

Molecular Simulation of Water and Ethanol

Sanne van Kempen 1017389
Jan Moraal 1016866

Abstract

In this report we perform a molecular dynamics (MD) simulation to study properties of water, ethanol and a mixture of 14.3% ethanol in water. Theoretical aspects of molecular dynamics are discussed, including a literature overview of previous works, elaboration on the computation of forces and a discussion on different kinds of ensembles. An important part of the model is the choice of integrator. Based on an analysis of various integrators, the Velocity Verlet scheme is implemented. The simulation parameters and implementation are discussed in detail, and a runtime analysis is performed. The results of the simulation are analyzed in terms of energy levels and radial distribution function (RDF). It is concluded that the simulation performs well in line with the theory, but that the underlying model is too simplistic in comparison with other works on the subject to draw more detailed conclusions.

1 Introduction

In this report we use the molecular simulation to model various particle systems. To create the model we will use molecular dynamics (MD) which is a general and long existing method. It was introduced by Alder and Wainwright, 1959. We consider a system of N particles and denote

- \mathbf{F}_i force experienced by particle i ,
- m_i mass of particle i ,
- \mathbf{q}_i position of particle i ,
- $\mathbf{v}_i = \dot{\mathbf{q}}_i$ velocity of particle i ,
- $\mathbf{a}_i = \dot{\mathbf{v}}_i = \ddot{\mathbf{q}}_i$ acceleration of particle i ,
- $\mathbf{p}_i = m_i \dot{\mathbf{q}}_i$ momentum of particle i .

Starting from some initial $(\mathbf{q}_0, \mathbf{v}_0)$, the state of such a system is completely defined by the collection $(\mathbf{q}_i, \mathbf{v}_i)$, since by Newton's second law we have

$$\dot{\mathbf{q}}_i = \mathbf{v}_i, \quad \dot{\mathbf{v}}_i = \frac{\mathbf{F}_i}{m_i}. \quad (1)$$

The $6N$ dimensional space of all possible $\{\mathbf{q}_i(t), \mathbf{v}_i(t)\}$ is called the **phase space** of the system. MD is based on Hamiltonian mechanics, which uses momentum rather than velocity. The **Hamiltonian** of the system \mathcal{H} is the total energy $E = \mathcal{H}(\mathbf{q}, \mathbf{p})$ acting on the phase space $\{\mathbf{p}_i(t), \mathbf{q}_i(t)\}$. Important properties of a Hamiltonian system are the conservation of energy and the symplectic structure. Symplectic geometry is quite difficult and not the topic of this report, but is needed to be able to integrate over the phase space in a coordinate free fashion. It ensures us that every Hamiltonian \mathcal{H} will produce a phase flow $\phi : \mathbb{R}^{6N} \rightarrow \mathbb{R}^{6N}$, $\phi_t(\mathbf{q}_0, \mathbf{p}_0) \mapsto (\mathbf{q}(t), \mathbf{p}(t))$ that maps from a point at $t = 0$ to any other point for all $t \in \mathbb{R}$. Writing the total energy as sum of potential and kinetic energy, we obtain

$$E = \mathcal{H}(\mathbf{q}, \mathbf{p}) = \sum_{i=1}^N \frac{1}{2m_i} \mathbf{p}_i^2 + \mathbf{v}(\mathbf{q}). \quad (2)$$

Energy conservation implies rest, so the derivative of (2) with respect to time should equal 0. This leads to the **Hamiltonian equations**, which are equivalent to (1),

$$\dot{\mathbf{q}} = \nabla_{\mathbf{p}} \mathcal{H}(\mathbf{q}, \mathbf{p}), \quad \dot{\mathbf{p}} = -\nabla_{\mathbf{q}} \mathcal{H}(\mathbf{q}, \mathbf{p}). \quad (3)$$

An important assumption for the MD method is the **ergodic hypothesis**, stating that on average the time a system is present in some region of the phase space is proportional to the volume of that region in the phase space, i.e. that all possible states are equally likely for the system to be in. Mathematically, for any observable that is a function of $(\mathbf{q}(t), \mathbf{p}(t))$ and any point Γ in the phase space holds

$$\bar{A}_t = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t A(\Gamma(s)) ds = \int A(\Gamma) \rho(\Gamma) d\Gamma = \langle A \rangle, \quad (4)$$

where ρ is a probability density function of the location of the system in the phase space. While assuming that (4) holds, we aim to find a phase flow $\phi_t = (\mathbf{q}(t), \mathbf{p}(t))$ that solves the Hamiltonian equations (3) with initial condition $\mathbf{q}_0, \mathbf{p}_0$.

In practice, MD is used to study properties of substances like diffusion constants, aggregation rates, local densities, particle trajectories or thermodynamic properties. Since the theory is based on particle systems, properties can be analyzed on molecular level. Analyzing systems using classical MD becomes problematic when the number of particles in the system becomes large. In that case, performing simulations using numerical solving methods rather than analytical is a good alternative.

2 Theory

2.1 Physical properties of substances

Physical properties of substances can be studied on a macroscopic level, like density, thermal conductivity, flammability and viscosity, or microscopic level, like the molecular structure and the intra- and intermolecular forces.

A mixture of water and ethanol is interesting to study since both liquids are used extensively in many products and processes. Knowledge of properties of this mixture is applicable in various industries to, for example, determine precisely what concentration mixture should be used in a process to achieve desired properties, or how long diffusion takes before the mixture is homogeneous. Zhang and Yang, 2005 studies mixtures of water and ethanol using MD simulation and compares the results to experimental data. Berryman, 2006 does the same, but also varies pressure and focuses on clustering of ethanol. Kvamme, 1997 compares MD simulation of aqueous mixtures of small alcohols, among which ethanol, to integral equation studies of model systems. Soper, 2013 studies the radial distribution function of pure water using scatter experiments.

The molecular formula of water is H_2O and that of ethanol is $\text{C}_2\text{H}_5\text{OH}$. Next to the strong dipole-dipole attraction, both molecules have an OH-group forming even stronger hydrogen bonds. These properties can be studied using energy diagrams and plotting the radial distribution function (RDF), which is a measure for the density of atoms as function of distance from a reference atom. It is computed as the ratio of density of atoms at distance r divided by the overall density. In the MD simulation we expect to see strong intermolecular forces between water and ethanol due to dipole-dipole attraction and hydrogen bonds. In order to draw conclusions, it is important to simulate systems of pure water and pure ethanol as well.

2.2 Integrators

To simulate a system of particles, we need to transform continuous-time equations of motion into discrete equations. This can be done by using Taylor series approximations (5) and finite differences (6):

$$\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \Delta t \dot{\mathbf{q}}(t) + \frac{(\Delta t)^2}{2} \ddot{\mathbf{q}}(t) + \frac{(\Delta t)^3}{6} \dddot{\mathbf{q}}(t) + \frac{(\Delta t)^4}{24} \mathbf{q}^{(4)}(t) + \dots \quad (5)$$

$$\dot{\mathbf{q}}(t) = \frac{\mathbf{q}(t + \Delta t) - \mathbf{q}(t)}{\Delta t}, \quad (6)$$

both with Newtonian dynamics encoded via $\ddot{\mathbf{q}}(t) = \mathbf{F}_i(\mathbf{q}(t))/m_i$. There exist various algorithms that use variations of (5) and (6) to update the position and velocity in computer simulations, called **integrators**. Desired properties for integrators in general are high order accuracy, numerical stability and preservation of physical properties of the system.

For MD simulation, other aspects of the integrator are also of importance (Holm, 2013). An integrator is said to be **time reversible** if trajectories can be reproduced, i.e. if it is possible to determine $\mathbf{q}(t - \Delta t)$ from $\mathbf{q}(t)$. An integrator is **symplectic** if it preserves the symplectic structure of the phase space. Lastly, **energy conservation** of an integrator is a desired property; an integration step should not change the total amount of energy in the simulated system.

2.2.1 Euler

One of the simplest integrators is the Euler integrator which uses the equations

$$\begin{aligned}\mathbf{q}(t + \Delta t) &= \mathbf{q}(t) + \Delta t \mathbf{v}(t) + \frac{\Delta t^2}{2} \frac{\mathbf{F}(\mathbf{q}(t))}{m}, \\ \mathbf{v}(t + \Delta t) &= \mathbf{v}(t) + \Delta t \frac{\mathbf{F}(\mathbf{q}(t))}{m}.\end{aligned}$$

The Euler method has first-order accuracy, meaning that the total error in each step is proportional to the timestep Δt . This method is not appropriate for MD simulations, since it is not time reversible nor conservative.

2.2.2 Runge-Kutta 4

The Runge-Kutta method is based on the same principle as Euler, but uses intermediate steps in the interval $[t, t + \Delta t]$. The most common implementation is the 4-th order integrator, using a weighted average of the derivatives at t , two estimates of the derivatives at $t + \Delta t/2$ and an estimate of the derivative at $t + \Delta t$ to compute the updates for $t + \Delta t$,

$$\begin{aligned}\mathbf{q}_1 &:= \mathbf{q}(t) + \Delta t \mathbf{v}(t), \quad \mathbf{q}_i = \mathbf{q}(t) + \frac{\Delta t}{2} \left[\mathbf{v}(t) + \frac{\mathbf{v}_{i-1}}{2} \right] \quad (i = 2, 3), \quad \mathbf{q}_4 := \mathbf{q}(t) + \Delta t [\mathbf{v}(t) + \mathbf{v}_3], \quad \mathbf{v}_i := \Delta t \frac{\mathbf{F}(\mathbf{q}_i(t))}{m}, \\ \mathbf{q}(t + \Delta t) &= \mathbf{q}(t) + \Delta t \mathbf{v}(t) + \frac{\Delta t^2}{6} (\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3), \quad \mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{v}_1 + 2\mathbf{v}_2 + 2\mathbf{v}_3 + \mathbf{v}_4}{6}.\end{aligned}$$

2.2.3 Verlet

The Verlet integrator is designed to overcome the deficiencies of the Euler integrator. It is time reversible, symplectic, preserves energy and is much more stable than Euler. The equations are given by

$$\begin{aligned}\mathbf{q}(t + \Delta t) &= 2\mathbf{q}(t) - \mathbf{q}(t - \Delta t) + \frac{\Delta t^2}{2} \frac{\mathbf{F}(\mathbf{q}(t))}{m}, \\ \mathbf{v}(t + \Delta t) &= \frac{1}{2\Delta t} [\mathbf{q}(t + \Delta t) - \mathbf{q}(t - \Delta t)].\end{aligned}$$

Note that the velocity is not taken into account in the position update and the velocity update can only be computed after $\mathbf{q}(t + \Delta t)$ is determined. A disadvantage to this integrator is the need for an initialization step $\mathbf{q}(-\Delta t)$. In practice this is solved by performing one Euler step on $\mathbf{q}(0)$ to obtain $\mathbf{q}(\Delta t)$ and continuing with Verlet with the pair $\mathbf{q}(0), \mathbf{q}(\Delta t)$.

