

Numerical simulation of particles driven by a Lennard-Jones potential and ideas for modeling ferrofluids in a Hele-Shaw cell

Lucas CARON¹ Paul SIMON¹

¹Université Paris-Saclay, Magistère de physique fondamentale d'Orsay (91400)

Abstract

This article describes some simulations of particles interacting through an Lennard-Jones potential, in two dimensions, in order to obtain the behaviour of a fluid. Initially, only one type of particle is simulated, then two. Finally, an attempt is made to simulate magnetic interactions within these particles, with the aim of recovering the behaviour of ferrofluids in a Hele Shaw cell.

1 Introduction

The study of molecular dynamics by non-bounded interactions using a soft-disk fluid simulation has existed for many years. Indeed, such a method allows the connection between mesoscopic and macroscopic theories. However, molecular dynamics suffers from a fundamental problem: the computational power required to reach macroscopic scales is tremendous. However, it may not be necessary to scale up to the same levels in order to have a coherent behaviour, in particular if we consider the isotropy of a perfect fluid, that allows us to use periodic conditions. For example, it has been shown that the simulation of 864 particles interacting through a two body potential can be a good approximation of the equilibrium state of argon [1]. The objective of this paper is to verify whether such a method is applicable to simulate a ferrofluid deposited in an propanol solvent under a Hele-Shaw cell, in particular to try to study its behaviour when exposed to a magnetic field. In Sec. 2, we will develop theoretical background needed for the simulations. In sec 3, we will explain the general method in order to build our simulation. We will then give, in Sec. 4, the details of the different simulations computed and their coherence with the expected properties. Sec 4.1 4.2 will be dedicated to the simulation of one type of particle. Sec 4.3 4.4 will be about two types of particles. Finally, sec 5 will resume our different approaches to model a magnetic fluid.

2 Theoretical background

2.1 Lennard-Jones potential

In the simulations, the interaction forces between two molecules i and j are modeled by a Lennard-

Jones potential defined as:

$$V_{ij}(r_{ij}) = 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \quad (1)$$

Where σ_{ij} is a characteristic distance, r_{ij} the distance between the particles, ϵ_{ij} a constant (dimension of an energy) depending on the molecule studied. The attractive term represent the Van der Waals interaction, and the repulsive term, dominant in short range, is empirical: it expresses the Pauli exclusion principle, preventing the interpenetration of the electronic clouds of the 2 molecules. A graph of the Lennard-Jones potential is given by the Figure 2.1.1.

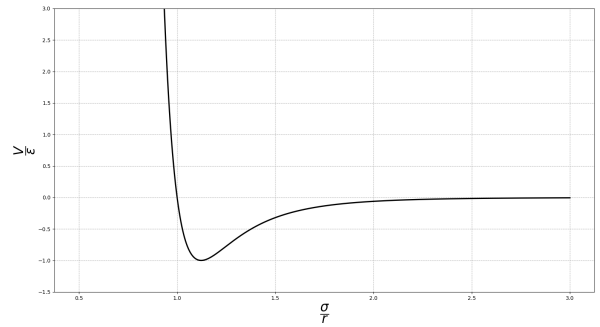


Figure 2.1.1: Lennard-Jones potential.

2.2 Equations of motion

If we only consider this interaction, the problem consists in integrating the equation of motion

$$m \frac{d^2 \vec{r}_i}{dt^2} = \sum_{i \neq j} \vec{f}(r_{ij}) \quad (2)$$

With $\vec{f}(r)$ written as:

$$\vec{f}(r_{ij}) = 48\epsilon_{ij} \left[\frac{\sigma_{ij}^{12}}{r_{ij}^{13}} - \frac{1}{2} \frac{\sigma_{ij}^6}{r_{ij}^7} \right] \vec{u}_r \quad (3)$$

The total energy of the system with N particles is written as follows:

$$E = \frac{1}{2}m \sum_i v_i^2 + 4\epsilon_{ij} \sum_{i<j}^N \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \quad (4)$$

By considering the above conditions, the total energy should be constant. It will therefore be a good indicator of the consistency of the simulation.

