Molecular Dynamics simulation

Ayoub Sbai

May 3, 2023

1 Introduction

Molecular dynamics simulation is a numerical simulation used to study the behavior of molecular systems at the atomic scale. This modeling technique makes it possible to predict the physical and chemical properties of these systems by simulating their temporal evolution under the effect of intermolecular forces. In this project, the goal is to simulate the molecular dynamics of ACN-[BMI][BF4] electrolyte and examine how the diffusion coefficients change in a considered temperature interval.
In this document, we will explain in detail the principle of the simulation, the simulation steps by describing them, from the preparation of the initial data to the analysis of the results, then we will identify the indicators used to verify the convergence of the calculation for a simulation of the dynamics analysis of an ACN-[BMI][BF4] electrolyte, as well as a detailed discussion of the results obtained.

2 Software used for the simulation:

•fftool: A python script program used for preparing force field inputs for the construction of the molecular system and the simulation of molecular dynamics.
•Packmol: generation program of the studied molecular system (initialization of molecules in a way that prevents the disruptive effects of short-range repulsion forces arising form Pauli's exclusion principle)
•VMD: visualization program.
•LAMMPS: distribution of programs allowing the simulation of molecular dynamics (OpenMM, CHARMMS, GROMACS, NAMD..etc can be used too)

3 Principle of MD simulation

MD simulation is based on the "Laplacian" vision of classical Newtonian mechanics (Laplace's demon thought experiment). It's a computational method used to predict the dynamics of atoms and molecules as they evolve in time. In contrast to statistical mechanics, MD simulation solves the complexity of physical systems (system size; number of particles) by numerically (as opposed to statistically) solving Newton's equations iteratively using Verlet integration, Euler integration (implicit or explicit) or Leapfrog integration...etc.
Because Molecular Dynamics is a simulation, then it's subject to the laws of simulaton. One example is the fact that MD fidelity, speed and numerical stability depend on the simulation components (timestep, timespan, simulation size, simulation box...etc).
.

3.1 General description

In any numerical simulation, time is treated discretely, that is, it is divided into a series of distinct time intervals.

The objective is to determine the position and the speed of each particle at each time interval, based on the data obtained at the previous instant.

This involves applying Newton's second law to each particle, formulated as follows:

$\sum_i \overrightarrow{\mathrm{F}}_i = m\vec{a}$.

This can be done either statistically via averaging out the details of each atom/molecule in a probability distribution or statistical ensemble (statistical mechanics) or, in our case, numerically by iteratively solving Newton's second law using Verlet integration.

3.2 Boundary conditions

The system, which is the subject of the study in molecular dynamics, generally takes the form of a three-dimensional parallelepipedal "simulation box" with determined dimensions.
During the simulation, the boundary conditions are periodic, that is to say, if a particle leaves the simulation box by one of its faces, it will be reintroduced into the box by the opposite face with the same speed.

3.3 Force Fields (or interatomic potential)

In molecular dynamics, to calculate the interaction forces between the particles in a studied system, it is necessary to use a potential function which describes the interaction between the particles. This potential function depends on the positions of the particles and can be calculated for each pair of particles in the system.
The most commonly used interaction potentials in molecular dynamics are the Lennard-Jones potential and the Coulomb potential. The Lennard-Jones potential describes the short-range interaction between particles and is given by:

$$V(r) = 4\varepsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right] \tag{1}$$

r: the distance between the particles.
σ: the distance at which the potential energy is minimum.
ε: the depth of the energy sink and determines the force of attraction between the particles.
The Coulomb potential describes the long-range interaction between particles carrying electrical charges and is given by:

$$V(r) = \frac{q_1 q_2}{\varepsilon_0 \pi r} \tag{2}$$

q_1 and q_2: the charges of the two interacting particles.
ε_0: the vacuum permittivity and r is the distance between the particles.

Once the potential functions have been determined, the interaction forces can be calculated from the gradients of these functions. These forces are then used to solve the equations of the motion of the particles in the system, thus making it possible to simulate their motion over time.

3.4 Molecular visualization using VMD

VMD (Visual Molecular Dynamics) is a powerful and widely used software tool for visualizing molecular dynamics simulations. It is designed to handle large biomolecular systems and provides a wide range of features for analyzing and visualizing the results of molecular dynamics simulations.