2.2.4 Velocity Verlet

The Velocity Verlet integrator extends the Verlet integrator in the sense that the velocity update is determined explicitly,

$$\begin{aligned}\mathbf{q}(t + \Delta t) &= \mathbf{q}(t) + \Delta t \mathbf{v}(t) + \frac{\Delta t^2}{2} \frac{\mathbf{F}(\mathbf{q}(t))}{m}, \\ \mathbf{v}(t + \Delta t) &= \mathbf{v}(t) + \frac{\Delta t}{2} \left[\frac{\mathbf{F}(\mathbf{q}(t + \Delta t))}{m} + \frac{\mathbf{F}(\mathbf{q}(t))}{m} \right].\end{aligned}$$

The position and velocity updates are calculated at the same time while retaining numerical problems of the Δt^2 term. Thus, when implementing the integrator, the timestep and its unit must be carefully chosen such that the integration scheme is numerically stable. Like Verlet, this integrator is time reversible, symplectic and preserves energy.

2.3 Forces

In order to apply any integrator to the system of particles, we need to be able to compute the force \mathbf{F}_i that acts on particle i . For simplicity, we will consider the intramolecular forces that result from bond, angle and proper dihedral¹ potentials and the intermolecular forces that result from Lennard-Jones (LJ) potentials. More extensive MD simulators may also include improper torsion, Van der Waals, electrostatic or other types of forces. The total force on a particle is the sum of forces resulting from bond, angle, dihedral and Lennard-Jones potentials, so $\mathbf{F}_i = \mathbf{F}_{i,\text{bond}} + \mathbf{F}_{i,\text{angle}} + \mathbf{F}_{i,\text{dih}} + \mathbf{F}_{i,\text{LJ}}$.

2.3.1 Intramolecular forces

Bond and angular forces are harmonic, implying that these potentials can each be approximated by a quadratic term with a unique global minimum where the potential is 0. Dihedral forces are also harmonic, however we choose to model these using a trigonometric periodic function. An overview of the different potentials and forces is given in Table 1.

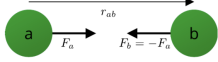
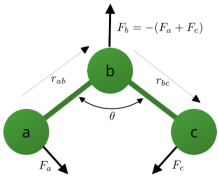
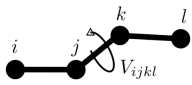
Type	Potential	Force
	$V(r) = \frac{1}{2}k_b(r - r_0)^2$	$\mathbf{F}_a = -\frac{dV(r)}{dr} \frac{\mathbf{r}_{ab}}{\ \mathbf{r}_{ab}\ }$ $\mathbf{F}_b = -\mathbf{F}_a$
	$V(\theta) = \frac{1}{2}k_\theta(\theta - \theta_0)^2$	$\mathbf{n} = \mathbf{r}_{ab} \times (\mathbf{r}_{ba} \times \mathbf{r}_{bc})$ $\mathbf{m} = -\mathbf{r}_{bc} \times (\mathbf{r}_{ab} \times \mathbf{r}_{bc})$ $\theta = \arccos\left(\frac{\mathbf{r}_{ab} \cdot \mathbf{r}_{bc}}{\ \mathbf{r}_{ab}\ \ \mathbf{r}_{bc}\ }\right)$ $\mathbf{F}_a = -\frac{dV(\theta)}{d\theta} \frac{1}{\ \mathbf{r}_{ab}\ } \frac{\mathbf{n}}{\ \mathbf{n}\ }$ $\mathbf{F}_c = -\frac{dV(\theta)}{d\theta} \frac{1}{\ \mathbf{r}_{bc}\ } \frac{\mathbf{m}}{\ \mathbf{m}\ }$ $\mathbf{F}_b = -\mathbf{F}_a - \mathbf{F}_c$
	$V(\psi) = \frac{1}{2}C_1(1 + \cos(\psi)) + \frac{1}{2}C_2(1 - \cos(2\psi)) + \frac{1}{2}C_3(1 + \cos(3\psi)) + \frac{1}{2}C_4(1 - \cos(4\psi))$	$\mathbf{n} = \mathbf{r}_{ij} \times \mathbf{r}_{jk}$ $\mathbf{m} = -\mathbf{r}_{jk} \times \mathbf{r}_{kl}$ $\theta = \arccos\left(\frac{\mathbf{n} \cdot \mathbf{m}}{\ \mathbf{n}\ \ \mathbf{m}\ }\right)$ $\theta_1 = \arccos\left(\frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{jk}}{\ \mathbf{r}_{ij}\ \ \mathbf{r}_{jk}\ }\right)$ $\theta_2 = \arccos\left(\frac{\mathbf{r}_{jk} \cdot \mathbf{r}_{kl}}{\ \mathbf{r}_{jk}\ \ \mathbf{r}_{kl}\ }\right)$ $\mathbf{F}_i = -\frac{\partial V(\theta)}{\partial \theta} \frac{1}{\ \mathbf{r}_{ij}\ \sin(\theta_1)} \frac{\mathbf{n}}{\ \mathbf{n}\ }$ $\mathbf{F}_j = -\mathbf{F}_i - \mathbf{F}_k - \mathbf{F}_l$ $\mathbf{F}_k = (-\mathbf{r}_{ok} \times \mathbf{F}_l + \frac{1}{2}\mathbf{r}_{ij} \times \mathbf{F}_i - \frac{1}{2}\mathbf{r}_{kl} \times \mathbf{F}_l) \times \frac{\mathbf{r}_{ok}}{\ \mathbf{r}_{ok}\ ^2}$ $\mathbf{F}_l = -\frac{\partial V(\theta)}{\partial \theta} \frac{1}{\ \mathbf{r}_{kl}\ \sin(\theta_2)} \frac{\mathbf{m}}{\ \mathbf{m}\ }$

Table 1: Intramolecular potentials and forces used in the simulation. Images from 2MMN40 course page.

2.3.2 Intermolecular forces

The Lennard-Jones potential between two atoms i, j , which ought to be not part of the same molecule, is given by

$$U^{\text{LJ}}(r) = 4\epsilon \cdot \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right], \quad (7)$$

where σ and ϵ are parameters that depend on the atom type. Figure 1 shows this potential as function of the distance r between the atoms. At $r = \sigma$, equation (7) equals zero so the graph intersects the x -axis. For $r < \sigma$, the fraction σ/r is greater than 1 so the twelfth

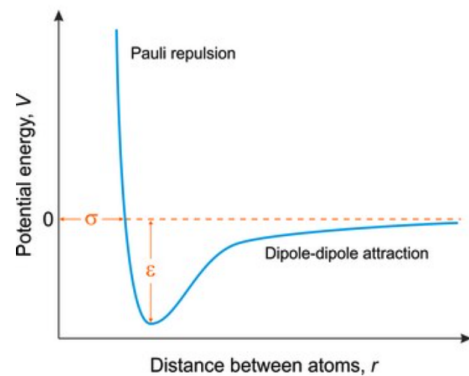


Figure 1: Lennard-Jones potential. Taken from [Generalic's Chemistry Glossary](#).

¹Often called torsion force/potential in literature.

power dominates the sixth power, therefore the potential is positive and grows to ∞ as $r \rightarrow 0$. On the other hand if $r > \sigma$ then $\sigma/r < 1$ so the sixth power dominates and the potential has negative value. For $r \rightarrow \infty$ we have $\sigma/r \rightarrow 0$, so also $U \rightarrow 0$.

The potential has a global minimum r_{min} , implying that for $r < r_{min}$ the force are repulsive, while for $r > r_{min}$ the force is attractive. By setting $\frac{dU}{dr} = 0$ we find that this minimum is attained at $r_{min} = \sqrt[6]{2}\sigma$ with value $U(r_{min}) = -\varepsilon$, as shown in the Figure 1.

The potential as described in (7) acts on atoms of the same type. In our simulation however, we need to determine these forces between atoms of different types as well. Therefore we introduce *mixing rules* for the values of σ and ε , given by the following equations for atoms i and j , known as the Lorentz – Berthelot rules,

$$\varepsilon_{ij} = \sqrt{\varepsilon_i \varepsilon_j}, \quad \sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j).$$

Note that these mixing rules do not have any effect if the two atoms are of the same kind. The mixed σ_{ij} is defined as the arithmetic mean of σ_i and σ_j . This makes sense since we have seen that this parameter indicates the distance at which the potential equals zero. Mathematically, interpolating between two points on the x -axis leads to the average of those points. Physically, interpolating between the two distances σ_i and σ_j gives a good estimate for the distance at which the potential between different types of atoms will be zero. The mixed ε_{ij} is defined as the geometric mean since this parameter scales the global minimum in the graph and can therefore be thought of as exponent. A disadvantage of these mixing rules is that if either $\varepsilon_i = 0$ or $\varepsilon_j = 0$, the potential between atoms i and j will be zero everywhere. Other types of mixing rules exist, like Lorentz rules (taking both arithmetic mean), Berthelot rules (taking both geometric means) or Kong rules including sixth and third powers of σ , as described by Delhommelle and Millié, 2001. The greater the differences $|\sigma_i - \sigma_j|$ and $|\varepsilon_i - \varepsilon_j|$ for different types of atoms in the system are, the greater the influence of the mixing rules on the simulation.

2.4 Thermostat

In a practical setting, it is often not possible to ensure the energy of a system is constant. What can be done however, is keeping the temperature of a substance fixed. To accurately represent this, we consider the canonical NVT ensemble (indicating number of particles, volume and temperature) instead of the microcanonical NVE ensemble (indicating number of particles, energy and temperature). In our simulation, we implement this by means of a thermostat, keeping the system temperature constant throughout the simulation. We use a velocity rescaling thermostat, which combines well with the Velocity Verlet integration scheme that we use, as explained by Mudi and Chakravarty, 2004. This means that at every timestep, all particle velocities are scaled by the same factor in order to obtain the desired system temperature:

$$\mathbf{v} \mapsto \sqrt{\frac{T}{T_0}} \cdot \mathbf{v},$$

where T is the desired temperature and T_0 is the current system temperature. This is calculated from the velocities of all particles using the equipartition theorem:

$$E_{kin} = T_0 \cdot N_f \cdot k_B \quad \Longleftrightarrow \quad T_0 = \frac{E_{kin}}{N_f \cdot k_B}, \quad E_{kin} = \sum_{i=1}^N \frac{1}{2} m_i |\mathbf{v}_i|^2$$

where N_f is the number of degrees of translational freedom and k_B is the Boltzmann constant. Since particles move through three-dimensional space, each particle contributes three degrees of freedom; thus, $N_f = 3 \cdot N$.

Note that using a thermostat also affects the total energy level in a system. In a closed system, energy is conserved. In other words, the sum of potential and kinetic energy always remains constant. A thermostat, however, keeps the temperature fixed by adding or subtracting a bit of kinetic energy (e.g. at every timestep in a simulation), which causes slight fluctuations in the total energy. There are several drawbacks to this thermostat. First of all, rescaling velocities at regular time-intervals causes discontinuities in the phase-space as the velocity of a particle is no longer continuous with respect to time. In particular, the equipartition theorem is violated, and the simulation does not sample the canonical, microcanonical nor isokinetic ensemble anymore. The system also is not time-reversible anymore.

Secondly, if an initial configuration of molecules is unnatural in such a way that some forces are much larger and others are much smaller than they would be in a natural setting, rescaling all velocities by the same factor results in ‘wrong’ rescalings; the velocity of too fast particles might be increased, or the velocity other particles might be decreased unreasonably. In practical terms, this means it takes a much longer time for the system to converge.

