Université de Strasbourg

Birth-death process and Schrodinger ground state calculation

Project 8

Gitub link : https://github.com/Zoe-zbn/Projet-AMC

2023

ZOBUNDZIJA Zoé, PACREAU Angeline, SAKHO Awa

M1 Physique 2022-2023

Abstract

The interest of this project comes from the paper by James B. Anderson where he calculates the ground state of the atom H_3^+. In this report, we will relate the method used in the paper to two particular cases, the 1D quantum harmonic oscillator and the 3D hydrogen atom.

Introduction

In 1975 , James B. Anderson devised a new way of determining the ground state of a quantum particle in a potential. It calculates, in fact, the ground state of the hydrogen atom using a technique known as quantum Monte-Carlo.

Quantum Monte Carlo encompasses computational methods that aim to provide a reliable solution to the multi-body quantum problem. It is based on an analytical continuation of the quantum mechanical problem in imaginary times. It is a standard trick in theoretical physics that allows us to link quantum problems to problems of statistical mechanics, and vice versa.

It is to this extent that our project consisted of applying this method to two particular cases: the quantum harmonic oscillator with one dimension and the hydrogen atom in 3D.

The method of resolution used, which we will detail later in the report, is based on a decomposition of the schrodinger equation into a diffusion term D_q and a "birth-death rate term" $(E - V(\mathbf{r}))$.

$$\frac{\partial \psi}{\partial \tau} = D_q \Delta \psi + (E - V(\mathbf{r}))\psi \tag{1}$$

The Monte-Carlo algorithm will then play on the two parameters above using a time step τ.

Thus, firstly, we will detail the method of resolution used, secondly we will present the results obtained and finally we will discuss and give some ideas about the data obtained.

1 Resolution method

To solve the studied problem we used a quantum Monte Carlo, with a Brownian diffusion characterized by a Gaussian dispersion defined with the following parameters $(0, \sqrt{2 * Dq * dt})$.

We have therefore used a "virtual" time loop of intervals dt. On each time interval dt, we first make the position of each living or dead particle evolve by a random addition dx , constrained by a Gaussian (due to the diffusion term Dq).

1.1 Brownian Diffusion

The first step of our loop is the diffusion. It allows us to make the position of each living or dead particle evolve by a random addition dx , constrained by a Gaussian (due to the diffusion term Dq).

$\hookrightarrow$ All our particles have evolved spatially

1.2 Test

The second step of the time loop is the survival test only on living particles ($s = 1.0$) defined as:

- First we check that the particle is alive $s = 1$ if it is not alive we pass ($s = 0$ to the next particle if it is alive we enter the test.
- After the reference energy is then compared to the potential $E - V(x)$ of the considered particle:$E - V(x)$.
- If $E - V(x) > 0$: We have a probability of creating a particle defined as $p = e^{-dt*(V(r)-E)/hbar}$. The probability is then compared to a random number between 0 and 1 if this probability is lower than this random number then we create a particle by adding in the list of positions and in the list s a particle.

- If $E - V(x) < 0$: We perform the survival test, the probability that the particle dies is defined as $p = e^{dt*(V(r)-E)/hbar}$. The probability is then compared to a random number between 0 and 1. If this probability is lower than this random number, we delete a particle by adding the reference of this particle in a storage list and by changing the value of s to 0.

The potential functions used are :
The Harmonic Oscillator

$$V(x) = \frac{mw^2}{2} * x^2 \tag{2}$$

Hydrogen atom

$$V_c(x) = -\frac{e^2}{4 * \pi * e_0 |\mathbf{r}|} \tag{3}$$

↪ All our particles were tested under the physical constraint of a reference energy according to their potential.

1.3 Energy readjustment

At each time step and after the test, we try to make the reference energy evolve. In order to have an energy that makes the system evolve in an equivalent way in both directions. Let $E \sim V(r)$. This allows us to work on the term "birth-death rate" $(E - V(\mathbf{r}))$.