VMD allows users to visualize the 3D structure and motion of biological molecules, including proteins, nucleic acids, and lipids, and can display various properties of the system, such as electrostatic potentials, hydrogen bonding patterns, and secondary structure. Users can manipulate the visualized structures in real-time, such as rotating, zooming, and translating the structures, or adjusting the color and transparency of individual atoms or residues.

VMD also provides a range of analysis tools, including measurements of distances, angles, and torsion angles, as well as visualization of trajectories, RMSD (root mean square deviation), and RMSF (root mean square fluctuation) plots, which are commonly used to analyze the stability and dynamics

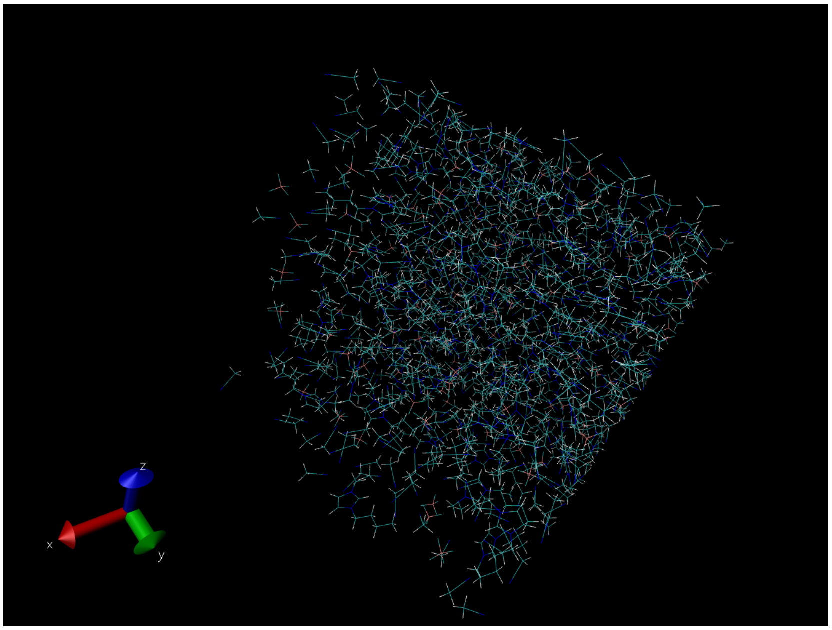

Figure 1: Visualization of molecules

of biomolecules.

The visualization capabilities of VMD make it an essential tool in the field of molecular dynamics, as it enables researchers to better understand the behavior of biological molecules and how they interact with each other and their environment. The ability to visualize complex systems and their dynamics in real-time is critical for researchers to gain insights into the behavior of biological systems and to design experiments and simulations that can provide meaningful results. Thus, VMD is an extremely useful tool for researchers in molecular biology, biochemistry, biophysics, and other related fields.

4 Simulation components

4.1 Timestep

In MD simulations, the value of the timestep is as crucial as the other simulation components. Usually, the timestep size is of the order of 1 femtosecond (10^{-15} s) as it must be smaller than the period of the fastest vibrating atom/molecule in the system in order to avoid discretization errors.
the timestep size can be increased via SHAKE constraint algorithm which is based on fixing the fastest vibrating atom.

4.2 Timespan

The timespan or total duration of the simulation is the timescale of the simulation and shouldn't be confused with the running time of the simulation though they tend to correlate. The timespan should be large enough in order to be relevant and numerically valid by matching the kinetics of the natural processes.
In MD simulations, the timespan is usually of the order of microsecond (10^{-6} s) to nanosecond (10^{-9} s) which is equivalent to CPU-days to CPU-years.

4.3 Simulation Size

In the context of MD simulations, simulation size is the number of particles of the system. Simulation size dictates the computational methods used; explicit or implicit. Naturally, in the case of short systems, explicit methods are preferred while implicit methods (usually those based on mean field theories) are more compatible with large systems.

4.4 Simulation Box

The simulation box is inside of which the simulation takes place; the extent to which the molecules move. Simulation box affects the convergence rate; it must be large enough to be representative of the degrees of freedom of the system but not too large for a convergence criterion.
In MD simulations, the simulation box size is usually intermediate for convergence purposes. Moreover, simulation boxes are either reflective (Von Neumann boundary conditions) or absorbing (Dirichlet boundary conditions) or periodic. For instance, here is a visualization of the ACN-[BMI][BF4] electrolyte within the simulation box (45A size) using VMD software:

5 Simulation steps

Molecular Dynamics is performed following a number of specific steps which are:

5.1 Initialization

it can be random Gaussians for velocities and/or coordinates. This process might be followed by ”packing” all atoms in one corner (thanks to Packmol software) to avoid perturbation that may result from short-range repulsions (Pauli exclusion principle)

5.2 Potential evaluation (interatomic potential or force field):

This is the most CPU intensive step due to the evaluation of non-bonded (pairstyle) potential ie; Couloumb potential and Van Der Wals force. This is due to the fact that although intermolecular forces are weaker than intramolecular forces (which is the reason why we have molecules), intermolecular forces are non local which implies that their influence pervades all the system (worst-case scenario of a complexity of O(n**2)).