However, the consequences of these issues are limited, especially for well-behaved systems (i.e. systems that show convergence of energy levels). Morishita (2000) argues that the properties of the canonical ensemble is preserved approximately. Moreover, the simple velocity rescaling thermostat is among the most-used thermostats in MD simulations and is efficient to use; hence, we decide to use it despite the drawbacks. (Braun et al., 2018)

3 Simulation

3.1 Simulation parameters

3.1.1 Timestep

In order to find an appropriate timestep for integration, we must determine which inner molecular oscillation has the largest frequency, or equivalently the smallest period. These oscillations are harmonic, meaning that a displacement of Δx of a particle in the system leads to a restoring force \mathbf{F} acting on that particle which is proportional to Δx . In a more mathematical fashion, this implies that we can write the force \mathbf{F} as $\mathbf{F} = -k\Delta x$ for some constant k . Note the minus sign, since a displacement of a particle in *positive* direction will cause a *negative* force on that particle (Newton’s third law). By Newton’s second law, we have that the force is also equal to mass times acceleration, the second derivative of x w.r.t. t . If a mass is displaced x units at time t then

$$\mathbf{F} = ma = m \frac{d^2x(t)}{dt^2} = -kx(t) \iff \frac{d^2x(t)}{dt^2} + \frac{k}{m}x(t) = 0.$$

This equation forms a differential equation which has as general solution $x(t) = A \cos(\omega t + \phi)$ with $\omega = \sqrt{\frac{k}{m}}$. This implies that the oscillation is indeed periodic with constant amplitude A , shift ϕ and period $T = 2\pi/\omega$. Thus we can use the following equation to determine the periods of oscillations in our molecules,

$$T = 2\pi \sqrt{\frac{m}{k}}. \quad (8)$$

Formula (8) is used for single-body systems containing one mass. However, a bond force acting on two connected atoms clearly is a two-body system. We can transform this into a one-body system by defining the *reduced* mass of two atoms a and b as $\mu = \frac{m_a m_b}{m_a + m_b}$.

We aim to express the oscillation period T in the unit ps. We are given the force constant for bonds in the unit kJ/(mol nm²) and the masses in amu, so we must rewrite μ/k_b in the unit ps². Note that mol · amu = g and kJ = $\frac{10^3 \text{ kg m}^2}{\text{s}^2}$, so

$$\left[\frac{\mu}{k_b} \right] = \frac{\text{amu mol nm}^2}{\text{kJ}} = \frac{\text{g nm}^2 \text{s}^2}{10^3 \text{ kg m}^2} = \frac{\text{g} \cdot 10^{-18} \text{ m}^2 \cdot 10^{24} \text{ ps}^2}{10^6 \text{ g m}^2} = \text{ps}^2.$$

Now using the parameters as stated in Appendix A we can now easily determine the period of the oscillations due to bond potentials in the water and ethanol molecule, given in Table 2.

	H ₂ O		C ₂ O ₅ OH		
bond	12, 13	15	31, 41, 21, 65, 75	58	98
μ	0.948258	6.0055	0.929955	6.86062	0.948258
k	502416	224262.4	284512.0	267776.0	462750.0
T	0.008632	0.0325144	0.0113595	0.0318036	0.00899435

Table 2: Oscillation periods in ps in a water and ethanol molecule.

We see that the bond force in the water molecule has the smallest oscillation period, closely followed by the O-H bond in ethanol. This means that in our simulation we should use a timestep certainly smaller than $0.0086 \text{ ps} = 8.6 \text{ fs}$.

The respectively three- and four-body systems for angular and dihedral forces are not easily reduced to a one-body system. The periods of these oscillations can be determined for instance using IR spectral analysis (Wang et al., 2017). However, from the provided force constants we observe that all angle and dihedral force constants are significantly smaller than the bond force constants. Via relation (8) we see that a decrease in k leads to an increase in T , so the springs become less 'stiff' and the system oscillates slower. Therefore a timestep sufficiently small for bond forces should certainly suffice for angle and dihedral forces as well. This suspicion is confirmed by both numerical and experimental studies (Saiz et al., 1997).

3.1.2 Lennard-Jones cut-off parameter

As discussed in Section 2.3, the Lennard-Jones force is an intermolecular force acting on atoms from different molecules. This implies that in a system of N particles, there are $O(N^2)$ Lennard-Jones forces to be determined. The graph of the Lennard-Jones potential in Figure 1 shows that the potential converges to 0 as the distance between atoms r increases, also the slope of the graph becomes more flatten as r increases. Next to that the term $1/r$ in (7) converges to 0. This means that atoms at a great distance of each other have negligible LJ force.

Since the number of calculations for all LJ forces is the most demanding in terms of computation time, it is common to introduce a cut-off distance in the simulation. LJ interactions between atoms at a distance beyond r_{cut} are ignored. According to Toxvaerd and Dyre, 2011, this value is usually chosen as $r_{cut} = 2.5\sigma$. At this value, the LJ potential is merely 1.6% of its minimum. Since we have several values for σ in our simulation, depending on the atoms, we choose $r_{cut} = 2.5 \max(\sigma)$.

3.1.3 Box size and number & placement of molecules

We study three systems: one with pure water, one with pure ethanol and one with a mixture of water and 14.3 % mass ethanol. For all three, pressure is atmospheric and temperature is kept at 298.15 K.

The approximate periodic boundary condition box size to be studied is $3 \times 3 \times 3 \text{ nm}$. However, we simulated each of the three systems with slightly different box sizes, to be able to construct a well-distributed initial placement of molecules.

Box sizes were chosen such that the number of molecules placed in a uniform grid inside the box is a power of 3. The box size then followed from the densities of water, ethanol and the mixture, taken from Washburn, 2003.

Ethanol molecules have to be rotated first in such a way that the Lennard-Jones potential of the initial setting is not too high. For the mixture, we choose to place ethanol molecules at (more or less) regular distances from one another, because otherwise they would be too close together and convergence of the system to a natural state would take too long. In this way, the initial position of molecules may not have been natural. But according to Martínez and Martínez, 2003, for simple liquids and mixtures such as ours, a regular lattice placement may be used.

3.1.4 Units & Parameters

Taking all the above into account, we arrive at the following settings for our simulation.

	Water	Ethanol	Mixture
Timestep	0.004 ps (or 4 fs)	0.003 ps (or 3 fs)	0.002 ps (or 2 fs)
Duration	600 ps (or 0.6 ns)	100 ps (or 0.1 ns)	1000 ps (or 1 ns)
Boxsize (L)	31.08 Å	32.22 Å	32.29 Å
Cutoff (r_{cut})	7.876525 Å	8.75 Å	8.75 Å
Number of molecules (N)	1000 water	343 ethanol	939 water, 61 ethanol

Table 3: Simulation settings.

Here the units are chosen such that the simulation does not yield numerical inaccuracies. In order to achieve this, we arrive at the units in Table 4.

Symbol	Quantity	Unit	SI equivalent
x	position	Å	10^{-10} m
r	distance	Å	10^{-10} m
θ	angle	rad	rad
m	mass	amu	$1.66054 \cdot 10^{-27}$ kg
t	time	ps	10^{-12} s

Table 4: Overview of symbols, quantities and their units.

3.2 Implementation

The MD simulation is built in Python 3.8 without any packages other than `numpy` and `matplotlib`. It consists of four code files. The first one sets the initial configuration of molecules in a box of given size, including all necessary force constants and topology information. The next file executes the simulation itself, and writes simulation results to two files. These are then each read by a script analyzing the results; one for the radial distribution function, and one for the analysis of energy levels. As the simulation file is the most complex and the most relevant code for this study, in the following sections we will only discuss this file.

3.2.1 Simulation flow chart

A flow chart of our simulation code is shown in Figure 2. One loop in the gray box indicates one timestep. The simulation stops when the time has reached the predetermined end time. The updating of x, v, a depends on the used integrator. The integrator scheme for Velocity Verlet is shown on the right.

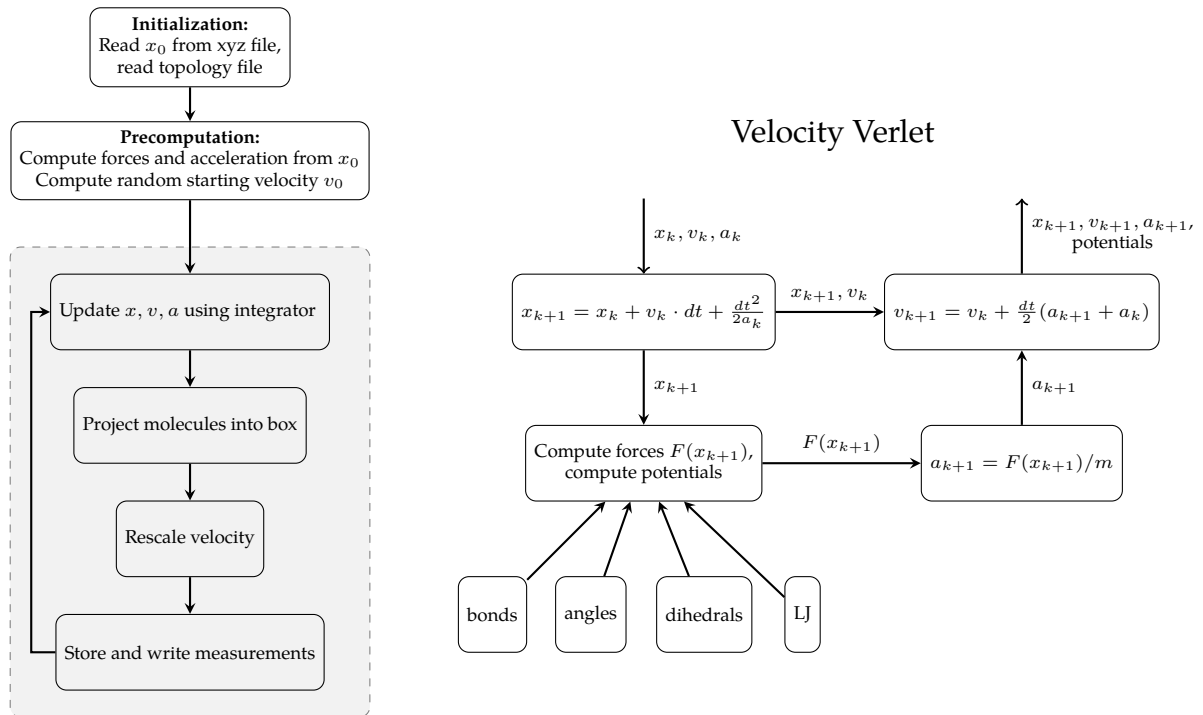


Figure 2: Flow chart of simulation.

3.2.2 Runtime analysis

For MD simulations, performance and scalability play an important role as calculations can quickly become complex and simulated systems are often large. For this reason, our aim is to make efficient use of the possibilities that especially the `numpy` package offers to make computations as efficient as possible.

One of the great benefits of using `numpy` is that many large operations can be sped up significantly, not rarely by an order of 100 (Van Der Walt et al., 2011). For example, often 'for' and 'while' loops, which are slow in Python, can be avoided using proper indexing or `numpy` functions. In our code this

replacement could be made successfully for almost all computations. This reduces hidden constants in the analysis of the order of the runtime below.

The parts contributing most to the runtime of our simulation are the computation of Lennard-Jones forces and determining the distances between all atoms. There may be other slow parts in the initialisation, but as these are only executed once, they barely influence the runtime, whereas distance and Lennard-Jones force computations are updated every timestep.

