

Simulating Langevin dynamics using Effective Verlet-type algorithm

Gayatri Panda

2011068

Abstract

This computational project explores the dynamics of interacting particles through Langevin dynamics simulations employing Lennard-Jones potentials within the canonical ensemble (NVT ensemble). The study focuses on simulating the behavior of N particles over time using a modified velocity Verlet algorithm. The project aims to understand the intricate interplay between deterministic and stochastic forces governing particle motion in a dissipative environment by integrating Langevin dynamics with Lennard-Jones potentials. Through extensive simulations and analysis, insights into the particle ensemble's collective behavior, equilibrium properties, and dynamic phenomena are obtained. The simulation parameters, process, and results are thoroughly discussed

1 Molecular Dynamics

Molecular dynamics (MD) is a computational technique simulating the movements of atoms and molecules, pivotal in understanding molecular behavior in diverse fields like chemistry, biology, and materials science. It employs Newton's laws to predict particle trajectories, often using force fields to model interactions. By numerically solving equations of motion, MD elucidates complex phenomena such as protein folding, chemical reactions, and material properties. Advancements in computing have broadened MD's applicability, enabling simulations of larger and more intricate systems over extended timescales. Its synergy with experimental methods enhances our understanding of molecular processes and aids in designing new materials and drugs. MD provides a powerful tool for exploring molecular dynamics, offering insights into fundamental mechanisms and fostering innovation across scientific disciplines.

1.1 Evolution in Time

In Monte Carlo (MC) methods, a fundamental approach in statistical mechanics, averaging occurs across the system's phase space rather than along time-dependent trajectories as in Molecular Dynamics (MD). Both methods face inherent errors: MC requires averaging across the entire phase space but can only sample a limited number of points. At the same time, MD theoretically demands infinite time for accurate averaging but practically operates over finite durations. Nevertheless, these errors can be mitigated. In MC, optimization of phase space sampling can minimize inaccuracies, while in MD,

precise estimation of the system's characteristic timestep aids in error reduction. By addressing these challenges, both MC and MD simulations contribute valuable insights into the behavior of complex systems in physics and chemistry. write differently

Molecular Dynamics (MD) primarily operates within the microcanonical ensemble, where the number of particles (N), volume (V), and total energy (E) of the system are kept constant. This means that we fix the system's total energy when we simulate molecular dynamics. However, we must maintain other properties like temperature or pressure at a constant value in many real-world scenarios.

To achieve this, we must use different statistical ensembles, such as the canonical ensemble. In the canonical ensemble, instead of fixing the total energy, we keep the temperature of the system constant. This allows us to simulate more realistic situations where the system's temperature is critical. By employing different statistical ensembles, we can effectively model a wide range of physical conditions and better understand the behavior of complex systems in various environments.

1.2 Interactions- Lennard Jones Potential

One of the well-known interparticle potentials used in MD simulations is the Lennard-Jones (L-J) pair interaction. It provides a model for simple yet realistic interacting systems. In the L-J potential, the pairwise interaction is true of the van der Waals form, namely r^{-6} with r the interparticle distance,

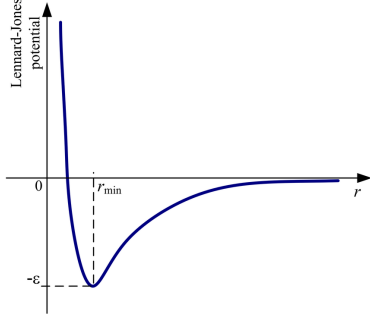


Figure 1: Lennard Jones potential

which describes attraction at long distances. This term could in fact be derived solely by using quantum mechanics. Adding repulsion at short distances via r^{-12} —according to the Pauli repulsion—then results in the L-J potential:

$$U_{\text{LJ}}(r) = -4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (1)$$

where $r = |\mathbf{r}_1 - \mathbf{r}_2|$, \mathbf{r}_1 and \mathbf{r}_2 denote respectively the position vectors of particles 1 and 2, ϵ is the depth of the potential well, and σ —often referred to as the particle size—is the distance at which U_{LJ} vanishes. $r_{\text{min}} = 2^{1/6}\sigma$ also corresponds with the minimum of the potential, as illustrated below:

2 Langevin Dynamics

Langevin dynamics is a stochastic differential equation commonly used in molecular dynamics simulations to model the behavior of particles in a dissipative environment. It incorporates both deterministic and stochastic forces to simulate the motion of particles under the influence of thermal fluctuations.