5.3 Energy minimization/equilibration:

Energy equilibration is rendering kinetic energy equally distributed among constituent particles (this can be done using methods like dynamically-updated velocity rescaling). Energy minimization is the process of relaxing the system to its stable state in order to render the simulation more accurate and more representative of reality:

5.4 LAMMPS run

Predict the motion by updating the values of coordinates and velocities by integrating Newton’s equations (this is basically what LAMMPS ”run” commands does)

5.5 Data collection (Production period):

This is the step after the simulation terminates, collecting average values of parameters of interest (diffusion, temperature..etc)

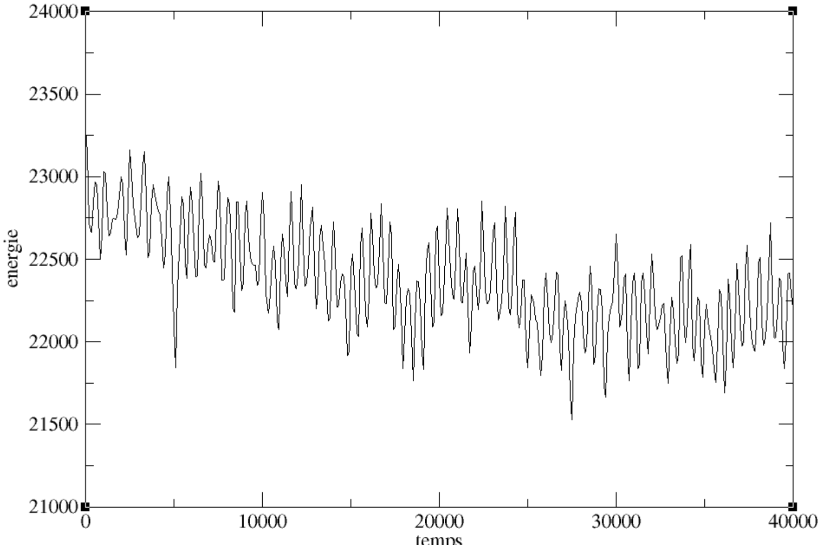

Figure 2: Energy Minimization:300K

5.6 Data Analysis:

This is a very important step (contrary to popular believes) because it's the step where one has to analyze the average values obtained from the simulation to see whether the simulation is accurate, converges or stable. If the simulation is fine, then it's a good opportunity to virtually investigate the system accurately without the need to experimentations (in the case of experiments being dangerous, time consuming or merely impossible). Data analysis is the step where it is possible to infer conclusions about atom-level phenomena that are not directly observed or purely virtual (alternative events that didn't happen in the physical world..etc)

6 Convergence criteria:

For energy equilibration phase (NPT ensemble), termination criteria are the desired values of thermodynamic properties like temperature, pressure, density...etc that are consistent with thermodynamic equilibrium state of the system.

For energy minimization phase, termination criteria are : energy tolerance (etol), force tolerance (ftol), maximum number of outer iterations (maxiter) and maximum number of energy/force evaluations (maxeval). This also includes the convergence criteria of the minimization algorithms specified by minstyle command (cg, sg, hltn...etc) and also the algorithm parameters specified by minmodify command (line value: forcezero, quadratic.. integrator: verlet, leapfrog.., dmax value..etc).

For production phase (NVT ensemble), termination criteria are the total number of steps in run command.

The matching between statistical ensembles and phases is not a clear-cut decision. It depends on the system in question and the thermodynamic properties of interest in MD simulation (it's not black and white).

7 Diffusion coefficient

Diffusion is the irreversible phenomenon of particles movement en mass from region of high concentration to reigions with low concentration following Fick's law.