The distance calculation is the only operation involving *all* pairs of atoms in the system, which means that the number of operations scales quadratically with the number of particles. This cannot be avoided, as the distance is essential in the computation of Lennard-Jones forces, and a cutoff can only be used once all distances are computed.

The reason for the slowness of the computation of Lennard-Jones forces is less apparent. A cutoff was introduced, which in all three simulations reduced the number of calculations of Lennard-Jones forces by about 90%. Yet, the computation time was only reduced by about 50%. We could not find an explanation for this during this project, so improving the efficiency of Lennard-Jones force computations would be a suggestion for further improvement. Still, introducing the cutoff is essential for runtime reduction; it reduces the computation of forces on all pairs ($O(N^2)$) to a number of pairs in a ball of fixed radius around each particle. Using the particle density of the mixture we simulate, the average number of particles present in such a ball, say N_B can be calculated. Computing Lennard-Jones forces then happens in linear time, scaling with factor N_B .

The slow distance and Lennard-Jones force computations are the reason for us to only consider a $3 \times 3 \times 3$ nm box instead of $5 \times 5 \times 5$ nm, as is suggested in the project description. This reduces the number of atoms by a factor of $(\frac{3}{5})^3 = 0.216$. Together with the quadratic scaling of distance computations, this means that using the smaller box, the number of computations is at around $0.216^2 \cdot 100\% \approx 4.7\%$ of the number of computations in the larger one.

All other calculations either scale linearly with the number of particles (e.g. velocity rescaling, molecule projection), are only executed once per simulation (reading XYZ or topology files, computing pairwise Lennard-Jones parameters) or only act on atoms in the same molecule (bond, angular and dihedral forces). The latter may seem like an operation scaling quadratically or worse, but for each type of molecule, the number of calculations is fixed by the number of subsets of atoms of that molecule on which the intramolecular forces act (i.e. the number of lines of the topology file, simply put). This means that computation of intramolecular forces also scales linearly with the number of atoms, albeit with a large constant.

For a system with N particles and number of simulation steps k , the runtime of our simulation is summarized in Table 5

Computation	Input	Precomputation	Distance	Intramolecular	Lennard-Jones	Rescaling
Time order	$O(N)$	$O(N^2)$	$O(k \cdot N^2)$	$O(k \cdot N)^2$	$O(k \cdot N)$	$O(k \cdot N)$

Table 5: Runtime order analysis for number of particles N and number of timesteps k

Taking only the highest runtime order into account, this means that our simulation runs in $O(k \cdot N^2)$ time. Alternatively, if the particle density ρ of a system is given, we could express the runtime as a function of the box size. In a cube with dimension $L \times L \times L$, we have $N = \rho \cdot L^3$, so the runtime becomes $O(k \cdot L^6)$.

3.2.3 Strengths, weaknesses and improvements

An important strength of our simulator is its flexibility. The code is designed in such a way that many parts could easily be replaced by other computations, without affecting the rest of the computations; for example, different shapes of boxes for the periodic boundary conditions only requires the function `DISTATOMSPBC` to be adapted, while the rest of the code can remain the same. In the same way, the integrator, representation of the different potentials and the thermostat³ can be adapted.

Another strength was already mentioned in Section 3.2.2; the greater part of our simulation is computationally efficient. There certainly is room for improvement, but already the runtime is reduced greatly compared to other implementations.

³Possibly depending on the complexity of the thermostat.

Still, there are ample weaknesses, and with that opportunities for improvement. Most importantly, as also mentioned in Section 3.2.2, the simulation scales badly with the number of particles, especially concerning the calculation of distances and Lennard-Jones forces. The latter could possibly be sped up by introducing an atom-dependent cutoff. This would introduce additional operations, but could save a significant amount of computation for atoms that have small or zero Lennard-Jones forces.

Besides the runtime, memory usage could be reduced as well. For example, whether or not two atoms belong to the same molecule is stored in a matrix, even though it is sparse. Instead, this could be stored much more efficiently in a list, similar to the adjacency list of a graph. Because this does not significantly influence the performance of our system, we decided to spend our time improving other parts, but still some efficiency could be gained.

The flexibility of our code could be improved as well. Now, for the sake of efficiency, forces are computed within the integrator. Also, since force and potential energy calculations are similar, potentials are included in the function computing forces, so these are also returned within the integrator. For this simulation this is not an issue, but re-using the integrators in a different setting would require additional work.

Finally, our programming approach is a mix of functional and procedural, which might be a drawback if others were to use our code. Changing to a fully functional approach could improve readability and overview, especially in combination with the option to read settings from XML files instead of including them in the simulation code.

4 Results

4.1 Integrators

In order to analyze the different integrators Euler, Verlet, Velocity Verlet and RK4, we simulate a single hydrogen molecule. The bond force constant for this molecule is $k = 245.31 \text{ kJ}/(\text{mol nm}^2)$ and the equilibrium bond length is $r_0 = 0.74 \text{ \AA}$ (Haken and Wolf, 2013). Using the calculations as described in Section 3.1.1 we find that the oscillation period of hydrogen is $T = 0.0285 \text{ ps}$. In the simulation we have two atoms with starting position $\mathbf{q}_1 = (0, 0, 0)$, $\mathbf{q}_2(0, 0, 1)$, $\mathbf{v}_1 = \mathbf{v}_2 = (0, 0, 0)$ so that the forces and potentials are one dimensional (there is only movement in the z -direction). The total energy in the system can be split into kinetic and potential energy which can be expressed as

$$E = E_{kin} + E_{pot} = \frac{1}{2}\mu\mathbf{v}^2 + \frac{1}{2}kA^2,$$

where $\mu = 0.504 \text{ amu}$ is the reduced mass, $A = 1 \text{ \AA}$ is the amplitude of the oscillation and $\mathbf{v} = \mathbf{v}_1 - \mathbf{v}_2$. For all four integrators we plot energy diagrams and phase-space diagrams which are shown in Appendix C.

The energy diagram of Euler shows the instability of this integrator: both the kinetic and potential energy increase over time, causing the particles to move further apart at each oscillation, so the system is not closed. This instability is confirmed by the phase space diagram, the points $(r(t), p(t))$ do not converge but spiral outwards, meaning that no solution will be found for the differential equation (1). Using Euler as integrator would cause the system to ‘explode’.

The RK4 integrator also shows instability, but note that the timestep used in this simulation is $0.5T$. For a small timestep like $0.1T$, the integrator does converge. Again the energy is increasing over time and the phase space plot does not converge to a set of solutions. The ‘spiked’ line of piecewise smooth functions arises from the linear combination of intermediate steps used in the computations of the integrator.

The Verlet integrator shows very different diagrams than the two previous integrators. Since the integrator is conservative, the total energy remains constant and the phase space diagram shows convergence. However, the energy in the system is extremely high and increases further when the timestep Δt decreases. This is caused by the term $\frac{1}{2\Delta t}$ in the computation of the velocity. A small timestep means high velocity which in turn means high kinetic energy. Indeed, in the phase space diagram we see that the particles are not moving very far apart (low potential energy) but achieve great speed in their motion (high kinetic energy).

The energy diagram of the Velocity Verlet integrator shows a recognizable pattern of a mass-spring system in which $E = E_{pot}$, $E_{kin} = 0$ at times when the atoms are at distance A from each other, and $E = E_{kin}$, $E_{pot} = 0$ at times when the atoms are exactly at distance $r_0 = 0.74 \text{ \AA}$ from each other. The

phase space diagram shows immediate convergence. We can even check the known properties of a two-body mass-spring system in this diagram, and indeed find that the middle of the ellipse is $(r_0, 0)$, the horizontal radius of the ellipse is $A = 1 \text{ \AA}$ and the vertical diameter is $\sqrt{k/\mu} = \sqrt{245.31/0.504} \approx 22 \text{ amu \AA/ps}$.

In the simulation, computing the forces on all particles is the most computationally demanding task, therefore we want to use an integrator which works accurately not only for small timesteps. We want to use an integrator that preserves the physical laws and properties which hold in Hamiltonian systems, so the Velocity Verlet is a suitable choice.

4.2 Energy

In order to better understand the behavior and properties of our simulated systems, we can first study the different energy levels. These give valuable insight in the relative importance of different kinds of energy, as well as possible energy equilibria. A first distinction can be made between kinetic and potential energy, which together make up the total energy in a system. These energy levels are shown for each of our three substances in Figure 3.

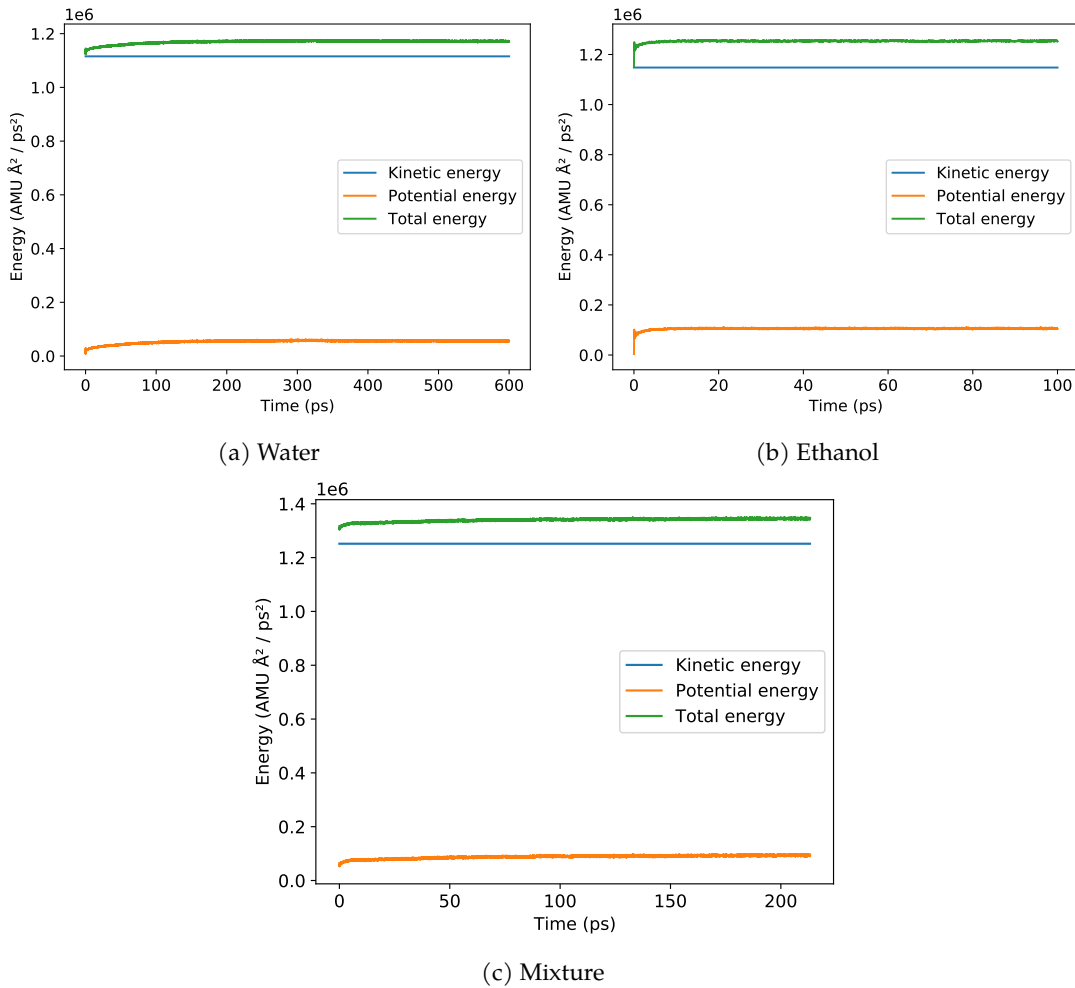


Figure 3: Kinetic, potential and total energy plotted against time

In general, the three systems seem similar. However, there are two main differences. First, the total energy level in the mixture is higher than in both the system with pure water and that with pure ethanol. This is not only a result from the difference in box size (see Table 3); after compensating for this difference⁴, the total energy level is highest in the mixture and lowest in ethanol. On the other hand,

⁴By a factor of $(\frac{L_1}{L_2})^3$ for box sizes L_1 and L_2

the average amount of total energy per particle is almost equal for ethanol and the mixture at 405 AMU $\text{\AA}^2/\text{ps}^2$, and slightly lower in water with 396 AMU $\text{\AA}^2/\text{ps}^2$.