The readjustment is done in the way given by Anderson's article [1]:

$$E = V_{mean} - \frac{n - N0}{N0 * dt} \tag{4}$$

We can qualitatively see that with the sign of $\delta E = -\frac{n-N0}{N0*dt}$. If we create particles then δE will be negative, because we have to lower the energy of our system. And conversely if we kill too many particles it means that our reference energy is too low so $\delta E > 0$.

↪ The new value of the reference energy E is thus obtained.

1.4 Data recording

The created lists of the number of particles, the average energy of the system and the reference energy are filled in.

1.5 How the code works

This project has 5 files:

We can modify two files which are: ($Physicalparameter.py$) and ($MC.py$) .

- The first one ($Physicalparameter.py$) being the physical constants of the system as well as the 2 potential functions used.
- The second file ($MC.py$) is filled with the Monte-Carlo parameters (time, number of initial particles, temporal step). These parameters being put in a separate file, it allows us to simply modify our initial conditions and thus to modify without touching the main code the precision of our algorithm.
- The first code file is ($Browniandiffusion.py$). In this code we have performed the test for Brownian diffusion in one dimension and then in 3 dimensions that we will use in the 2 following algorithms allowing to calculate the fundamental state of the harmonic oscillator and of the hydrogen atom.
- The last 2 files ($Oscillatorharmonic.py$) and ($hydrogenatom.py$) are made of the algorithm defined above for the harmonic oscillator and the hydrogen atom. The output results of these 2 algorithms are graphs that can show the evolution of the energy as a function of virtual time as well as the number of particles.

So we can modify two files which are: ($Physicalparameter.py$) and ($MC.py$) .
All our code is presented in the following GitUb space:
https://github.com/Zoe-zbn/Projet-AMC

1.6 Optimization of the compilation time

While programming our code, we encountered the problem of the compilation time of our files (too long). At the beginning we had decided to create a new particle each time and to leave the killed particles in our lists. But this would generate large lists of particles, which would imply a very long compilation time. To solve this problem we thought about a storage list. Where the position information of the dead particles is integrated. This allowed that when we create a new particle. We retrieve a position from this list (we then "resurrect" the dead particle). Thanks to this method we managed to minimize the compilation time of our programs and not to exceed the size of the maximum allowed list $n_{max} = N0 * 10$.

The results of the difference in compilation times will be presented to you during the oral.

2 Resultat

2.1 Brownian Diffusion

At the very beginning of the implementation we first tried to verify that diffusing with a Gaussian function gives a diffusion similar to a Brownian propagation. We then perform these tests in the code $Browniandiffusion.py$.

The results obtained are presented in the two figures below:

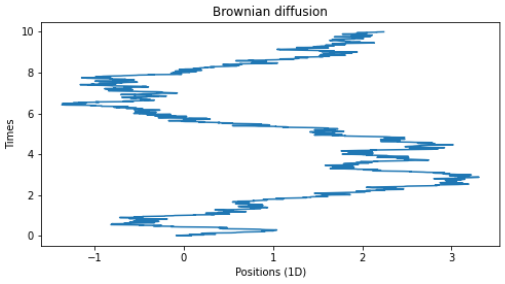

Figure 1: Brownian Diffusion : 1D

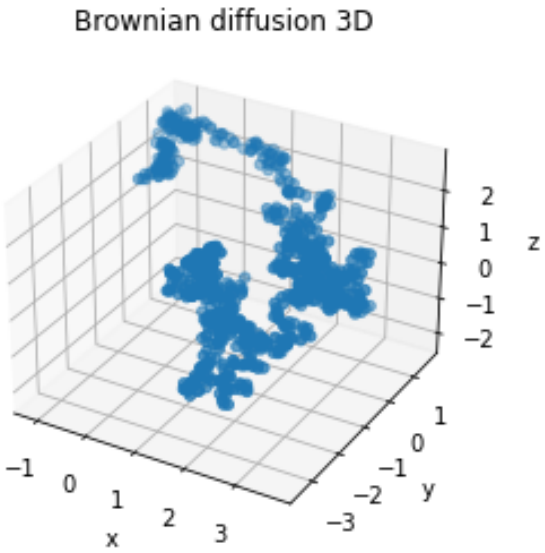

Figure 2: Brownian Diffusion : 3D

These 2 figures give us the certainty that using a Gaussian function allows to have a Brownian diffusion.

