Monte Carlo simulation of recent experiments with ultracold dipolar atoms in one dimension

Author: Giorgi Mikautadze
Director: Grigori E. Astrakharchik
Departament de Física (FIS)

January 23, 2022

Facultat d'Informàtica de Barcelona (FIB)
Bachelor Degree in Informatics Engineering
Computer Science Major

Abstract

The quantum many-body problem refers to the problem of predicting the properties of a microscopic quantum system from the first principles of quantum mechanics. Under this umbrella fall various problems of fundamental importance in fields fields of quantum chemistry, condensed matter physics, and materials science. Recent experiments consisting on the creation of topological quantum states in a bosonic one dimensional quantum gas by stabilizing it against collapse and thermalization, and proceed to apply contact interactions to the gas using energy-space topological pump. The aim of this Thesis it to implement quantum Monte Carlo algorithm in order to simulate a Bose gas in the presence of topological excitations and to contrast the obtained data with the recent experimental results. We were able to correctly describe the energy of the many-body quantum system in an *ab-initio* simulation based directly on a microscopic model.

Resumen

El problema de muchos cuerpos se refiere al problema de predecir las propiedades de un sistema microscópico a partir de los principios de la mecánica cuántica. Bajo este paraguas se encuentran varios problemas de fundamental importancia en los campos de la química cuántica, la física de la materia condensada y la ciencia de los materiales. Experimentos recientes consisten en la creación de estados topológicos en un gas cuántico unidimensional bosónico mediante su estabilización frente al colapso y la termalización, y se procede a aplicar interacciones de contacto al gas mediante bombeo topológico energía-espacio. El objetivo de esta Tesis es implementar el algoritmo cuántico de Monte Carlo para simular un gas de Bose en presencia de excitaciones topológicas y contrastar los datos obtenidos con los resultados experimentales recientes. Pudimos describir correctamente la energía del sistema de muchos cuerpos en una simulación *ab-initio* basada directamente en un modelo microscópico.

Resum

El problema de molts cossos fa referència al problema de predir les propietats d'un sistema microscòpic a partir dels principis de la mecànica quàntica. Sota aquest paraigua cauen diversos problemes d'importància fonamental en els camps de la química quàntica, la física de la matèria condensada i la ciència dels materials. Experiments recents consistents en la creació d'estats topològics en un gas quàntic unidimensional bosònic estabilitzant-lo contra el col·lapse i la termalització, i es procedeix a aplicar interaccions de contacte al gas mitjançant una bomba topològica espai-energia. L'objectiu d'aquesta Tesi és implementar un algorisme quàntic de Monte Carlo per simular un gas Bose en presència d'excitacions topològiques i contrastar les dades obtingudes amb els resultats experimentals recents. Hem pogut descriure correctament l'energia del sistema de molts cossos en una simulació *ab-initio* basada directament en un model microscòpic.

Contents

Figures

Tables

List of Algorithms

Part I
Introduction

1 Prelude

What is the difference between a molecule of water and a droplet? Or a droplet and an ocean? What is the difference between a person and a society? An important aspect of the difference is the complexity that arises from collecting a bunch of those individuals. Start with two atoms they will bounce around, but add more and you obtain a gas, or a liquid or a solid. Collective properties are present everywhere, wherever you look, in the sea, the atmosphere, the coffee you drink or the cells that form your body. Collectivity always takes over and creates unique patterns and emergent phenomenons. [1] The many-body problem is a general name for a vast category of physical problems pertaining to the properties of microscopic systems made of many interacting particles

It is a known saying that "the universe tends to chaos´´. Given enough time, even the tidiest room will get messy. Papers, bed sheets and clothes will leave their ordered state and scatter around the surface of the room. This tendency towards untidiness reflects a law of the nature of gradual decline into disorder.

For example, a can of coke is pumped with carbon dioxide at high pressures. Once we open the can the pressure of the carbon dioxide inside is relived and the dissolved gas escapes as bubbles. If we place an ice cube in a cup full of hot water, the molecules frozen in the ordered crystalline lattice will break their bonds and disperse as the system strives toward equilibrium with its environment. This process is called thermalization.[2]

Many-body scars are a feature that have been shown to appear in isolated quantum systems that would be expected to thermalize but the observables in this system seem to avoid thermalization and rewind back to its initial state. As if the ice placed in the hot cup melted and crystallized, melted and crystillized and so on continuing to oscillate a few times — long after it should have thermalized.