Second, in the system of pure ethanol, there are consistently higher levels of potential energy than in a system of pure water; in ethanol, potential energy makes up for around 8.4% of the total energy, in water this is 6.2%. This could be explained by the fact that ethanol molecules are larger, which means they are subject to more, possibly conflicting intramolecular forces (relative to the number of particles), which in turn yields higher potential energy.

Kinetic energy is straightforward as it is fully determined by the masses and velocities of all particles. Potential energy however is more complex; hence, we study it separately in the next section.

4.2.1 Potential energy

The potential energy in the systems we simulate correspond to the forces we compute: within molecules there may be bond, angle and (in case of ethanol) dihedral potentials, and between molecules acts the Lennard-Jones potential. The levels of these different energies are plotted in Figure 4.

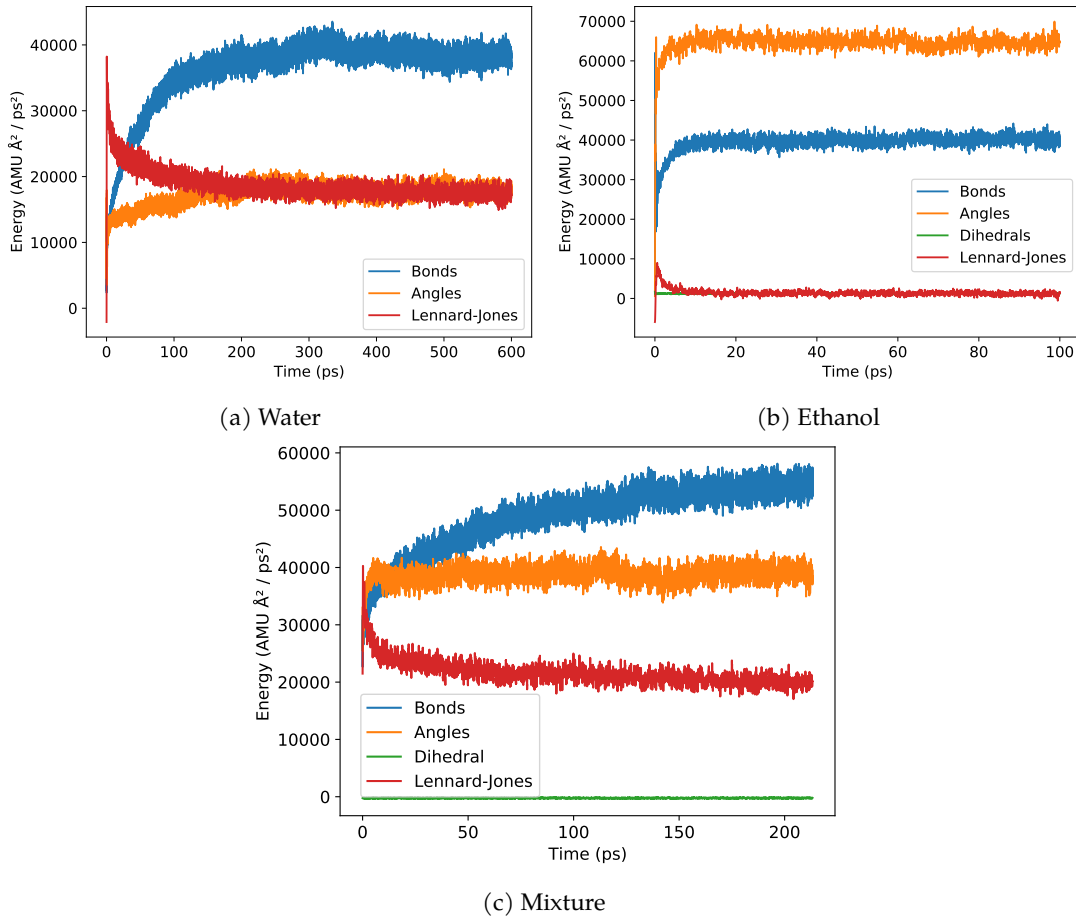


Figure 4: Different types of potential energy plotted against time

Immediately we see that there are far greater differences between the systems than the total energy plots in Figure 3 would suggest.

First of all, Figures 4a and 4c show that it takes the water and mixture systems much longer to converge. As the ethanol system does not show this behavior in Figure 4b, this could indicate that the initial placement of water molecules is unnatural, either with respect to each other (relative positions of molecules) or internally (wrong shape). As described in Section 2.4, this unnatural placement combined with the use of a simple velocity rescaling thermostat could explain the slow convergence of the system.

Next, we see a great difference between the system with pure water and that with pure ethanol. In the former, the bond potential is the highest, and angular and Lennard-Jones potential are at about the

same level. In contrast, the latter has almost no energy from the Lennard-Jones potential, and angular potential is the most important.

In the mixture, as one might expect, we see features of both systems. Bond potential contributes most to the total potential energy, which is not strange as it is highest in pure water and also significant in pure ethanol. The high levels of angular potential in ethanol yield a higher level of angular potential the mixture than in pure water. In particular, angular potential energy levels are now significantly higher than Lennard-Jones potential levels.

Remarkably, in both pure ethanol and the mixture, dihedral potential is negligible compared to the other types of potential energy. This is not surprising, as the force constants used in our simulation are much lower for dihedrals than for bonds or angles. This is also in line with other studies; for example, Zhang and Yang, 2005 assumes the angle between the $\text{CH}_3\text{-CH}_2\text{-O}$ and $\text{CH}_2\text{-O-H}$ planes to be constant, so that the torsional energy for this dihedral angle is zero. Whether or not these low levels of dihedral potential are accurate should be subject to further research.

4.3 Temperature

For all three simulations, temperature is kept constant at $T = 298.15$ K, which seems to make the system temperature not interesting to analyze. The system temperature was however also measured after the integration step but before rescaling velocities to obtain the desired temperature. For all three systems, this temperature is on average equal to the desired value of 298.15 K. The only (slight) difference between the simulations is that the system of pure water has a slightly higher deviation from this mean, with a standard deviation of $\sigma \approx 0.11$ K as opposed to $\sigma \approx 0.07$ K for pure ethanol and the mixture. This indicates that the thermostat influences this system more than the other two, as more energy is added or subtracted in order to keep the temperature constant.

4.4 Radial distribution function

The radial distribution function is computed in the following way. Denote N_k the number of atoms in the shell of distance $[r, r + \delta]$ from one reference atom. The volume of such a shell is

$$V_r = \frac{4}{3}\pi((r + \delta)^3 - r^3) \text{ \AA}^3.$$

The average density of atoms ρ given by the total number of molecules divided by the volume of the box L^3 . To obtain the RDF, we need to normalize N_r and choose δ small, so

$$g(r) = \lim_{\delta \rightarrow 0} \frac{N_r}{\rho V_r}$$

Due to the periodic boundary conditions, we compute $\text{mod } \frac{L}{2}$, which means two atoms cannot be further apart than $\frac{L}{2}$ in one dimension. Therefore in two dimensions, the maximal distance is equal to the diagonal of right angled triangle with two equal sides of length $\frac{L}{2}$, which equals $\frac{L}{\sqrt{2}}$. In three dimensions, the maximal distance is equal to the diagonal of right angled triangle with sides of length $\frac{L}{\sqrt{2}}$ and $\frac{L}{2}$, thus

$$r_{max} = \sqrt{\left(\frac{L}{\sqrt{2}}\right)^2 + \left(\frac{L}{2}\right)^2} = \sqrt{\frac{L^2}{2} + \frac{L^2}{4}} = \frac{\sqrt{3}L}{2}.$$

The initial position of our simulation is very artificial and not representative for a real physical system. Therefore we consider the average of the RDF over the *last* 300 timesteps of our simulation. The results are shown in Figure 5.

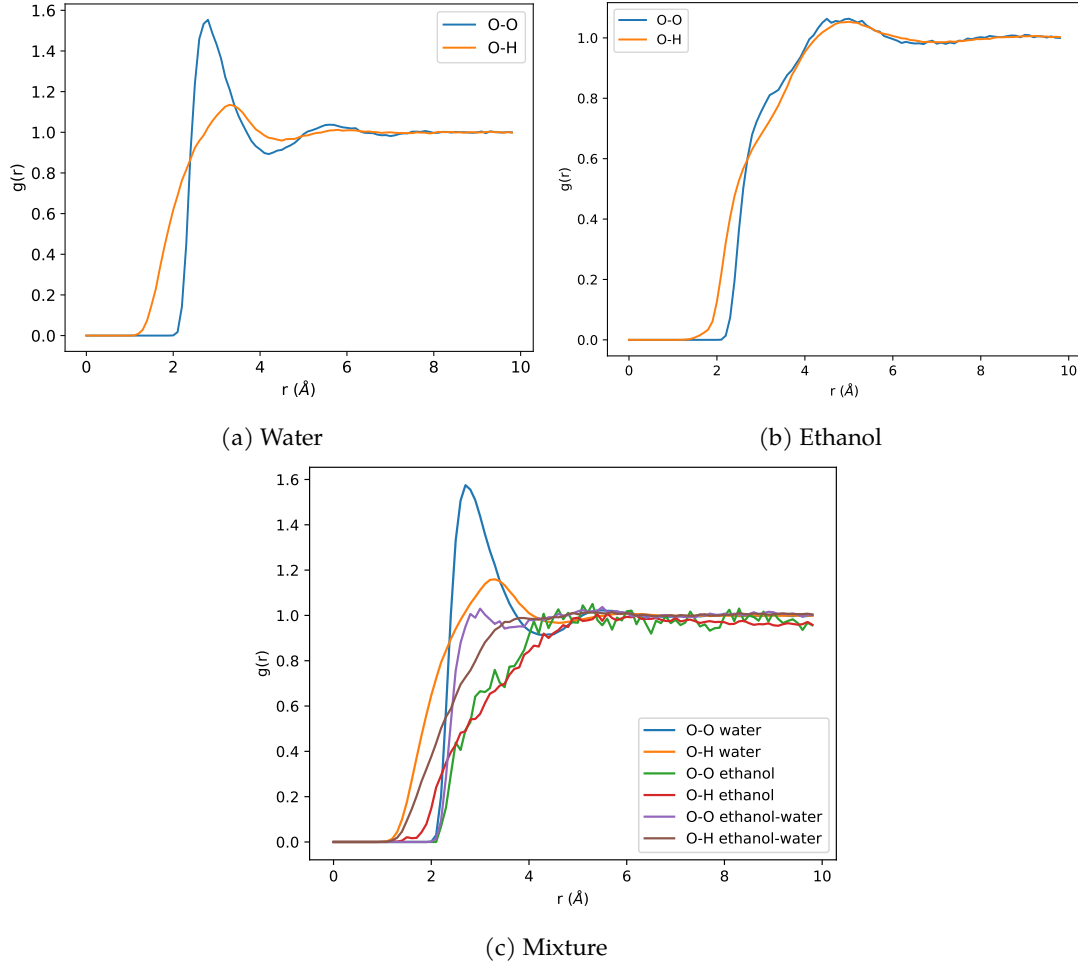


Figure 5: Radial distribution function.