2.3 Störmer-Verlet Integration

The integration of the equation (2) can be done using the Störmer-Verlet integration[1]:

$$\begin{aligned} \vec{v}_i(t + \Delta t) &= \vec{v}_i(t) + \Delta t \vec{a}_i(t) \\ \vec{r}_i(t + 2\Delta t) &= \vec{r}_i(t) + 2\Delta t \vec{v}_i(t + \Delta t) \\ \vec{a}_i(t + 2\Delta t) &= \sum_{i \neq j} \vec{f}(\vec{r}_{ij}(t + 2\Delta t)) \\ \vec{v}_i(t + 2\Delta t) &= \vec{v}_i(t + \Delta t) + \Delta t \vec{a}_i(t + 2\Delta t) \end{aligned}$$

With Δt the time step.

This method is used because it is simple and has a rather good energy conservation, integrating the coordinates and velocities of the particles.

3 General approach to simulations

3.1 Limits of the LJ potential

Lennard-Jones potential is effective within a relatively short distances. It is therefore appropriate to establish a critical distance r_c , from which it will be considered null.

3.2 Periodical conditions

A single droplet requires way too many particles to be simulated. Thus, only a very small area, a square of size L is simulated, to which we apply periodic conditions: e.g. a particle leaving the top will appear again at the bottom. In this way, we will consider the whole system as the repetition of this small square. This can be done because the interactions considered are only over small areas.

3.3 Division of the simulated area

To limit the amount of calculation needed, we need to restrict the number of particle interaction to the ones that are under the limit r_c .

We divide the square of length L in smaller square, of length $l_c > r_c$ that we call cell. Every particle i is associated to the cell it is in. Thus, the search for

particles such as $r < r_c$ will be done only in the cell of the considered particle and in the eight adjacent cells, which will considerably decrease the computation time (we do not test all possible particles).

3.4 Verlet neighbour method

An alternative, more optimised method of calculating neighbours has been developed by Verlet. It uses a list of neighbour pairs of particles, that is regenerated regularly. To know when to regenerate it, we consider a neighbourhood distance r_v greater than the critical distance r_c , such as $r_v = r_c + \Delta r$. Δr is chosen such that several time steps Δt can occur without the particles initially further away than r_c entering the interaction radius. The regeneration of the neighbour list is done when $\Delta d > \Delta r$ with $\Delta d = \Delta d + 2v_m \Delta t$, where v_m is the maximal speed of the sphere, computed for each time step. We will see later a comparison of the computation time using the method with the cells or the list of neighbours

4 Details on program code

4.1 Simulation of one unique type of particle

For a single type of particle, the Lennard-Jones coefficients σ and ϵ are constant. To make the program easier to run, we consider the equations (2) and (3) as adimensional, with $m = \epsilon = \sigma = 1$. The initialization is achieved by distributing in a unified way the whole particles N in a square of side L . All these particles are given a random speed direction, whose norm is set by the user. Then the integration presented in section 2.3 is performed by calculating the force for each particle using the cells method (Sec 3.3) or the Verlet neighbours list method (Sec. 3.4).

4.2 Result and analysis

The visual appearance is similar for both methods of calculating neighbours.

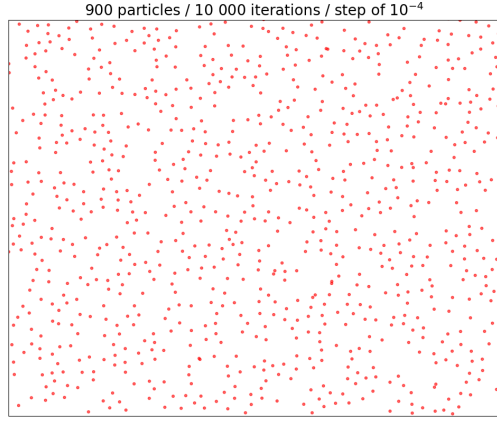


Figure 4.2.1: Simulation after 10000 steps (parameters: $N = 900$, $r_c = 2.5$, $\Delta t = 10^{-4}$)

4.2.1 Total energy analysis

As said before, one way to verify if this simulation is consistent is to look at the total energy of the system, which is supposed to stay constant. 3 factors here are crucial : the length of the time step Δt , the number of steps and the density ρ of particles. The higher they are, the more likely the simulation will become obsolete.