Physicists have designated this bizarre behavior as *quantum many-body scarring*. Because as if scarred, the particles seem to bear an imprint of the past that draws them back to their original configuration over and over, therefore appearing to defy the universe's push towards disorder. [3]

Recent experiments consist on the creation of topological quantum states in a bosonic one dimensional quantum gas by stabilizing it against collapse and thermalization, and proceed to apply contact interactions to the gas using energy-space topological pump.

1.1 Context

This project is developed as a Bachelors Degree Thesis in Informatics Engineering at *"Facultat d'Informàtica de Barcelona"*, FIB, part of *"Universitat Politècnica de Catalunya"*, UPC. It is a computing major project with the aid of the physics department of the FIB.

As a context, we will try to simulate the experimental results of the scientific research paper *Creating quantum many-body scars through topological pumping of a 1D dipolar gas*[4] via Monte Carlo sampling. Monte Carlo methods are the

most accurate techniques to microscopically study the ground state. We will then proceed to define some concepts.

- Monte Carlo simulation is a method of stochastic simulations in real-time that incorporate random variability into their model. The sampling distribution has to be defined *a priori* and the Monte Carlo algorithm will repeatedly simulate the model, each time drawing different configurations, the result of which is a set of possible outcomes.[5]

- The model used for this project is known as delta potential, and is described by Dirac delta distribution which is distribution over the real numbers, whose value is zero everywhere except at zero, and whose integral over the entire real line is equal to one. It is used so simulate a box, in this case one-dimensional, where particles freely roam inside the box but can not escape it or even hit its walls [6]. Therefore, we precise some boundary conditions.

- The experiment of the paper, and similar ones are carried out in a cigar shaped atom clouds containing up to 10^6 atoms in a single experiment. While a simulation such as Monte Carlo uses a box that could contain up to 10^3 particles, due its quadratic complexity. This number is obviously few orders of magnitude smaller, but this can be solved by the use of periodic boundary conditions. In which case any particles that would leave the box by one side will reenter by the other side. Making it a loop with as many particles as we want.

- Dimensionless units will be used to make our computations. This is a system of units where physical quantity can be transformed to an equivalent dimensionless quantity by way of scaling factor[7]. These numbers are close to the unity, hence much more manageable when working than values on atomic scale. Obviously, dimensionless units can be converted back any time, thus not losing the actual physical values.

It is very important to denote that the total energy of a quantum mechanical system is defined as the expectation value of the Hamiltonian operator H, which includes both kinetic and potential energies, [8]

$$H = E_{kin} + E_{pot}. \tag{1}$$

Due to its close relation to the energy spectrum and time-evolution of a system, it is of fundamental importance in most formulations of quantum theory.

1.2 Project Goal

The goal is to develop, test and apply an *ab-initio* Monte Carlo code which would be able to explain the experimental results starting from a microscopic model. The simulation has to be able to reproduce experimental results of the scientific research paper *Creating quantum many-body scars through topological*

pumping of a 1D dipolar gas [4], where they stabilize a dipolar gas against collapse and thermalization and begin to contact interact with it in order to show an energy-space topological pump. In other words, carrying out a broad range of interactions described by potentials such as the Dirac delta and its derivatives.

The experiment conveys that for fixed values there exist different possible topological solutions, with different energies. Therefore the Hamiltonian does not provide complete information of the system.

1.3 Justification

Since quantum many-body scars are long-lived excited states of correlated quantum chaotic systems that seem to rewind thermalization, they are of great fundamental and technological interest.[4] Especially in the quantum computing field, which in the future is likely to solve problems that have been impossible to solve before.

The need for this project was the lack of exact many-body calculations which could simulate the properties of the quantum system under experimental conditions. To fill this void, a Monte Carlo algorithm capable of simulating quantum many-body systems will be implemented from scratch and assessed with the finality of describing on field values.

1.4 Stakeholders

For the purpose of carrying out this draft we have to mention the stakeholders. Without them the project could not be done.

- **The authors of the experiment and the research paper**. Their experimentally obtained values will be put to use to compare with the results of our Monte Carlo code simulation.
- **The director of the project**. The director is part of the department of physics of the FIB. He is the lion's share in the theoretical quantum mechanics field of the project.
- **The author of the project**. The author is designated to work on the development of a Monte Carlo code. He is in charge of choosing suitable data structures and implementing the algorithms required to obtain the physical quantities we want to observe.