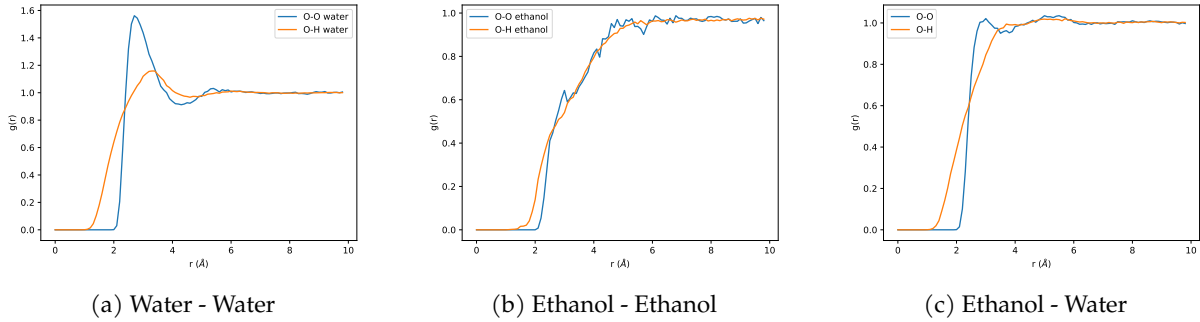


Figure 6: Radial distribution function of mixture per molecule type.

We expect that the local density converges to the average density as the distance increases. This is indeed the case since all RDF's in Figure 5 converge to 1. Note that intramolecular bonds are not included, since the structure of the molecules is fixed so this information is not interesting. The O-O density of water shows a peak at approximately $r = 3\text{Å}$ implying that the local density at this distance is about 50% higher than average. O-H also shows a peak, around $r = 3.5\text{Å}$, but less high.

For H atoms in the simulation, the LJ parameters are $\varepsilon = \sigma = 0$. This means that there are no LJ forces acting on O-H pairs. Looking at the most left part of any of the three plots, we see that the minimal distance between O-H pairs is smaller than the minimal distance between O-O pairs. This might seem counter intuitive, however there are twice as many H atoms as O atoms present in the simulation.

The RDF of ethanol shows similar curves for O-O and O-H densities with non-significant peaks around $r = 5\text{Å}$. In comparison with water, the overall distance between atoms is larger, due to the larger

size of the molecules. The absence of high peaks indicates that there is no 'typical' distance between the molecules, but they are distributed evenly in space.

Looking at the RDF of the mixture, we can still recognize the shapes of the O-H and O-O bonds for water and ethanol individually. Figures 6a, 6b and 6c show the same results split over the different kinds of bonds. Apparently the mixture does not influence the typical distances between these molecules. The RDF of O-O between water and ethanol (6c) interpolates between those of pure O-O pairs of water (6a) and ethanol (6b). The result for O-H pairs is similar.

We can compare our results with those of Soper, 2013 and Zhang and Yang, 2005. The first article focuses specifically on the RDF of water, which is obtained by scatter experiments rather than MD simulation. The shape of the graph for O-O (Fig. 12) is similar but its peak is higher. Moreover a significant peak for the O-H pairs is present around 2Å, which does not appear in our plot. This can be explained by the fact that our model does not include electrostatic forces, which means that our model lacks the very strong force of hydrogen bonds between the O-H groups of different molecules, as described by Lippincott and Schroeder, 1955.

For the mixture we consider Figures 2,3 and 4 of Zhang and Yang, 2005. Similar to our results, the positions of the peaks in the RDF does not change as compared to the pure liquids, only their heights. However the changes in heights are more significant in the results of than ours. In this article we see that the peak for O-O of pure ethanol is much higher than our result in Figure 5b. Next to that we see a high peak for the O-H pairs around 2Å which is not visible in our plot. Again, this can be explained by the missing electrostatic hydrogen bonds.

5 Conclusion

Molecular dynamics simulation is used to model properties of water, ethanol and a mixture of ethanol and water. Based on both theoretical and experimental arguments, the Velocity Verlet scheme is chosen as integrator in the simulation.

From the output of the simulations, the energy levels and radial distribution functions are analyzed. In all substances, the kinetic energy is much higher than the potential energy. Between the substances, the different kinds of potential energies play a different kind of role. The energy levels of the mixture contain features of both pure systems.

The result show a remarkable difference in rates of convergence between the three systems which might be caused by the initial positions of the water molecules being too artificial. Our simulation of the mixture crashed after 212 ps due to unfortunate timing of a Windows update. A better analysis of this system can be done by running a longer simulation.

The model for the simulation only includes bond, angles, dihedral and Lennard-Jones forces. Our results show that these forces are well implemented in the simulation, while they also show the lack of intermolecular forces like electrostatic forces causing differences in the radial distribution function between our results and those of previous works.

Extensions to our model can be made by increasing the volume of the simulated mixtures, varying the temperature, modeling a different type of thermostat in NVT or simulation an ensemble other than the canonical. A more thorough improvement would be to also include electrostatic charges.

Finally, we briefly want to reflect on the programming aspect of this project. In the entire process of developing our MD simulator, we have gained experience in programming in Python, which neither of us had done before. Moreover we have learnt to translate theoretical concepts into modeling concepts and to critically evaluate the resulting model, both from a practical and theoretical viewpoint.

A particularly impressive aspect is that although we start entirely from scratch, even with a fairly simple set of rules we could develop a simulator capable of simulating systems that are far from trivial. Even more, as we have seen in Section 4, the simulator turns out to replicate results from far more complex simulators with reasonable accuracy.

References

- Lippincott, E. R., & Schroeder, R. (1955). One-dimensional model of the hydrogen bond. *The Journal of Chemical Physics*, 23(6), 1099–1106. <https://doi.org/10.1063/1.1742196>
- Alder, B. J., & Wainwright, T. E. (1959). Studies in molecular dynamics. i. general method. *The Journal of Chemical Physics*, 31(459).
- Kvamme, B. (1997). Molecular dynamics simulations and integral equations studies of model systems for aqueous mixtures of small alcohols. *Fluid Phase Equilibria*, 131(1-2), 1–20. [https://doi.org/10.1016/s0378-3812\(97\)00002-2](https://doi.org/10.1016/s0378-3812(97)00002-2)
- Saiz, L., Padró, J. A., Guà, E., & Guàrdia, E. (1997). Structure and dynamics of liquid ethanol. *Journal of Physical Chemistry B*, 101(1), 78–86. <https://doi.org/10.1021/jp961786j>
- Morishita, T. (2000). Fluctuation formulas in molecular-dynamics simulations with the weak coupling heat bath. *Journal of Chemical Physics*, 113(8), 2976–2982. <https://doi.org/10.1063/1.1287333>
- Delhommelle, J., & Millié, P. (2001). Inadequacy of the Lorentz-Berthelot combining rules for accurate predictions of equilibrium properties by molecular simulation. *Molecular Physics*, 99(8), 619–625. <https://doi.org/10.1080/00268970010020041>
- Martínez, J. M., & Martínez, L. (2003). Packing optimization for automated generation of complex system's initial configurations for molecular dynamics and docking. *Journal of Computational Chemistry*, 24(7), 819–825. <https://doi.org/10.1002/jcc.10216>
- Washburn, E. (2003). International critical tables of numerical data, physics, chemistry and technology.
- Mudi, A., & Chakravarty, C. (2004). Effect of the Berendsen thermostat on the dynamical properties of water. *Molecular Physics*, 102(7), 681–685. <https://doi.org/10.1080/00268970410001698937>
- Zhang, C., & Yang, X. (2005). Molecular dynamics simulation of ethanol/water mixtures for structure and diffusion properties. *Fluid Phase Equilibria*, 231(1), 1–10. <https://doi.org/10.1016/j.fluid.2005.03.018>
- Berryman, P. J. (2006). *Molecular Dynamics Simulations of Ethanol and Ethanol-Water Mixtures* (Doctoral dissertation December).
- Toxvaerd, S., & Dyre, J. C. (2011). Communication: Shifted forces in molecular dynamics. *The Journal of Chemical Physics*, 134(081102).
- Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, 13(2), 22–30. <https://doi.org/10.1109/MCSE.2011.37>
- Haken, H., & Wolf, H. C. (2013). *Molecular physics and elements of quantum chemistry: Introduction to experiments and theory*. Springer Science & Business Media.
- Holm, C. (2013). *Simulation methods in physics 1*. Institute for Computational Physics University of Stuttgart.
- Soper, A. K. (2013). The Radial Distribution Functions of Water as Derived from Radiation Total Scattering Experiments: Is There Anything We Can Say for Sure? *ISRN Physical Chemistry*, 2013, 1–67. <https://doi.org/10.1155/2013/279463>
- Wang, L., Ishiyama, T., & Morita, A. (2017). Theoretical Investigation of C-H Vibrational Spectroscopy. 2. Unified Assignment Method of IR, Raman, and Sum Frequency Generation Spectra of Ethanol. *Journal of Physical Chemistry A*, 121(36), 6701–6712. <https://doi.org/10.1021/acs.jpca.7b05378>
- Braun, E., Moosavi, S. M., & Smit, B. (2018). Anomalous Effects of Velocity Rescaling Algorithms: The Flying Ice Cube Effect Revisited. *Journal of Chemical Theory and Computation*, 14(10), 5262–5272. <https://doi.org/10.1021/acs.jctc.8b00446>

A Force-field parameters

This appendix contains the parameters which are used in our simulation. We follow the labeling of atoms within each molecule as shown in Figures 7a and 7b.

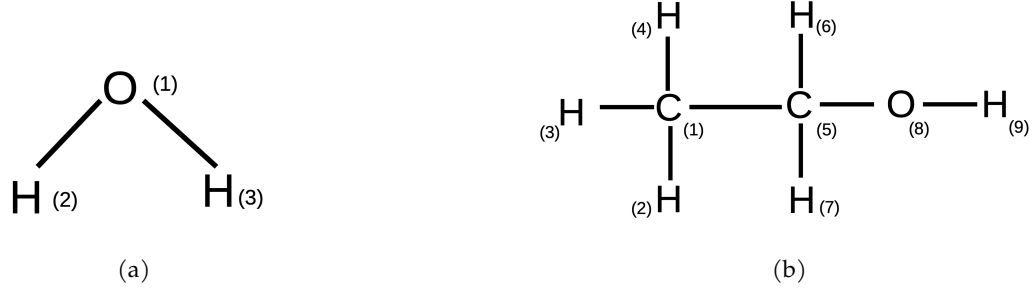


Figure 7: Atom labeling of water (7a) and ethanol (7b).