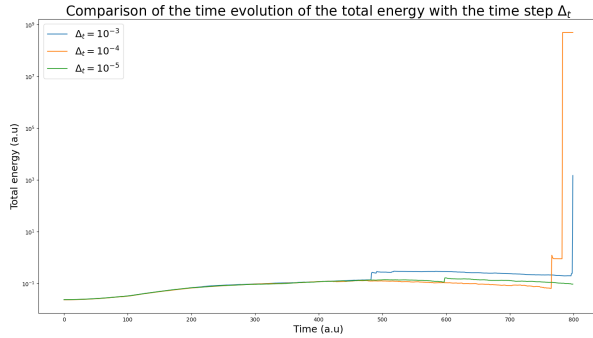


Figure 4.2.2: Total energy of the same simulation with different time step Δt

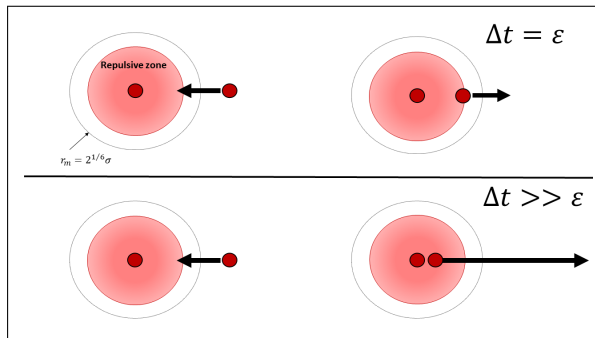


Figure 4.2.3: Explanation diagram

If the time step is too large, there is a possibility that two particles get closer than they should, resulting

in a faster than expected repulsion speed. Even with a really small time step, there is still a chance that a particle with a high speed get closer than it should, but it will happen later than with a larger time step. This kind of phenomenon leads to a divergence of the velocities of the particles. We have not found a satisfying way to avoid it. When the divergence occurs, our program will fail. We have nevertheless put the possibility to give a threshold force (*cutoff* in the program) which will delay the phenomenon. The ideal is to carry out the simulations without cut-off, using a very small time step.

4.2.2 Cells vs. Neighbours list

We have seen that it is necessary to perform a large number of iterations to avoid divergence, with the smallest possible time step. Therefore, it is crucial to compare the computing time of the two methods used.

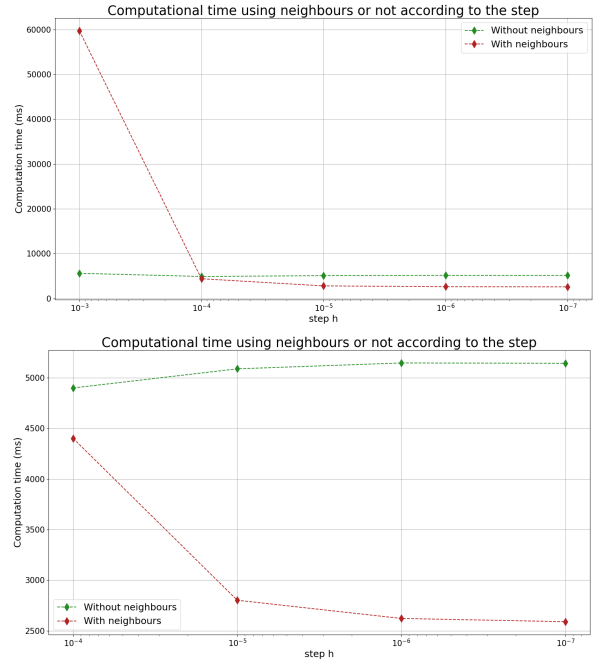


Figure 4.2.4: Comparison of the calculation time between the default and the Verlet neighbours method. (parameters: $N = 1000$, $r_c = 2.5$, $\Delta r = 0.5$)

The number of particles in the neighbour cells is approximately the same at any time. This is why the green line is constant. If the time step is small enough, the Verlet neighbours list method becomes way more efficient. For more precise simulations, it would be the preferred method. However, this method was realized late in the project and the rest of the code was implemented using the neighbouring cells method.