Fick's first law is a phenomenological law (just like its counterpart Fourier's law) meaning that it's not derived from theory nor it's universal, but it's derived from observation in a way that's nonetheless coherent with theory. Fick's first law indicates the relationship between the diffusion flux and concentration gradient as follow:

$$J = -D \frac{\partial \varphi}{\partial x} \tag{3}$$

Fick's first law is a form of a moderation law (in the sense of control theory) because it's a regulation of a perturbation (the gradient of the concentration ie; the heterogeneity of concentration in space)

Fick's second law, just like the first law, is describing the same phenomenon but in a way that provides a fundamental insight about the phenomenon of diffusion; namely its relationship with the second law of thermodynamics. Fick's second law is stated as follow:

$$\frac{\partial \varphi}{\partial t} = D \frac{\partial 2\varphi}{\partial x^2} \tag{4}$$

Fick's second law is similar to the heat equation. Both are characterized by a time-reversal asymmetry (unlike D'alembert wave propagation equation) because of the first derivative with respect to time which indicates the irreversible nature of the diffusion phenomenon. Such a time-reversal asymmetry is due to the second law of thermodynamics which states that in closed systems, entropy never decreases and only increases (thus engendering the arrow of time; a hypothesis under the name of thermal time hypothesis put forth by Carlo Rovelli and Alain Connes)
Given the time-reversal asymmetric nature of diffusion equation, it's said not to conserver information. This is known as Noether's theorem (symmetries in physics always correspond to a certain law of conservation; in the case of time-reversal symmetry, it's the information in the sense of Shannon)

From the point of view of simulation and mathematical modeling, diffusion can be modeled using random walks or Levy flights or Markov chains with the assumption of ergodicity. Because diffusion is a form of homogenization (recall the moderation law) therefore it's by definition an ergodic phenomenon (the second law of thermodynamics is a physical reformulation of the ergodic hypothesis first put forth by Ludwig Boltzmann)

8 Discussion of results:

The results of the simulation indicate that the total energy, diffusion coefficient and volume increase with temperature. The following figures are illustrative representations of the results of the simulation:

For the diffusion coefficient, the simulation results match the reality in encapsulating the relationship between diffusion coefficient and temperature. To recall, Einstein's relation (Stokes-Einstein-Smoluchowski equation) states that diffusion coefficient D is proportional to temperature T with the proportionality constant being the particles mobility (intrinsic to the molecules) times Boltzmann constant:

$$D = \mu k_B T \tag{5}$$

Albeit neglecting quantum effects playing out at the level of hydrogen bonds thanks to the Born-Oppenheimer approximation, we still managed to get results that reflect empirical reality. Therefore, this is an indication that the simulation is of high-fidelity which in turns implies that the model used (Newtonian dynamics) is accurate and representative of microscopic phenomena up to a certain threshold (dictated by the complementarity principle of Niels Bohr) beyond which the B-O approximation breaks down.

On a more fundamental level, the results of the simulation, guaranteeing high fidelity, reflect something deeper about the nature of the universe captured by one of the most famous concepts in statistical mechanics which is the fluctuation-dissipation theorem which is a phenomenological result from the field of statistical physics that is "fundamental" meaning that it occurs in different areas of physics (light absorption-heat, Johnson noise-Resistors, Diffusion-Joule Energy) due to the fact that it doesn't assume any natural laws (black-box) about the system it studies, resulting only from the a purely mathematical (statistical, to be precise) examination of the system. The theorem states that at equilibrium, any system in a detailed balance (stationary distribution of reversible Markov chain) such that for each elementary process there exists a reverse process that balances it, a process that causes energy dissipation (captured by abstracting the notion of impedance or admittance from electricity) by turning it into heat, is balanced by a reverse process in the form of thermal fluctuation which converts that heat back to the original form of energy (usually, kinetic energy).

In our example, incorporating solvent effects and using "fix viscous" LAMMPS command to fix the damp factor value (= 100), that caused the particles to dissipate their kinetic energy due to friction and viscosity, that in turn led to thermal fluctuations manifested in an increasing kinetic energy (Brownian motion) captured by an increasing diffusion coefficient.

9 Interesting applications of MD simulation in Optimization theory and Operations Research:

9.1 Demon Algorithm

This is a Monte Carlo method (or a metaheuristic to be more precise) used to "solve" the system in non-equilibrium statistical mechanics or when the system size is too large (a lot of number of particles) to the point where brute-force exhaustive (explicit) search becomes computationally expensive

The method assumes ergodicity (or at least quasi-ergodicity) for efficiently sampling from the micro-canonical ensemble NVE with an additional degree of freedom called "demon" (intended here to be the reversed version of Maxwell's demon) that exchanges energy with the system to reach equilibrium while maintaining constant energy (energy conservation) without interacting with particles.