The following parameters are used in all simulations.

Species	Mass (amu)
O	15.9994
H	1.0080
C	12.0110

Table 6: Atom masses

ij	R_0 (Å)	k_b (kJ/(mol nm ²))
12, 13	0.9572	502416

Table 7: Bond force constants water

ij	R_0 (Å)	k_b (kJ/(mol nm ²))
15	1.529	224262.4
31, 41, 21, 65, 75	1.090	284512.0
58	1.1410	267776.0
98	0.945	462750.0

Table 8: Bond force constants ethanol

ijk	θ_0 (rad)	k_θ (kJ/(mol rad ²))
213	1.8242	628.02

Table 9: Angular force constants water

ijk	θ_0 (rad)	k_θ (kJ/(mol rad ²))
215, 315, 415	1.8937	292.880
413, 412, 312, 657	1.8815	276.144
157, 156	1.9321	313.800
158	1.9111	414.400
589	1.8937	460.240
658, 758	1.9111	292.880

Table 10: Angular force constants ethanol

$ijkl$	C_1 (kJ/mol)	C_2 (kJ/mol)	C_3 (kJ/mol)	C_4 (kJ/mol)
2156, 3156, 4156 2157, 3157, 4157	0.62760	1.88280	0	-3.91622
2158, 3158, 4158	0.97905	2.93716	0	-3.91622
1589	-0.44310	3.83255	0.72801	-4.11705
6589, 7589	0.94140	2.82420	0	-3.76560

Table 11: Dihedral force constants ethanol

Species	σ (Å)	ε (kJ/mol)
O	3.15061	0.66386
H	0	0

Table 12: Lennard-Jones parameters water

Species	σ (Å)	ε (kJ/mol)
H (bonded to C)	2.5	0.125520
H (bonded to O)	0	0
C	3.5	0.276144
O	3.12	0.711280

Table 13: Lennard-Jones parameters ethanol

B Calculation of number molecules to simulate

All densities and molecule weights are taken from Washbrun, 2003.

Pure water: Density of water at $T = 298.15K$ is $\rho = 997kg/m^3$ Washbrun, 2003, so we have $1.25 \cdot 10^{-25} \cdot 997 = 1.24625 \cdot 10^{-22}$ kg of water. One molecule of water weighs $2.991508 \cdot 10^{-26}$ kg, so we need $1.24625 \cdot 10^{-22} / 2.991508 \cdot 10^{-26} \approx 4166$ molecules.

Pure ethanol: Density of ethanol at $T = 298.15K$ is $\rho = 0.7849g/mL = 784.9kg/m^3$ Washbrun, 2003, so we have $1.25 \cdot 10^{-25} \cdot 784.9 = 9.81125 \cdot 10^{-23}kg$ of ethanol. The molecular mass of ethanol is $46g/mol = 46 \cdot 10^{-3}kg/mol$, so one molecule of ethanol weighs $46 \cdot 10^{-3} / 6.02214076 \cdot 10^{23} = 7.63847 \cdot 10^{-26}kg$. Thus, we need $9.81125 \cdot 10^{-23} / 7.63847 \cdot 10^{-26} \approx 1284$ molecules.

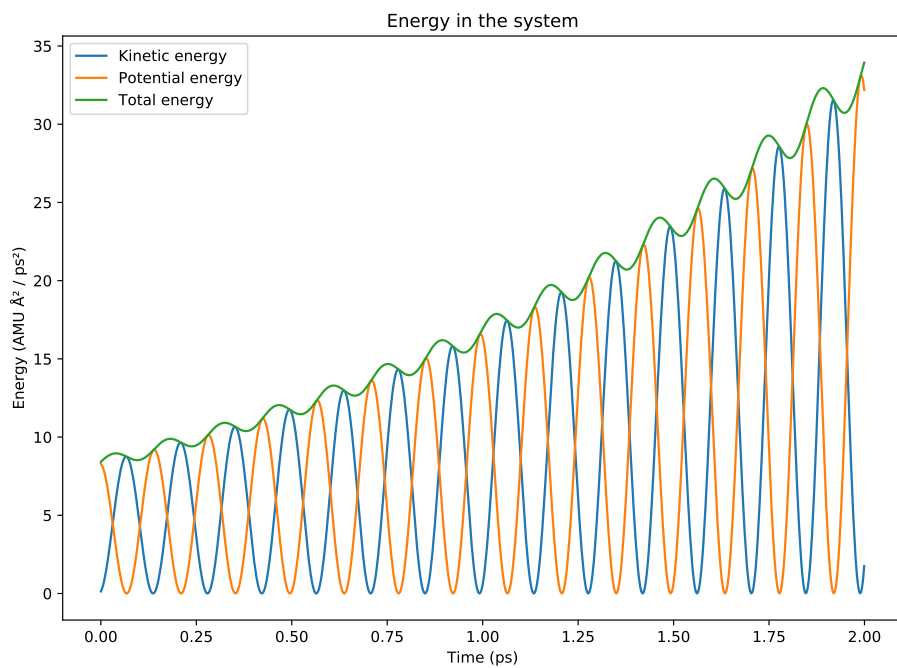
Mixture: The density of a mixture of 14.3% ethanol and 85.7% water at $T = 298.15K$ is $\rho = 0.97398g/mL = 974.0kg/m^3$ (Washbrun, 2003). Hence, we have $1.25 \cdot 10^{-25} \cdot 974.0 = 1.2175 \cdot 10^{-22}kg$ of mixture. This means we have $0.143 \cdot 1.2175 \cdot 10^{-22} = 1.741025 \cdot 10^{-23}kg$ of ethanol and $0.857 \cdot 1.2175 \cdot 10^{-22} = 1.0433975 \cdot 10^{-22}kg$ of water.

Analogous to the computations above, we find that we need $1.741025 \cdot 10^{-23} / 7.63847 \cdot 10^{-26} \approx 228$ molecules of ethanol and $1.0433975 \cdot 10^{-22} / 2.991508 \cdot 10^{-26} \approx 3488$ molecules of water.

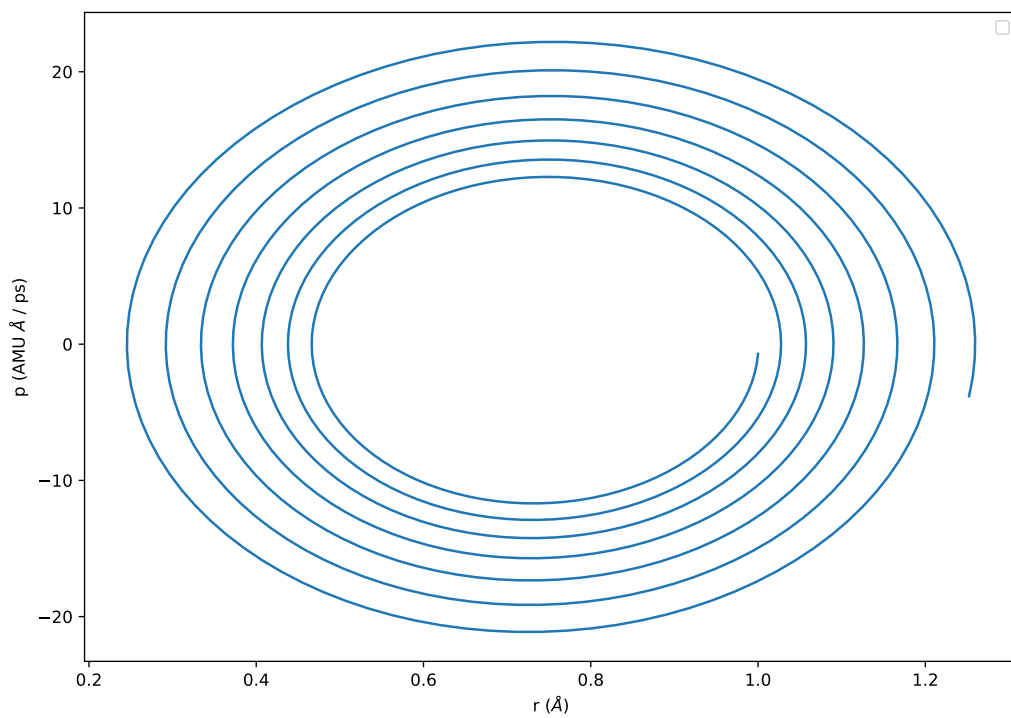
Summarising, a 5nm cubic box with atmospheric pressure at a temperature of 298.15K can be filled with 4166 molecules, 1284 ethanol molecules, or a mixture of 3488 water molecules and 228 ethanol molecules.

For a 3 nm box, the numbers of molecules resulting in the same approximate density are obtained by scaling the numbers above by a factor of $(\frac{3}{5})^3$. This results in 900 water molecules, 277 ethanol molecules, or a mixture of 753 water molecules and 49 ethanol molecules.