4.2.3 Physics characteristics: velocity distribution

Considering the data obtained by the simulation, it is possible to compute pressure, temperature, and others physics aspects of the dynamical interactions of molecules. We have implemented the functions allowing to study these properties but unfortunately we did not have time to study them more deeply. We only looked at the speed distribution:

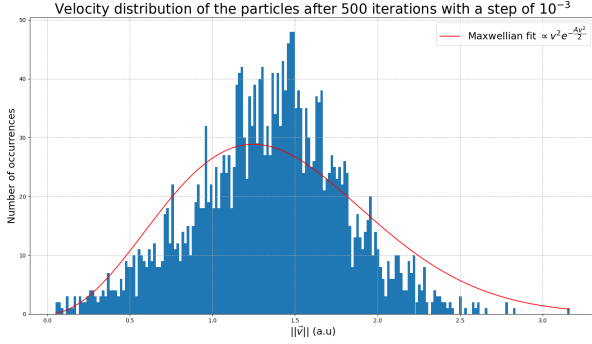


Figure 4.2.5: Velocity distribution, Maxwellian fit

A similar result to a distribution of Maxwell-Boltzmann is expected, considering that the simulation is following the kinetic theory of gases. However, this distribution occurs at the thermodynamic equilibrium. The initial conditions lead to a large variation of the temperature at the beginning of the simulation. To adjust our model to a specific temperature, an adjustment of the velocities is required [2]. This is what we did by renormalizing the velocities at each step, to keep a constant temperature (function *adjust_v*). We obtain the following plot:

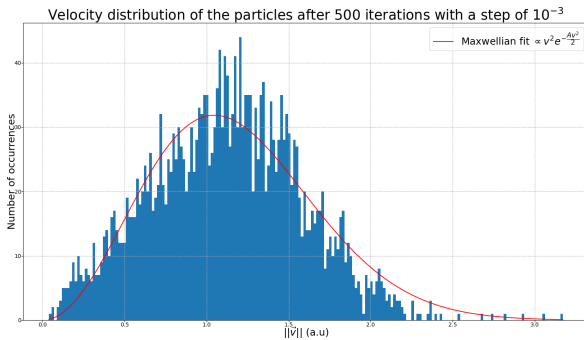


Figure 4.2.6: Velocity distribution with corrections, Maxwellian fit

Once the equilibrium is reached, the distribution is better fitted with a Maxwellian curve. Others statistical calculations can be performed.

4.3 Simulation of two types of particles

To get closer to the ferrofluid situation, we simulate 2 types of particles, which means three different interactions: the two between particles of the same type (with different constants) and the one between the two types of particles. All the constants values are written in the following tabulation

	$\epsilon (cm^{-1})$	$\sigma (\text{\AA})$
Propanol[3]	460	4.5
Hydrocarbon oil[4]	150	8.0
Interaction between them	262	6.4

Remark: To find the constants of interactions between molecules of different types, we used the combination rule of Lorentz and Berthelot:

$$\sigma_{12} = \sqrt{\sigma_{11} \sigma_{22}} \quad \text{and} \quad \epsilon_{12} = \frac{\epsilon_{11} + \epsilon_{22}}{2}$$