2 Scope

In the following sections the main objectives of the project as well as their sub-objectives and requirements will be thoroughly defined, with a fitting evaluation of the risks and obstacles that could appear.

2.1 Objectives and its sub-objectives

As mentioned previously on the project goal, our intention is to develop, test and apply an *ab-initio* Monte Carlo code which would be able to explain the experimental results starting from a microscopic model. In order to achieve such a complex goal, it has been divided in simpler tasks or sub-objectives that can be accomplished over an efficacious period of time.

1. **Develop**. Make a Monte Carlo simulation, in this case it was found suitable the use of a variant of Monte Carlo, the Metropolis algorithm. The first step is then to implement a code for Metropolis algorithm. The requisites for completing this objective are:
   (a) Define our model. Consists of a one dimensional box with a relation between its size (or length) and the number of particles it contains.
   (b) Implement and apply periodic boundary conditions.
   (c) Choose a trial wave function, and a probability density function that can be sampled from and eases the calculations for the different observables.
   (d) Implement a function that obtains the difference between contributions Δu, which states if a candidate particle position probability is higher than the current one.
   (e) Deriving the equations to calculate the different observables i.e. physical quantities in the simulated systems.
   (f) Implement the code to compute the observables with Metropolis algorithm.
      i. The pair distribution function $g(x)$, which is proportional to the interparticle distances of the system.
      ii. The drift force of a particle, a vector that defines the direction in which the wave function grows. The drift force is not an observable by itself, but it is needed for the calculation of the ground-state energy i.e. the lowest-energy state of the system.
2. **Test**. After developing the Metropolis algorithm, a verification of the code is needed to ensure its correct functioning. The verification is carried out through a simulation of an ideal Fermi gas system, which consists of fermions (or particles that do not interact). The results are to be compared with theoretical values. The requirements of this objective are:
   (a) Define the Jastrow term $u(x)$, a term which constructs the wave function. It is known exactly for the case of the ideal Fermi gas.
   (b) Derive the first and second logarithmic derivatives $u'(x), u''(x)$ of the Jastrow term to calculate the drift force and later the ground-state energy.
   (c) Test the pair distribution function $g(x)$ for a simulation of the ideal Fermi gas.

(d) Test the ground-state energy per particle of the ideal Fermi gas. In order to do this we will use different estimators, and analyse the differences between them.

(e) Implement a function able to measure the statistical error of the Energy obtained with independent measures obtained using the Metropolis algorithm.

(f) Examine the finite size effects of our simulation and how those effects relate to analytical equations.

3. **Apply**. Once the correctness of the code is verified, it can be applied to a dipolar system such as the experimental one.

   (a) Define a new wave function, with a new Jastrow term to operate with the new system.

   (b) Implement a recursive dichotomic search to solve the Jastrow term as a function of parameters, such as the scattering momentum and the phase shift.

   (c) Understand and explain the results.

      i. Find the experimental results of the research using g3data.

      ii. Reproduce the experiments with different values of the coupling constant, g, and explain them form a microscopic level.

2.2 Requirements

To conduct this piece of work and achieve all the objectives mentioned above we need to fulfill some requirements.

- **Programming prowess**. We need to implement the algorithms and functions to meet all the objectives.
- **Basic knowledge of quantum mechanics**. Understanding quantum mechanics at a base in level and being eager to keep learning during the implementation.
- **Time to interpret the results**. We need to take the time and patience to correctly interpret, analyze and rationalize the results obtained in order to keep advancing after every milestone or sub-objective.

2.3 Obstacles and risks

Involving a project of this characteristics a lot of possible obstacles and risks might appear around the corner at any given moment. The first obstacle faced by the author of the project was the sheer understanding of the Monte Carlo algorithm, the problem at hand and how Metropolis algorithm would be used to solve it. A great deal of research in the quantum mechanics field was made by my part with the purpose of assimilating the paper

Regarding implementation errors, those are typical in any kind of project where the development of code plays an important role. This types of errors are to a greater or lesser degree easily solved, unlike the errors that appear from incorrect theoretical assumptions. This is what made this project so different. Especially at the beginning, with the author's inexperience in the quantum mechanics field and his obliviousness towards some physical notions, one of the major risks became misunderstanding the theory. This could have misguided and lead to the implementation of awry functions which could have rapidly snowballed to a whole lot of erroneous values.