C Energy and phase space diagrams for integrator analysis

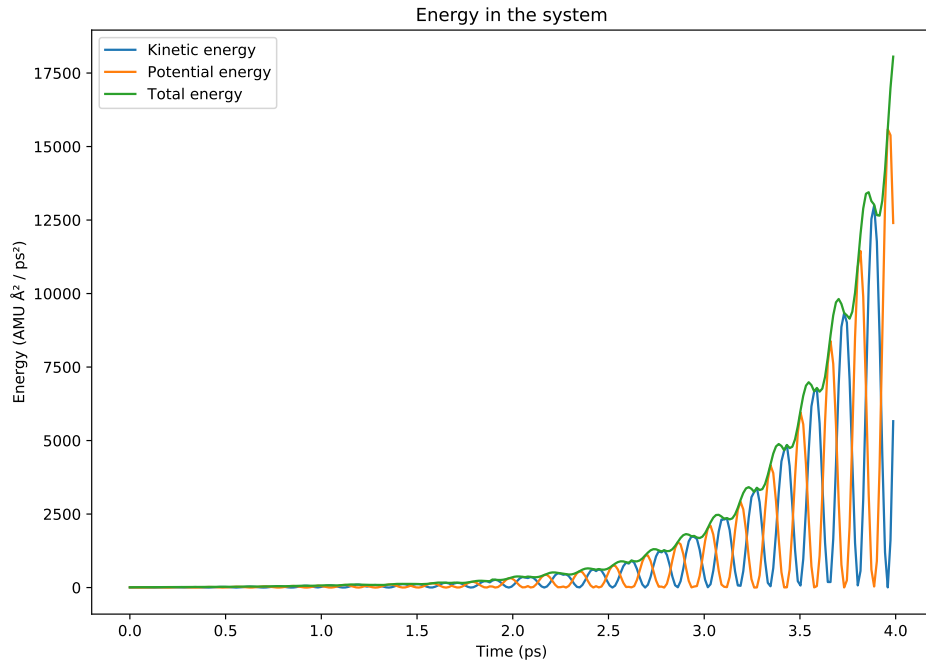


(a) Energy

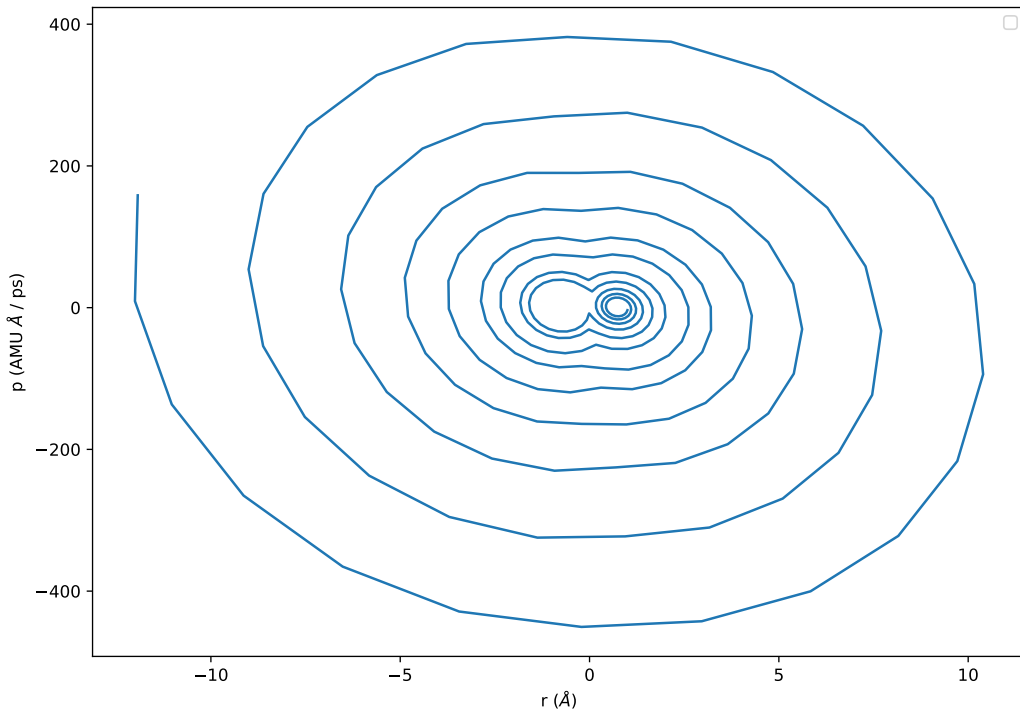


(b) Phase space

Figure 8: Energy and phase space diagrams of Euler integrator for $\Delta t = 0.1T$.

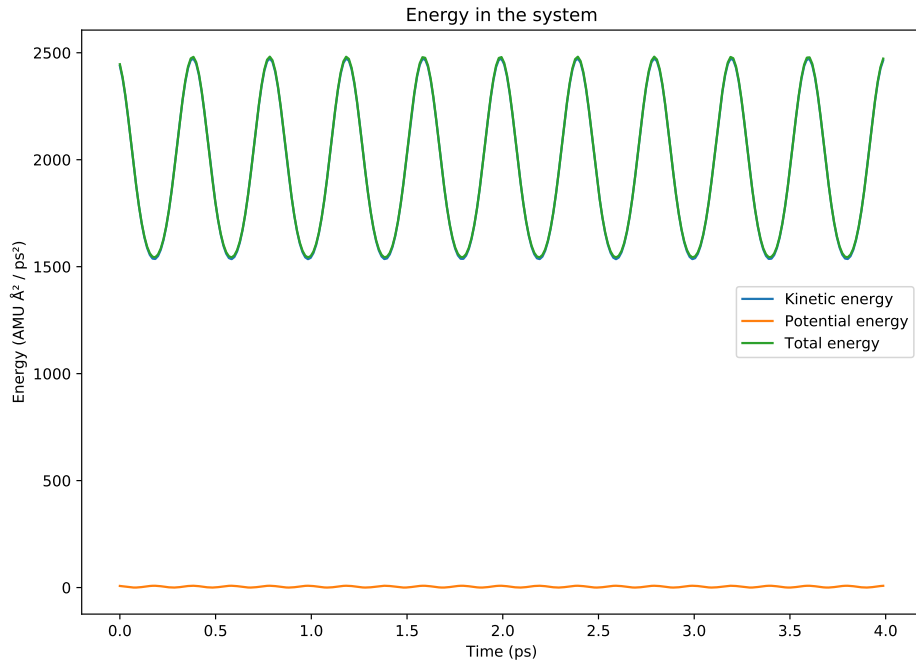


(a) Energy

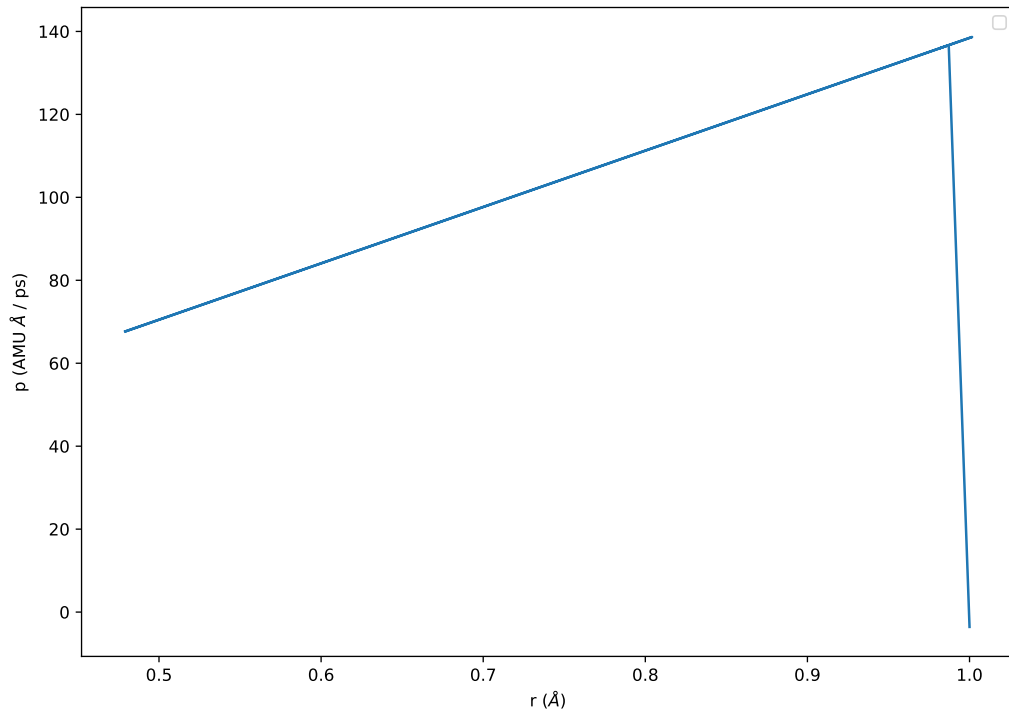


(b) Phase space

Figure 9: Energy and phase space diagrams of RK4 integrator for $\Delta t = 0.5T$.

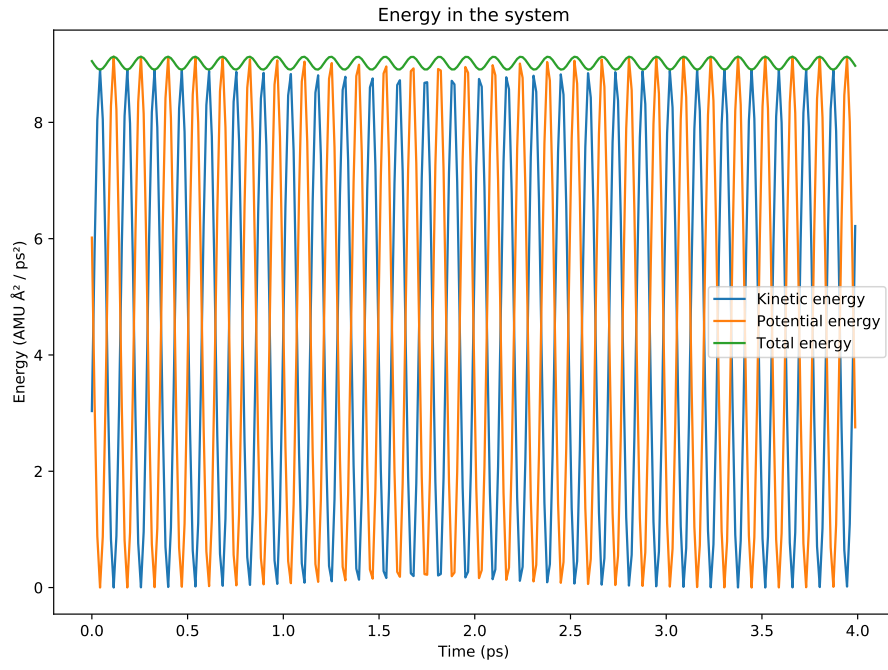


(a) Energy

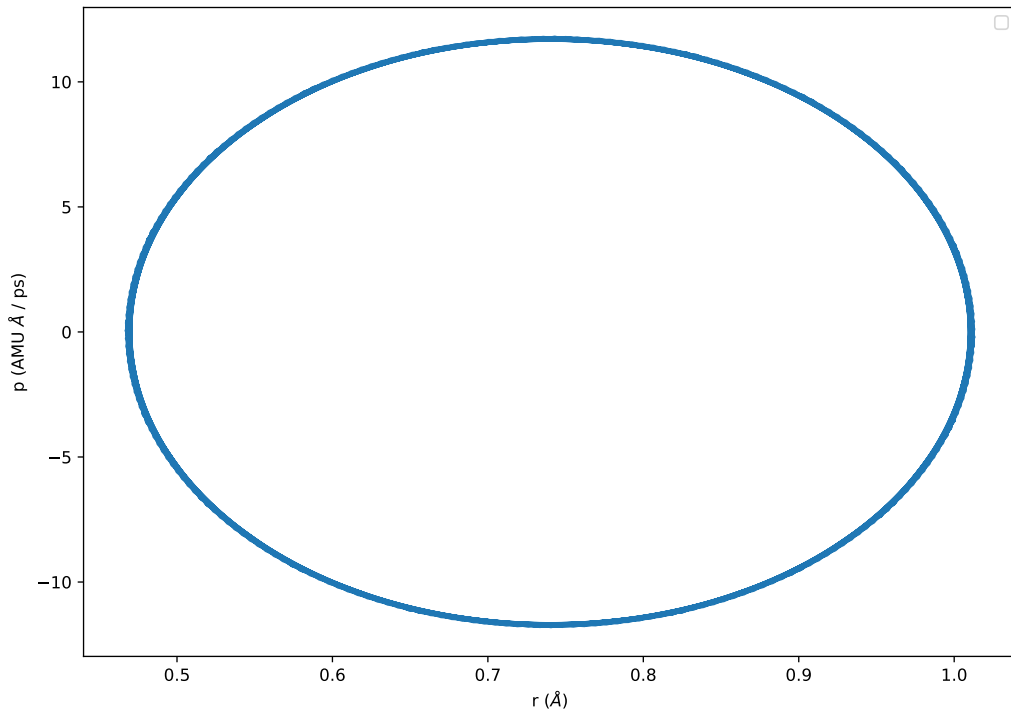


(b) Phase space

Figure 10: Energy and phase space diagrams of Verlet integrator for $\Delta t = 0.5T$.



(a) Energy



(b) Phase space

Figure 11: Energy and phase space diagrams of Velocity Verlet integrator for $\Delta t = 0.5T$.