The Langevin equation is given by:

$$m \frac{d^2 \mathbf{r}}{dt^2} = -\nabla U(\mathbf{r}) - \gamma \frac{d\mathbf{r}}{dt} + \sqrt{2mk_B T} \boldsymbol{\xi}(t)$$

where:

- m is the mass of the particle,
- \mathbf{r} is the position vector of the particle,
- $U(\mathbf{r})$ is the potential energy function,
- γ is the friction coefficient representing the strength of the dissipative forces,
- k_B is Boltzmann's constant,
- T is the temperature,

- $\boldsymbol{\xi}(t)$ is a Gaussian white noise term with zero mean and unit variance, representing random thermal fluctuations.

Let's break down the Langevin equation into its components:

2.1 Deterministic Force

The deterministic force term is given by the negative gradient of the potential energy function:

$$\mathbf{F}_d = -\nabla U(\mathbf{r})$$

2.2 Dissipative Force

The dissipative force term is proportional to the velocity of the particle and the friction coefficient:

$$\mathbf{F}_{\text{diss}} = -\gamma \frac{d\mathbf{r}}{dt}$$

2.3 Stochastic Force

The stochastic force term represents the random thermal fluctuations and is given by:

$$\mathbf{F}_{\text{st}} = \sqrt{2mk_B T} \boldsymbol{\xi}(t)$$

3 Simulation

3.1 Initialization

Proper initialization in Molecular Dynamics ensures realistic simulations. Randomly generating initial positions and velocities while avoiding particle overlap and excessive repulsion is crucial. Initial conditions should accurately represent the system's properties to achieve meaningful and stable results.

3.2 COM frame

We know that the temperature T of a d -dimensional system is given by the equipartition theorem as

$$T = \frac{1}{Nk_B d} \sum_{i=1}^N m_i v_i^2 \quad (2)$$

where the v_i values are calculated with respect to the center-of-mass (COM) reference frame. Hence, A computer routine must be written that resets the COM velocity to zero or subtracts the COM velocity from the atomic velocities every once in a while.

3.3 Solving differential equations of motion

In Molecular Dynamics (MD), solving the differential equations of motion is crucial for accurately simulating particle behavior. These equations describe how the positions and velocities of particles change over time due to forces acting upon them. Numerical integration methods, such as the Verlet algorithm, are commonly employed to solve these equations efficiently.

3.3.1 Verlet Algorithm

The Verlet algorithm is a symplectic integrator, meaning it preserves specific system dynamics properties, such as energy conservation, over long simulation times. It calculates particle positions at a future time step based on their current positions, velocities, and the forces acting on them. This algorithm is computationally efficient and stable, making it well-suited for MD simulations.

The Verlet method, which is based on truncated Taylor expansions for the evolution of a particle with mass m , coordinate $r(t)$, velocity $v(t)$, and force $f(r, t)$ [?, ?]. Introducing the discrete-time variables $r_n = r(t_n)$, $v_n = v(t_n)$, and $f_n = f(r_n, t_n)$, the so-called velocity explicit Verlet (or velocity Verlet) scheme reads:

$$r_{n+1} = r_n + v_n \Delta t + \frac{\Delta t^2}{2m} f_n,$$

$$v_{n+1} = v_n + \frac{\Delta t}{2m} (f_n + f_{n+1}).$$

3.3.2 Effective Verlet Method

The Verlet scheme, accurate to second order in time step dt , ensures trajectory deviations scale with the third power of dt . Widely favored for MD simulations, it offers convenience, efficiency, and time reversibility, preserving energy error bounds over long integrations. The canonical ensemble, common in statistical mechanics, maintains constant temperature. Integrators for Langevin dynamics (LD) simulations, a popular approach, incorporate frictional and thermal forces into the conservative force field. The Langevin equation of motion combines these forces, enhancing simulation accuracy and realism.

$$\dot{r} = v,$$

$$m\dot{v} = f(r, t) - \alpha v + \beta(t).$$

It is often assumed that the stochastic force is Gaussian distributed, and has the statistical properties [5]

$$\langle \beta(t) \rangle = 0,$$

$$\langle \beta(t) \beta(t') \rangle = 2\alpha k_B T \delta(t - t'),$$

The difficulty in developing accurate integrators for Langevin dynamics (LD) stems from the non-analytic nature of $\beta(t)$, which invalidates the Taylor expansion commonly used for the derivation of the Verlet scheme.