Concerning time issues, the analysis, interpretation and rationalization of the results obtained was another predominant and time consuming part of the project. It is obvious that every function had to be tested, but the outcomes had to be construed to ensure that we are taking a step forward and not going sideways. A faulty implementation of a function, would mean bad outcomes on the future implementations and more time spent finding the errors.

Fortunately, the weakly meetings between the author and the director were fruitful and minimized the risks and obstacles.

3 Project management

In this section we will discuss the methodology and the ways we will be monitoring it. The methodology's objective is to keep the project authentic to the quantum mechanics behind it. The technologies and tools used to conduct it, will be talked through in depth.

3.1 Methodology

It is well known that a methodology needs well documented procedures and guidelines to fall on in case we have any problem with the design, implementation or analysis. The first step was writing down our goal and dividing it into objectives. Then transforming those dense objectives into simpler tasks or sub-objectives (see Section 2.1). The approach for solving an objective is solving its sub-objectives one by one, not moving to next sub-objective until we have achieved our criteria of desired quality. That is solid, sturdy and fast code. The process for solving a sub-objective is listed in the following development progression:

1. **Understand the sub-objective**. As mentioned in previous sections, it is very important to assimilate in depth the new concepts given by each sub-objective to design correctly a code that solves the problem at hand. Hence, the first stage of the development progression is reading, researching and asking about any conceptual doubts until the notions are fully apprehended.

2. **Implement the code**. Once the task is deeply understood the next step consists on the development of a simple and efficient code that carries out

the task.

3. **Verify the results**. The results obtained by the preliminary implementation have to be reviewed and inspected thoroughly. Depending on the task, we can compare the results with analytically known answers, typical plots or using mathematical tools to check the outcomes.

   (a) If the results are not satisfactory, we have to find out if the problem is given by an implementation error or a misunderstanding of the theory, hence we have to decide if we go back to *Understand the sub-objective* or *Implement the code*. Regardless, after the appropriate changes are made we *Verify the results* again.

4. **Move forward**. Once the results of the preliminary implementation are verified through rigorous testing and it is guaranteed that it will keep working as intended, we have hit a milestone. Moving forward to the next sub-objective.

This process is repeated until the accomplishment of all objectives.

3.2 Monitoring

The whole project was strictly monitored and surveilled. An eye was kept on it all the time via weekly meetings between the author and the director, where the issues that occurred during the week were discussed, the progress was reviewed and the assignments for next week were set on a stone. In case problems occurred during the week, we held meetings or discussions via Google services, such as Google Meet and Gmail.

3.3 Tools and technologies

The code that performs our simulation is exclusively programmed in the programming language C++. There were multiple reasons for this choice. First of all, it's portability allows for a development indifferent to the hardware and to the operating system. Therefore, making it usable in most machines and both Windows and Linux without any issues. It is fast, powerful and even more important it has a high scalability. Hence, resource-intensive applications, like Monte Carlo, can be built using it.

The code is written using the code editor Visual Studio Code which proved useful correcting trivial mistakes and suggesting endings for code lines. For the version control we used GitHub to oversee the project, explore the changes that occurred and overall keeping track of the work.

As an operating system we used Windows but took advantage of its new feature Windows subsystem for Linux, WSL, to use features of Ubuntu's terminal.

For the purpose of plotting our results and advances during development we used GNUplot a portable command-line driven graphing utility.

As for a mathematical tool we used Wolfram Mathematica, a software system with built-in libraries for several areas of technical computing. This served as a tool to review any implementation that required complex mathematical cases.

Concerning the actual experimental data, we had to extract the values that the researchers obtained with their results. In order to do so we used a tool called g3data, which is used for extracting data from graphs. In publications, graphs are often included but the actual data is missing. G3data makes the extracting process much easier.

Lastly, all the meetings and communication between me and the director were scheduled using Gmail. Even though most meetings were in person, a couple of them were using the Google Meet service. For the documentation we opted to use the Latex text editor built in the Overleaf web application. This allowed me, the author, to freely document the work while the director could open the shared file any given time to check on anything I could need. Making the workload very dynamic.

3.4 Alternative tools

Evidently this work could've been done using alternative tools just as good as the ones we used.