This method is inspired from MD simulations in the attempt to solve the system implicitly (more precisely, stochastically) instead of expensive calculations of velocity and position for every particle, by defining the microscopic variables that are consistent with macroscopic variables (note that there are many microstate configurations that can result in the same macrostate thus the complexity of the problem; Multiple realizability)

The point of the algorithm is to lower the potential energy E in order to reach thermodynamic equilibrium while conserving total energy by storing lost E values in the demon energy Ed. In optimization problems, the potential energy E is the function we want to minimize. Aside from conserving total energy, Ed Demon's energy plays the role of an exploration catalyzer/tolerance, the greater its value the more incentive/tendency there is to explore the search space. For instance, dE ¡ 0 = exploitation phase. Also, the thing about the Demon algorithm is that the exploitation-exploration trade-off is balanced via a feedback loop via having exploiting a solution adds up to Ed. Demon algorithm can be used to optimize non-linear functions with multimodal features.

9.2 Particle Swarm Optimization

This metaheuristic is based on the notion of swarm intelligence defined as the collectively emergent behavior of a decentralized, self-organized system (swarm) of agents following simple rules (position, velocity) interacting with each others until a global intelligence behavior emerge (similar to individual-based models). Examples include: Ant colonies, Bee colonies, bird flock, fish schooling, social behavior...etc.

The idea behind PSO is simple; implementing a hybrid (collective and individual) exploration where

individual agents (particles) move according to simple rules dictated by a relation between position and velocity (Verlet integration) guided by a local best-known position in the search space and the global best-known position for the whole system that are discovered. The next step is to perform a movement en mass towards one of those position (depending on strength of social coefficient vs cognitive coefficient)
PSO is efficient at balancing between exploration and exploitation manifested in the value of vi,max for each particle (or vmax for all particles) such that high vmax encourages exploration and low vmax encourages exploitation of the path found to the solution. Of course, an adaptive strategy has to be performed in order to come up with the efficient value of vmax depending on the nature of the system (number of particles initialized, social/cognitive coefficients...etc)
There is a similarity between this metaheuristic and molecular dynamics simulation because they can be expressed in almost the same notions (particles, positions, velocities, energy minimization...etc) thus they are structurally the same implying that MD simulation can indeed be used to solve optimization problems involving non-linear, multimodal or non-convex objective functions
We wrote a modified version of PSO that is related to statistical physics (and thus to MD simulations) based on Maxwell-Boltzmann distribution instead of uniform or normal distribution for sampling velocities in the algorithm (original version) and it yielded the same results as the original which reflect physical reality (M-B distribution of velocity).

10 Conclusion:

In conclusion, the temperature study of the diffusion phenomenon in the ACN-[BMI][BF4] electrolyte has made it possible to observe the influence of temperature on the diffusion coefficient. The results show an increase in the diffusion coefficient with temperature, which highlights the importance of considering temperature in the design and modeling of electrochemical systems. This study also made it possible to set up a method to study the impact of temperature on diffusion in other electrolytes. These results could contribute to improve the understanding of electrochemical phenomena and to optimize the performance of electrochemical systems in many fields of application, such as energy storage, catalysis and green chemistry.
However, it's worth mentioning that whilst simulation is a strong tool to examine physical systems, it only comes at the cost of physical correspondence. In other words, simulation is not duplication; one is a physical reality, the other is merely a series of flipping 1s and 0s on the level of CPU. Nevertheless, simulations can be useful when used as post-theoretic examination of physical systems that are often hard and/or expensive to simulate or when the technology is not there yet to study the system empirically.