$$r_{n+1} = r_n + bdt v_n + \frac{bdt^2}{2m} f_n + \frac{bdt}{2m} \beta_{n+1},$$

$$v_{n+1} = v_n + \frac{dt}{2m} (f_n + f_{n+1}) - \frac{\alpha}{m} (r_{n+1} - r_n) + \frac{1}{m} \beta_{n+1}.$$

where,

$$b \equiv \frac{1}{1 + \frac{\alpha dt}{2m}}.$$

$$\beta_{n+1} \equiv \int_{t_n}^{t_{n+1}} \beta(t') dt'$$

is a Gaussian random number with $\langle \beta_n \rangle = 0$ and

$$[\langle \beta_n \beta_l \rangle = 2\alpha k_B T dt \delta_{n,l}.$$

We use this scheme for solving our equations of motions and evolving the system

3.4 Calculating forces

From the L-J potential (i.e., Eq. 1), the pairwise force along the x direction is obtained as follows:

$$U_{ij} = 4 \left(\frac{1}{r_{ij}^{12}} - \frac{1}{r_{ij}^6} \right)$$

$$\Rightarrow (f_{ij})_x = -\frac{\partial U_{ij}}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial x} = 4 \left(\frac{12}{r_{ij}^{12}} - \frac{6}{r_{ij}^6} \right) \frac{x_{ij}}{r_{ij}^2},$$

where σ and ϵ have been set to unity. Factoring out r_{ij}^{-2} on the right-hand side is crucial. This seemingly trivial simplification allows the calculation of r_{ij}^{-6} and r_{ij}^{-12} once for both the potential energy and the force. Otherwise, one would have to compute r_{ij}^{-6} and r_{ij}^{-12} for the potential, and r_{ij}^{-8} and r_{ij}^{-14} for the force, significantly increasing computational costs. Nearly all computational costs arise from force calculations due to their pairwise nature, unlike the Verlet algorithm, which is essentially a single particle. Specifically, the Verlet routine is of order N , while the force (acceleration) routine is of order.

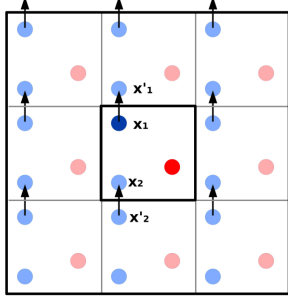


Figure 2: Periodic boundary conditions

3.5 Boundary conditions

3.5.1 Periodic Boundary Conditions

In MD simulations, periodic boundary conditions (PBC) are used for two purposes: (I) to keep fixed the number of particles within the simulation box (the primary cell) of the system; and (II) to eliminate particles out of the cutoff radius to reduce the computational cost

Keeping the total number of particles fixed simply so that once a particle leaves the box, the same one enters from the opposite side. This is as though the related computer routine only shifts the position of the outgoing particle to enter it from the opposite side of the box.

PBCs are also necessary when we use short-range potentials and will eliminate particles outside the cutoff radius. The minimum image convention means that each particle interacts with the nearest image of any other given particle. This is specific to periodic boundary conditions. Because of PBC, one particle within the primary cell interacts with other particles within the primary cell and those within image cells. Assume the box size to be L and the cutoff radius $r_c = L/2$

For the Lennard-Jones potential, the cutoff radius r_c is typically set to 2.50σ . However, a challenge arises because the L-J potential is not differentiable at $r = r_c$, as shown in the figure below. Consequently, the force at this point cannot be accurately calculated. As seen, $U_{LJ}(r_c) \neq 0$, posing a problem where particles experience sudden freedom upon leaving the cutoff sphere. The simplest solution is to shift the potential by U_{LJ} to $U_{LJ} - U_{LJ}(r_c)$.

