distribution function.

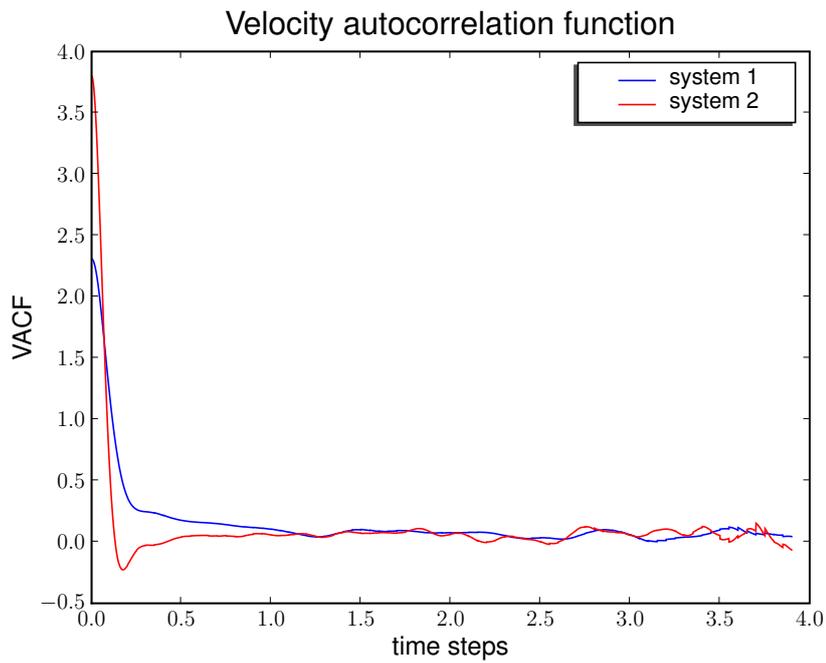


Figure 3: Example for the trend of the velocity autocorrelation function.

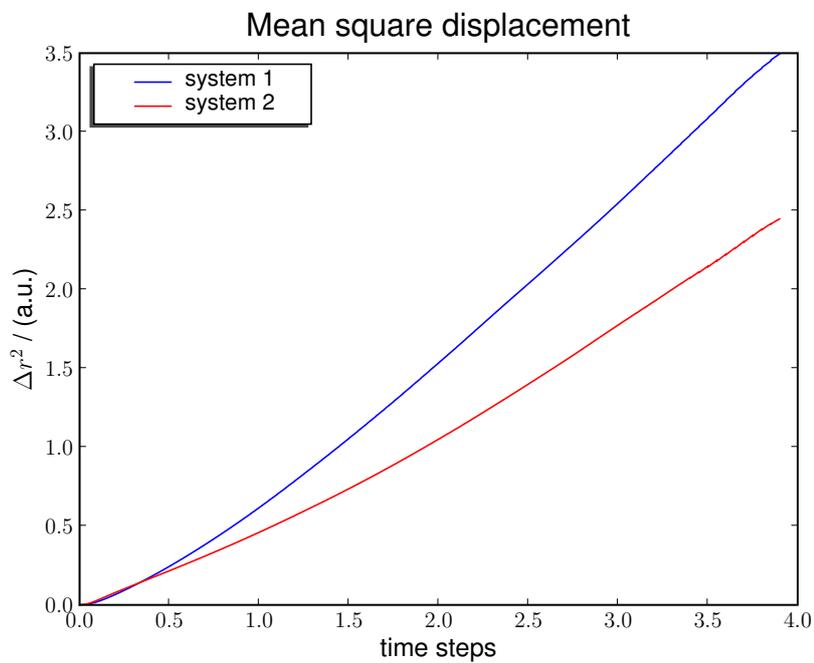


Figure 4: Example for the trend of the mean square displacement.