11 Figures/Results

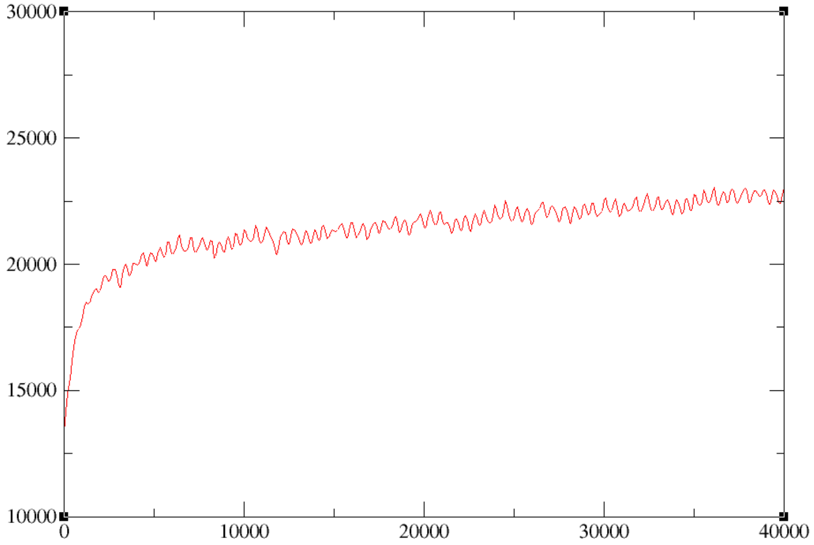

Figure 3: Energy at T=600K

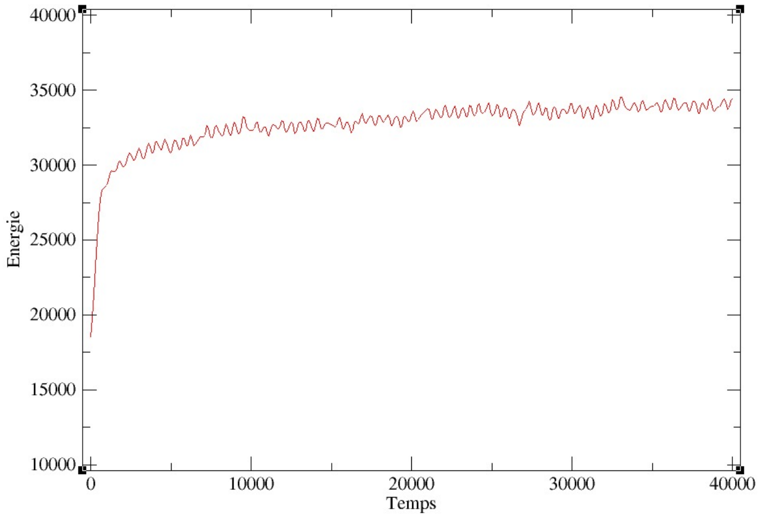

Figure 4: Energy at T=1200K

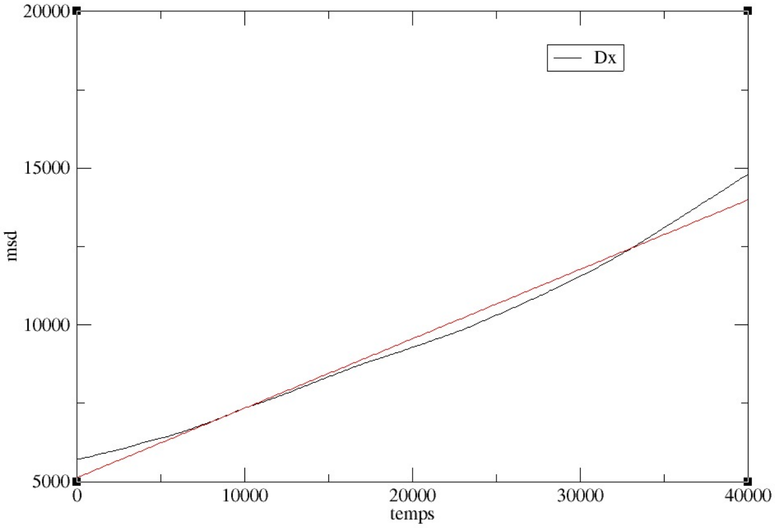

Figure 5: Diffusion coefficient along x axis

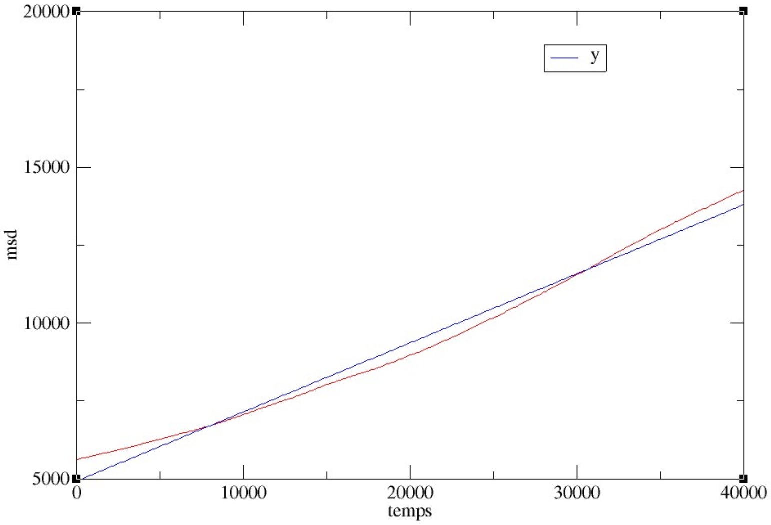