4.4 Results

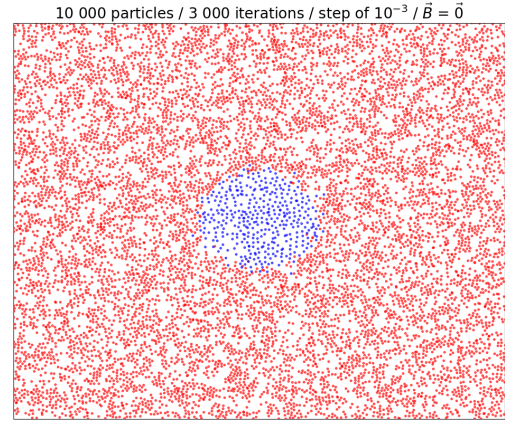


Figure 4.4.1: Simulation after 3000 steps (parameters: $N = 10000$, $r_c = 2.5$, $\Delta t = 10^{-3}$). In blue, the ferrofluids particles, in red the propane particles

A large number of molecules are needed to form a ferrofluid drop in the centre of the simulation. Our available computing power is not very high: this large number of particles limits the number of iterations. Nevertheless, we observe that the drop shape is maintained. This is what is observed experimentally. It would not necessarily be the case in another solvent.

5 Attempt to simulate a ferrofluid

5.1 Implementation of the magnetic force: hypothesis

We implement the magnetic interaction in the simulation. Our idea to realize a first model of a ferrofluid placed in a solvent in a Hele-Shaw cell (thus in two dimensions) was very simplistic. We consider a constant external magnetic field, very intense and oriented perpendicular to the 2D plane of the problem. Thus, we suggest that all the magnetic moments of the magnetic particles are oriented in the direction of the magnetic field. Then, at each step, in addition to computing the forces related to the Lennard-Jones potential, we calculate the force related to the interaction between the magnetic moments [5]:

$$\begin{aligned}\vec{F}(\vec{r}, \vec{\mu}_1, \vec{\mu}_2) &= \vec{\nabla}(\vec{\mu}_2 \cdot \vec{B}_1) \\ &= \frac{3\mu_0}{4\pi|\vec{r}|^4} \left[(\vec{\mu}_1 \cdot \vec{u}_r) \vec{\mu}_2 + (\vec{\mu}_2 \cdot \vec{u}_r) \vec{\mu}_1 \right. \\ &\quad \left. + (\vec{\mu}_1 \cdot \vec{\mu}_2) \vec{u}_r - 5(\vec{\mu}_1 \cdot \vec{u}_r)(\vec{\mu}_2 \cdot \vec{u}_r) \vec{u}_r \right] \\ &= \frac{3\mu_0\mu^2}{4\pi|\vec{r}|^4} \vec{e}_r\end{aligned}$$

with $\vec{\mu}_1 = \vec{\mu}_2 = \mu \vec{e}_z$

Since the patterns formed by ferrofluids are realized on large scales, we decided to compute the magnetic interactions between all magnetic particles (which increases the computation time).

5.2 Result of the attempts

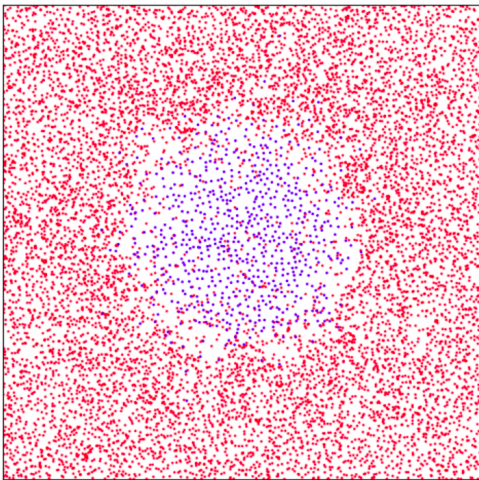


Figure 5.2.1: Magnetic simulation after 100 000 steps. (parameters: $N=10000$, $r_c=2.5$, $\Delta t = 10^{-4}$). In purple the magnetic particles, in red the solvent

The figure 5.1.1 is the result of a long computing time. The global behaviour is that the droplet of magnetic particles keeps spreading. This opening on ferrofluids is far from complete. We did not have the time to perform many simulations and therefore the implemented expression of the force remains to be adapted. Indeed, we would have to find a multiplicative coefficient that would allow to adapt to the arbitrary units taken for the Lennard Jones potential. Moreover, to make relevant simulations, we would need more computing power and more time, working on super-computers for example.