3.6 Thermodynamic Equilibrium

The reliability of simulations in calculating statistical averages emerges as the system nears thermodynamic equilibrium. Initially, the system is far from

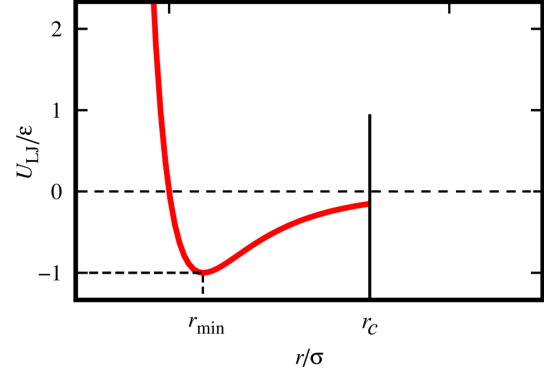


Figure 3: cutoff in lennard jones at r_c

equilibrium with random positions and velocities. Thermodynamic properties like temperature are only defined at equilibrium. Monitoring the time evolution of total energy given in the section of results

As mentioned earlier, The microcanonical ensemble defines the system's macrostate by fixed values of N , V , and E or a fixed energy range $(E, E + \Delta)$. The main challenge lies in determining the total number $\Omega(N, V, E)$ (or $\Omega(N, V, E; \Delta)$) of distinct accessible microstates, from which the system's complete thermodynamics could be derived. However, calculating Ω for most physical systems is often formidable if not intractable. Furthermore, the notion of a fixed energy, or even a fixed energy range, for real-world systems is unsatisfactory, given their inherent complexity.

An alternative to fixed energy is fixed temperature, which not only is measured directly in the lab using thermometers, but also is controllable and could be kept fixed by keeping the system in contact with an appropriate thermal bath. Such a so-called generalization is referred to as the canonical ensemble in that the macrostate of the system is defined by the fixed numbers N , V , and T . This is indeed one of the most prevalent ensembles used for MD simulations.

Maintaining a fixed system temperature in MD simulations requires an additional computational trick. Temperature quickly deviates from the target equilibrium value, reaching very high values. This issue is addressed by implementing a thermostat at the programming level. The thermostat ensures that the system's instantaneous temperature fluctuates around the target temperature, aligning with the equipartition theorem (Eq. 2). This technique is widely used in MD codes.

From Eq. 2, the temperature T of the system is directly correlated with atomic velocities—it is essential to mention that Eq. 2 gives the instantaneous temperature, or equivalently, the system’s average temperature at one timestep. Therefore, one has to place constraints on atomic velocities to control the instantaneous temperature. Assume that the target temperature is T_0 , which is not equal to the instantaneous temperature T . Therefore, it is natural to assume that multiplying the atomic velocity v_i by a dimensionless factor (say β) will eventually lead exactly to T_0 :

$$\frac{1}{Nk_Bd} \sum_{i=1}^N m_i (\beta v_i)^2 = T_0$$

$$\Rightarrow \beta^2 T = T_0 \Rightarrow \beta = \left(\frac{T_0}{T} \right)^{1/2}$$

As a result, the assignment statement $v_i = \beta v_i$ at the programming level makes the average of T very close to T_0 . This type of temperature control is called rescaling. It is through rescaling the instantaneous atomic velocities that the time dependence of the thermodynamic quantities of interest (such as energy or temperature) exhibit fluctuating patterns.

4 Simulation parameters

4.1 Making physical observables dimensionless

Here we consider the simulation of a system composed of argon atoms interacting via the Lennard-Jones (L-J) potential. The mass m_A of an argon atom and the related L-J parameters, namely ϵ_A and σ_A . $m_A = 6.7 \times 10^{-26} \text{ kg}$, $\epsilon_A \text{ (J)} = 1.6 \times 10^{-21}$, $\sigma_A = 3.4 \times 10^{-10} \text{ (m)}$

As a result, $m^* \equiv \frac{m}{m_A}$, $E^* \equiv \frac{E}{\epsilon_A}$, and $r^* \equiv \frac{r}{\sigma_A}$ are respectively the dimensionless mass, energy, and distance, which form the reduced units. These dimensionless quantities prevent the MD simulation from generating very large or very small values. Since the dimensions of mass, energy, and length are respectively M , ML^2T^{-2} , and L , the characteristic time for simulating a system of argon atoms,

$$\frac{r}{\frac{ML^2}{ML^2T^{-2}}} = T \Rightarrow \tau_A = \frac{\sigma_A}{r_m \sqrt{\frac{m_A}{\epsilon_A}}} \approx 10^{-12} \text{ s} = 1 \text{ ps}.$$

The dimensionless time is then $t^* \equiv \frac{t}{\tau_A}$. This specific value, namely τ_A , is indeed the characteristic timestep of a gas of argon atoms, ensuring a logical

relationship between consecutive events based on the principle of causality. Timesteps considerably larger than τ_A lead to simulation crashes due to loss of causality, while values much smaller increase computational cost and slow the simulation dramatically. Timesteps of about 10^{-3} , $-4 \tau_A = 10^{-15}$, -16 s are optimal for simulating systems at the scale of atoms and molecules.