2.2 Harmonic Oscillator

With the code explained above, we obtain the following figures, first for the harmonic oscillator. First you have the evolution of the energy at each time step, where the curve becomes constant Figure 3.

Then you can look at the evolution of the number of particles. Figure 4, you can see that it is constantly changing; according to the process we based our code on: the birth-death process.

Finally, we want to show you the distribution of particles at a random time along the only dimension we took in this code. With some oscillations, the Figure 5 looks like a Gaussian.

2.3 Hydrogen atom

A second part of our programing was to do the same thing but implementing a different potential to simulate the Hydrogen atom. The Figures 6 and 7 are the same figures drawn by this second code.

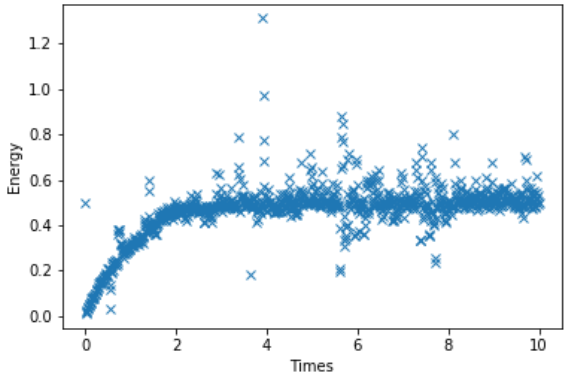

Figure 3: Evolution of the energy

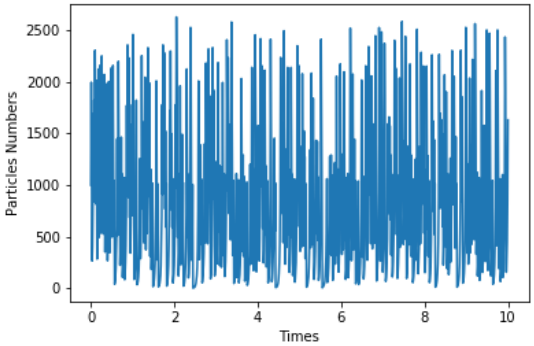

Figure 4: Evolution of the number of particles

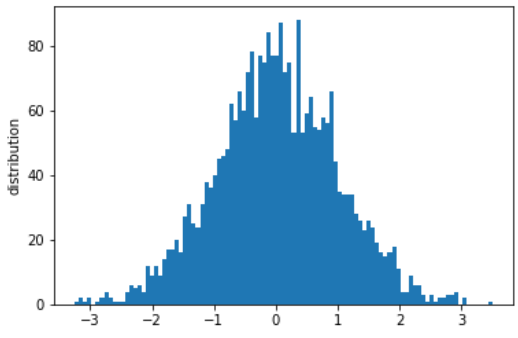

Figure 5: Repartition of the postions for the particles at the end

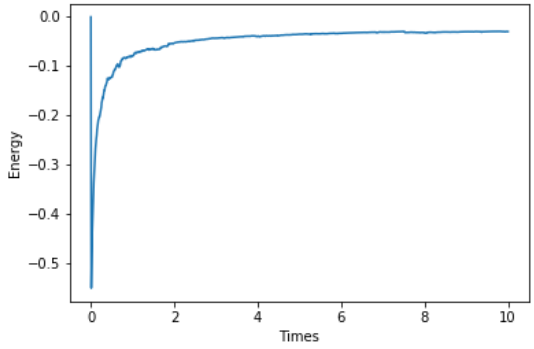

Figure 6: Evolution of the energy

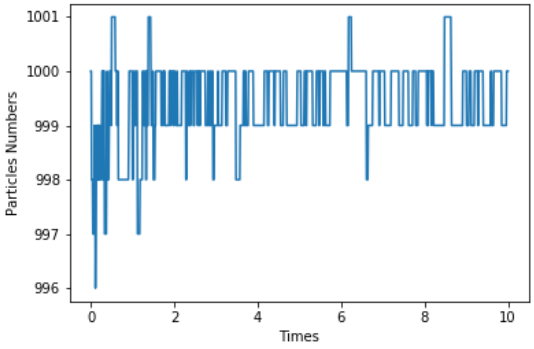

Figure 7: Evolution of the number of particles

3 Discussion

3.1 Some Precisions

The accuracy of our algorithm is directly related to the Monte Carlo parameters.
For example if we take a time T_1 and a time T_2 and that T_1 is greater than the time of T_2 . And that the step dt is the same then we will obtain more accurate results for the Monte Carlo having worked with a time T_1.
We can then also carry out this reasoning with the number of particles if we have a number of particles N_1

higher than a number of particles N_2 and having the same temporal parameters then the results obtained thanks to the simulation with the number of particles N_1 will have more precise results.
To summarize, increasing the time T and decreasing the step dt as well as increasing the number of initial particles N_0 allows to have a much higher number of iterations in the Monte-Carlo. This induces a better accuracy in our results.

3.2 Harmonic Oscillator

The results obtained for the harmonic oscillator are convincing. The energy of the fundamental state obtained by simulation is very close to the fundamental state obtained by an analytical calculation, i.e. $E_{exp} \sim E_0 = 0.5$ (Fig3). And when we change a physical parameter for example w, we also obtain a ground state very similar to the analytical ground state.