6 Complexity

We have briefly studied the complexity of the simplest program as a function of the number of particles, and then the program with the consideration of the magnetic interaction as a function of the proportion of magnetic particles.

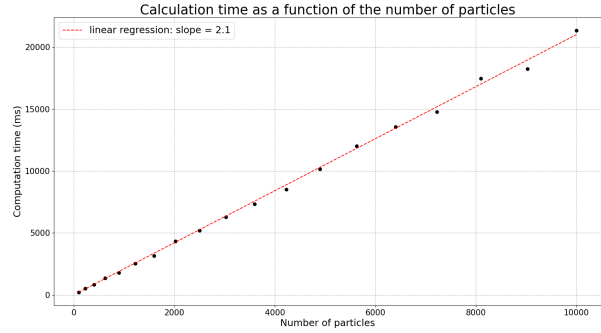


Figure 6.0.1: Complexity of the simplest program as a function of N

Thus, we can conclude that the computation time is linear with the total number of particles: the complexity is in $\mathcal{O}(N)$.

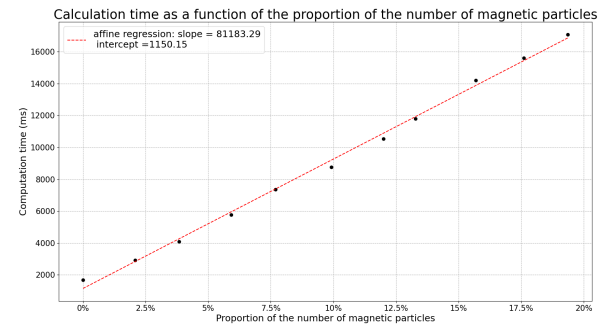


Figure 6.0.2: Complexity of the simplest program as a function of N

The same behavior is observed when we add the magnetic interaction, this time depending on the proportion of magnetic particles. The complexity is

in $\mathcal{O}(\frac{N_{mag}}{N})$.

Thus we understand that our program is very greedy in computing time.

7 Conclusions and perspectives

Molecular dynamics simulations based on the Lennard-Jones potential have been developed. The simplest one could be used to study different characteristics of this kind of physics systems. For the second simulation, an interesting work could be to look for a way of obtaining the surface tension and compare it to the experimental one. The last simulation, ambitious, did not meet its expectations: the results were not the same as experienced, and there are many hypothesis to explain this incoherence: the effectiveness of the hypothesis considered, the overall small time simulated which could not give enough time to see the global magnetic behaviour... Still, all these simulations are useful, but need a

high computing power to be used efficiently. On a personal note, even though we didn't achieve our goal as well as we had hoped, we really made progress during this project. Learning object-oriented programming in C++ by doing a project is, according to us, one of the best methods, if not the best, to progress.

8 Contributions

Even if we have thought together about its realization, Lucas CARON was the person who mostly developed the program code. Paul SIMON focused more on the theoretical work and the redaction of the article. The whole work has been a common collaboration, notably made in order to go further into a subject realized by both of us at the French Physicists' Tournament [6], and we are both happy about the achievement done.