The characteristic velocity is also $v_A = \frac{\sigma_A}{r_m \sqrt{\frac{m_A}{\epsilon_A}}} \approx 340 \text{ m/s}$, which is nearly the same as the velocity of sound.

4.2 Creating particles

The easiest way to generate initial positions is to use random number generators. However, a significant drawback arises when atoms are too close, causing a strong repulsive force and leading to simulation crashes. Employing random number generators with uniform distributions can mitigate this problem.

As for setting the total number N of particles and the size L of the simulation box within which particles move, let’s assume there exists a three-dimensional, uniform lattice at each point of which an argon atom is located, resulting in a total of 1000 atoms. According to previous discussions, L must be $10\sigma_A$.

Initial velocities and accelerations being random pose no significant problem; they can be set to zero. Despite initial velocities not conforming to the Maxwell-Boltzmann distribution at $t=0$, the system self-corrects over time steps.

5 Results

We ran the simulations numerous times with varying combinations of the number of atoms and the number of time steps; the discrepancies in evolution from theory can most probably be attributed to the small number of particles that could be simulated due to computation limitations. Several particles can be simulated with that code, but this cannot be done due to the limitation of computation resources.

5.1 Free particle langevin dynamics

Here we plot the evolution of system parameters for a system of 100 particles using random forces on each particles without any interacting potentials.

For free particles, where there's no external potential acting on them, the Langevin equation simplifies. The deterministic force term vanishes as there's no potential energy function to consider. Thus, the motion is primarily influenced by the dissipative force, which tends to dampen the particles' velocity, and the stochastic force, which introduces random fluctuations.

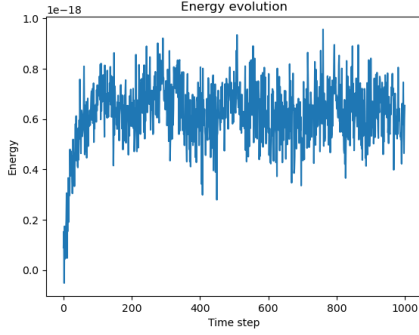


Figure 4: Energy Evolution of system with 100 particles .

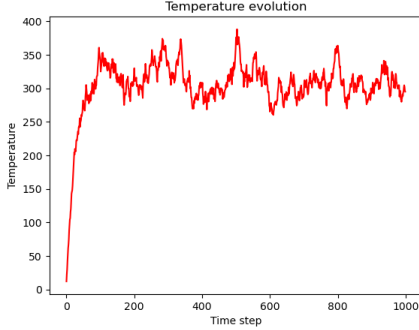


Figure 5: Temperature Evolution of system with 100 particles.

Using the velocity autocorrelation function $C_v(\tau)$, one can estimate the equilibrium time of the system. $C_v(\tau)$ measures how the velocity at $t + \tau$ is correlated with its value at t , providing an estimation of the time during which the system loses its memory of the previous atomic velocities. where the denominator is equal to $(N - 1)dT$ for a d -dimensional system of N particles at the equilibrium temperature T . $\langle v(t) \rangle$ is always zero based on the following reasoning. If the system is large enough, after an infinite time, and within a uniform and isotropic universe, the particles would accordingly move along all directions, and the average velocity would be zero.

$$C_v(\tau) = \frac{\langle v_i(t)v_i(t + \tau) \rangle}{(N - 1)dT}.$$

Fitting the exponential function $e^{-\tau/\tau_0}$ to the obtained curve gives τ_0 as the equilibrium time of the system.

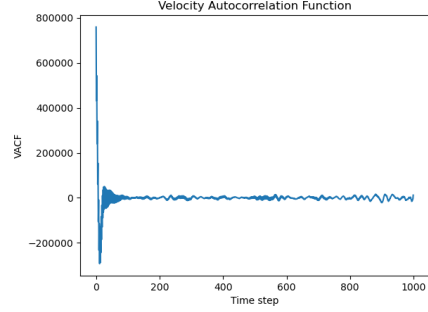


Figure 6: Velocity autocorrelation of system with reflective boundaries.