Figure 6: Diffusion coefficient along y axis

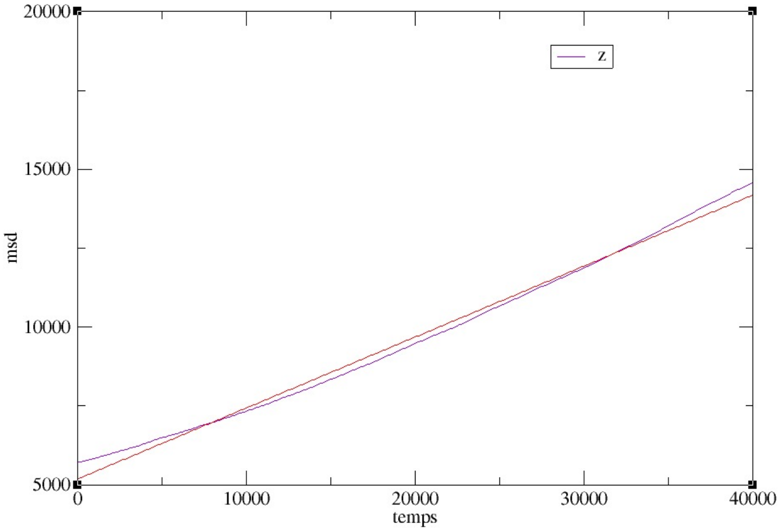

Figure 7: Diffusion coefficient along z axis

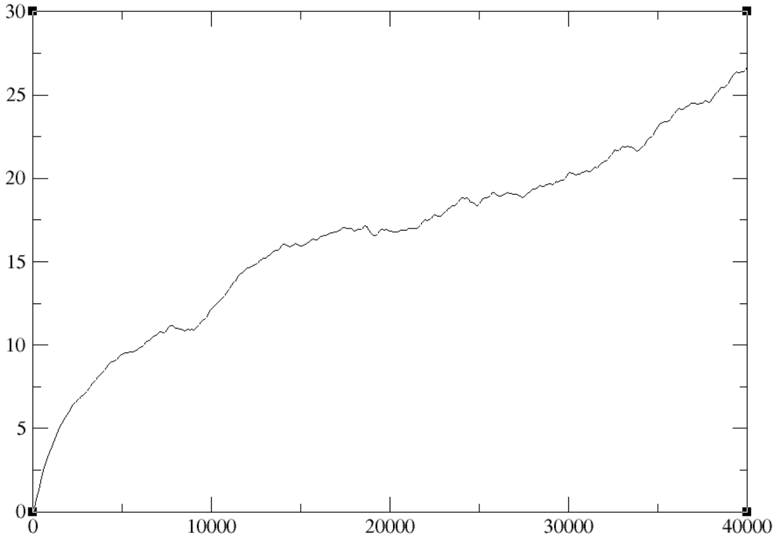

Figure 8: Diffusion coefficient at T = 300K

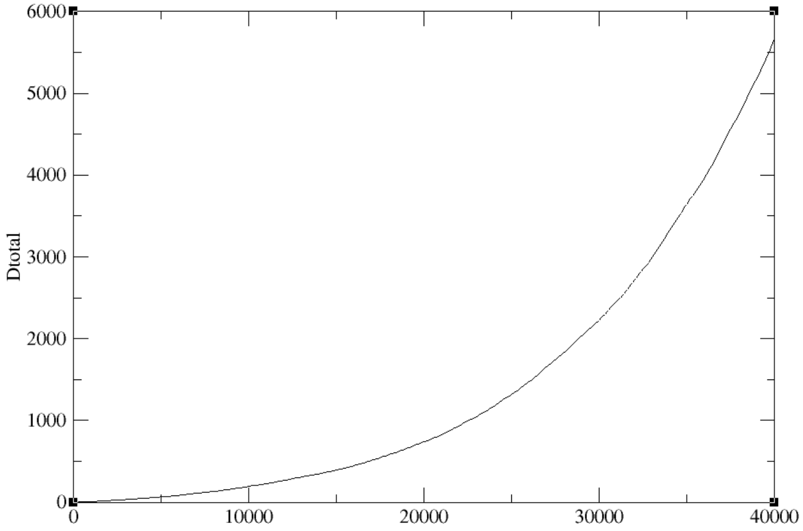