Appendix

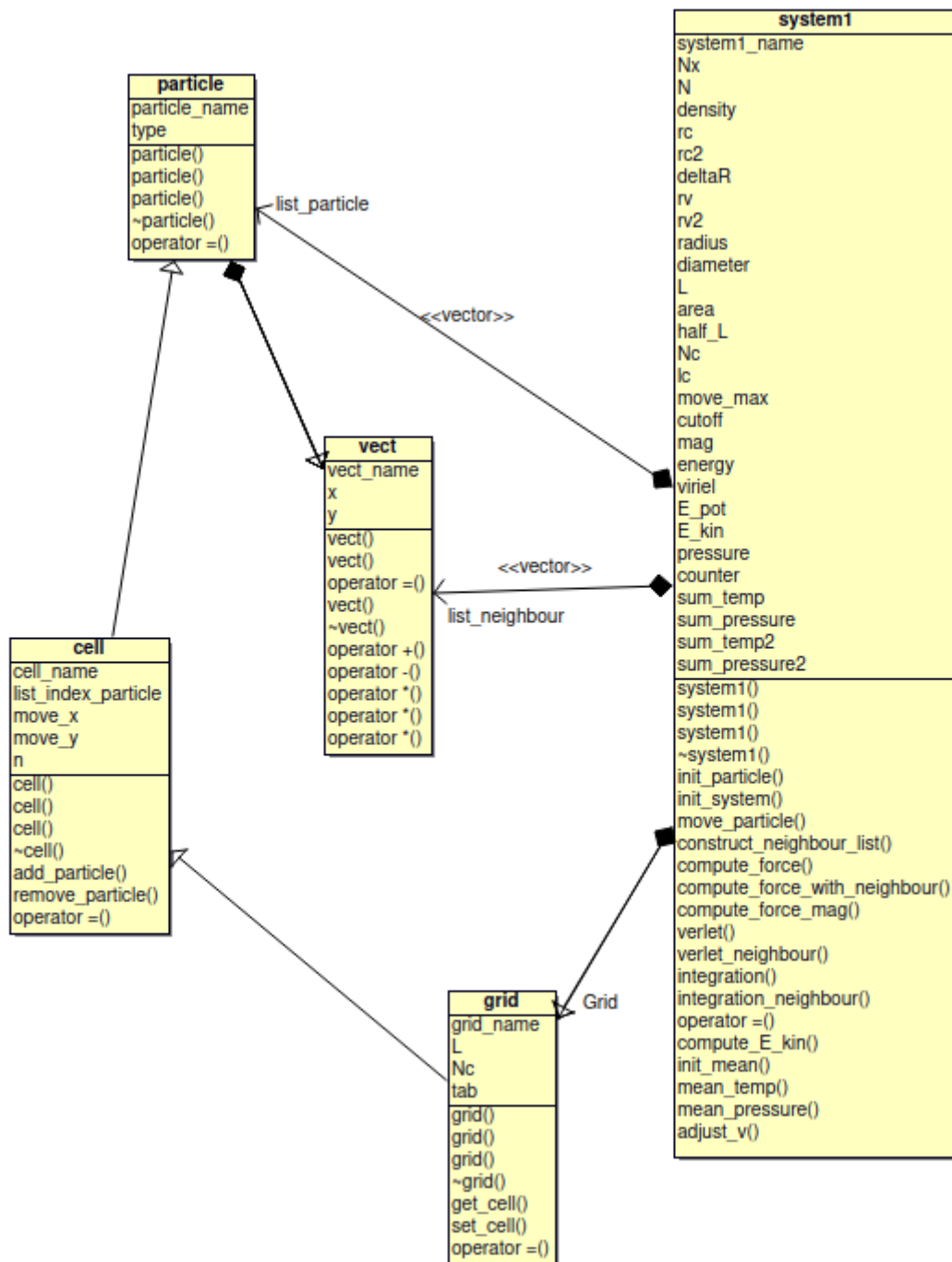


Figure 8.0.1: Class diagram of the code

References

- [1] Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159:98–103, Jul 1967.
- [2] D. C. Rapaport. *The Art of Molecular Dynamics Simulation*. Cambridge University Press, 2 edition, 2004.
- [3] José P. S. Aniceto, Bruno Zêzere, and Carlos M. Silva. Predictive models for the binary diffusion coefficient at infinite dilution in polar and nonpolar fluids. *Materials*, 14(3), 2021.
- [4] William L. Jorgensen, Jeffry D. Madura, and Carol J. Swenson. Optimized intermolecular potential functions for liquid hydrocarbons. *Journal of the American Chemical Society*, 106(22):6638–6646, 1984.
- [5] Edward P. Furlani. *Permanent Magnet and Electromechanical Devices : Materials, Analysis, and Applications*. Academic Press, 2001.
- [6] Problems of the french physicists' tournament. <https://france.iptnet.info/problemes-du-fpt-2022/>.