But nevertheless we can observe that our curve which tends towards this fundamental state is degraded by small energy jumps. These deviating points can perhaps be explained by the fact that we do not test the position due to diffusion. So maybe by testing this position with a probability density we will obtain a much more stable system and thus a curve tending towards the analytical ground state much less degraded.

3.3 Hydrogen Atom

The results for the hydrogen atom by testing only the Colombian potential function V_c, gives us a fundamental energy around 0.2 (Fig6).
The value of the energy of the analytical ground state is given by the following formula:

$$E_n = -\frac{1}{n^2} * \frac{1}{2}(u.a) = -\frac{1}{2} * \frac{1}{4\pi\epsilon_0}\frac{e^2}{a_0}(SI) \tag{5}$$

Considering all our parameters equal to one then we obtain a ground state about equal to $E_0 = \frac{1}{8} = 0.125$. So we have an analytical ground state energy more or less close to the obtained energy $0.2 \approx 0.125$. We can then say that by improving the accuracy of our Monte-Carlo and using as in the harmonic oscillator a density function during the scattering we would obtain results even closer to the analytical energy.

3.4 Perspectives

An extension of this work would be to compare this quantum Monte Carlo method with that of quantum creptation.

The latter is similar to the Monte Carlo diffusion, except that it works with paths instead of points. This presents some advantages related to the computation of some properties of the studied system with which Monte Carlo diffusion has difficulties.

Even if it first aims to solve the time-dependent Schrödinger equation in the imaginary direction of time, by propagating the Schrödinger equation in time, we obtain the dynamics of the studied system, and by propagating it in an imaginary time, we obtain a system that tends towards the ground state of the system.

This method could allow an update to the Monte Carlo method by slightly moving the walkers, then duplicating and deleting some of them.

Conclusion

To conclude, thanks to this method of programming, it is possible to find the energy of the groundstate of a given system, even if our code for the hydrogen atom wasn't totally concluding. It also allowed us, as students, to tackle the Monte-Carlo modelisations, from a point of view not oftenly used.

Our work here can make us think about the Hartree-Fock method used in computational phyisics and chemistery. This method is used to have the wavefunction and the energy of a quantum many-system in a stationnary state. It is a good first approximation for the interaction between the electrons and the nucleus of an atom. We could then complexify our code with a Hartree-Fock Hamiltonian to fine-tune our results in an atom.

References

[1] Anderson J. B. A random-walk simulation of the schrödinger equation: H+3. *The Journal of Chemical Physics*, 63, 1949.