Figure 9: Diffusion coefficient at T = 600K

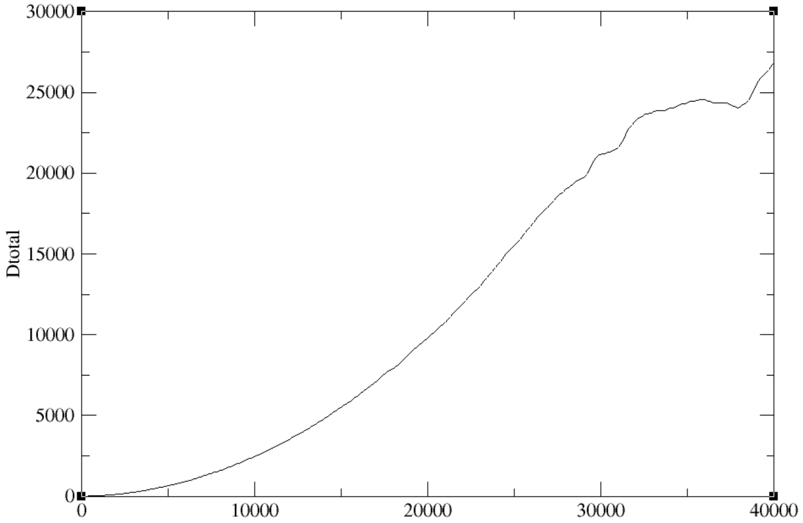

Figure 10: Diffusion coefficient at T = 1200K

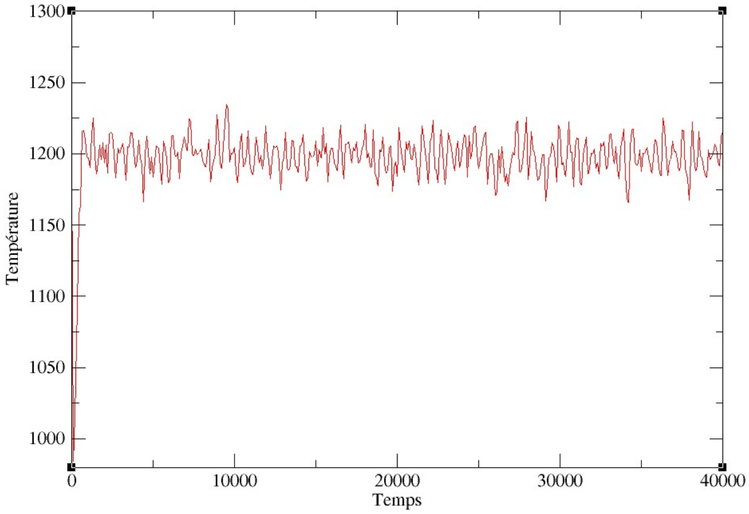

Figure 11: temperature fluctuation

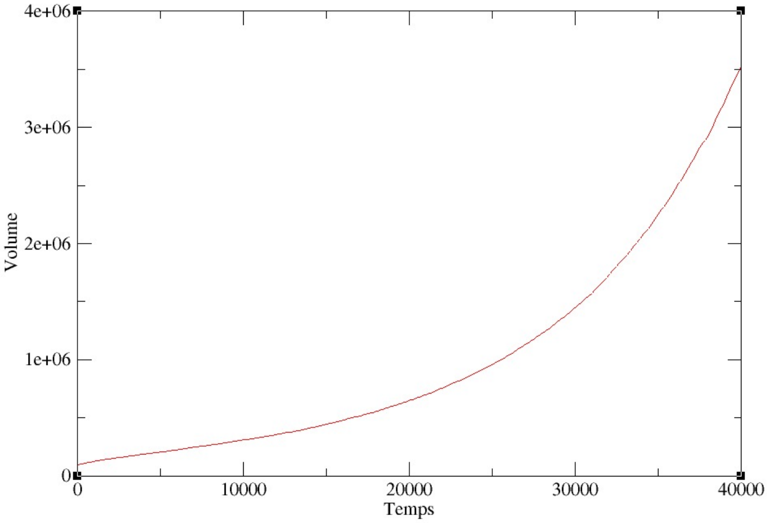

Figure 12: Volume at T = 1200K