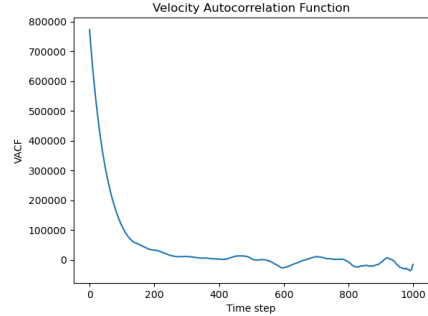


Figure 7: Velocity autocorrelation of system with periodic boundaries.

5.2 Lennard Jones + Random forces along with periodic boundary conditions

Here, one can see the thermostat being established in the temperature plots; The fluctuations in the temperature are due to the random Gaussian forces, and the plots of energy are reaching equilibrium around a point, the progress is not smooth, it is due to the fewer no. of steps and the fewer no. of particles we are considering, as we increase the no. of the particles, we will see a better graph

Here we give plots for evolution of temperature,

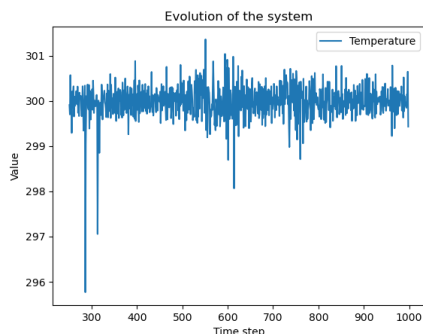


Figure 8: Temperature Evolution of system with 1000 particles.

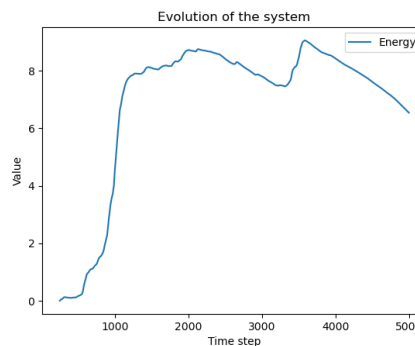


Figure 10: Energy Evolution of system with 150 particles.

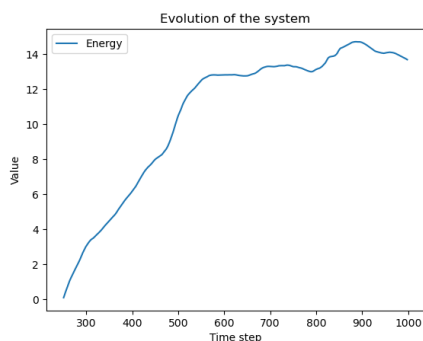


Figure 9: Energy Evolution of system with 1000 particles.

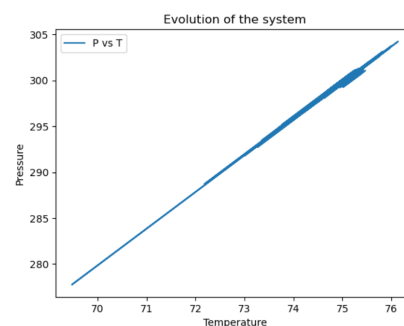


Figure 11: Pressure evolution of the system with 250 particles.

potential energy and pressure at for some of the simulation runs, all resultant plots for remaining runs have been given in the **Github** page: [Link to plots](#).

References

1. Khalatur, P. G. (2012). Molecular Dynamics Simulations in Polymer Science. In Elsevier eBooks (pp. 417–460). <https://doi.org/10.1016/b978-0-444-53349-4.00016-9>
2. Niels Grønbech-Jensen & Oded Farago (2013) A simple and effective Verlettype algorithm for simulating Langevin dynamics, Molecular Physics, 111:8, 983-991, DOI:
3. White, J. A. (1999). Lennard-Jones as a model for argon and test of extended renormalization group calculations. Journal of Chemical Physics Online/the Journal of Chemical Physics/Journal of Chemical Physics, 111(20), 9352–9356. <https://doi.org/10.1063/1.479848> 10.1080/00268976.2012.760055
4. Pastor, R. W. (1994). Techniques and applications of Langevin Dynamics Simulations. In Springer eBooks (pp. 85–138). <https://doi.org/10.1007/978-94-011-1168-35>