Regarding programming languages, we found C++ to be better suited for complex and highly dynamical data structures for generic purposes. Nonetheless, there are alternatives such as Fortran which is a language adapted to numeric applications.

As for a graphic utility, GNUplot was useful and enough for all the graphical work we planned to do in this project. An alternative graphic utility would be the matplotlib library from Python.

With respect to a mathematical tool, obviously any of the ones available could have been used. Wolfram Mathematica was chosen, but an alternative could have been the mathematics-based software Maple.

Part II
Project management

4 Time planning

The time planning and the resources employed on the elaboration of this project will be discussed in this section. In first place, denote that I, the author, started researching the subject of this Bachelors Degree Thesis around mid July 2021. The first mails between me and the director date back to 13th July 2021. And the Thesis is to be submitted the week of the 17th of January 2022. With a simple quotient we obtain 218 days worth of work. That is, a little over 7 months.

Out of this 218 days, the first month was invested in a general research about the topic chosen. Around 20 hours were spent during the first 30 days understanding the subject in a basic fashion. This makes for a more exact calculation of (218-30)days + 20 hours worth of work. The Bachelors Degree Thesis is worth 18 credits ETCS. Each of them representing 25 hours of work,

$$1\ ETCS = 25\ hour\ . \tag{2}$$

Therefore, 18 credits are worth 450 hours. For every one the 188 days the roughly estimated average of work per day is

$$\frac{430\ hours}{188\ days} + 20 hours\ . \tag{3}$$

Or 2.3 hours a day plus the previous 20 hours of research.

4.1 Description of the tasks

It is very important to have clearly defined tasks to organize our efforts. In this section the different tasks will be described and classified.

PTM - Project and Time Management

1. **[PTM1] - Scope**
   The scope of the project allows to set limits on the project and precisely define the objectives, deadlines and deliverables to be achieved. By cleary defining the scope of the project we ensure the realization of the project goals and objectives evading both delays and work overload. Approximately, 20 hours will be dedicated to the scope.

2. **[PTM2] - Project planning**
   Project planning consists of a systematic organization process of the different tasks to be carried out, as well as the resources necessary to conduct them. Therefore, planning a project is pulling of an action plan for all phases of it. Approximately, 15 hours will be dedicated to the project planning.

3. **[PTM3] - Sustainability report**
   The sustainability report aims to check if the process meets the necessary

conditions to be sustainable. Once reviewed, the necessary corrections in the process can be made in such way as to reduce the unsustainable conditions of the project. Approximately, 5 hours will be committed to making a sustainability report.

4. **[PTM4] - Meetings**
   Weekly meetings where the the author briefly communicates the status of his work to the director. The meetings last an average of two hours and as much as it is for solving and discussing problems, it is also a way of briefing about the upcoming work. There are 17 meetings planned from September 2021 to January 2022. Counting two hours per meeting, we dedicate 34 hours to monitoring.

5. **[PTM5] - Documentation**
   The memory of the project is of great importance for this Thesis, since it corresponds to a large percent of the grade. The documentation will be carried out in parallel to the development of the project, and the different sections and chapters of the memory will be gradually recorded and filled. The estimated time spent on documentation is that of 60 hours.

6. **[PTM6] - Preparation of the defense**
   The defense also plays an important role on the evaluation of the Thesis. The preparation of the defense will take place after submitting the memory and with and estimated time of 15 hours preparing it.

PW - Previous Work

1. **[PW1] - Research about quantum physics**
   It can not be expected to solve a problem without understanding it. Thus, a research on quantum mechanics had to be made to prone the project towards its success. The author of the project spent approximately 20 hours of quality time learning different concepts related to the project.

2. **[PW2] - Research Metropolis algorithm and Monte Carlo variants**
   The Monte Carlo algorithm is complex and requires a deep research to understand how it works before start coding. Around 8 hours were used researching the Monte Carlo and its variants.

3. **[PW3] - Create a working space**
   Creating an appropriate working space to write code in optimal conditions and have a solid testing environment is as important for the successful outcome of the project as it is researching. In 2 hours the author prepared the working environment.

IO - Development, testing and application of the Monte Carlo simulation

1. **[IO1] - Monte Carlo algorithm**

(a) **[IO1.1] - Define a 1D box**
Define a one dimensional box with a relation between its size (or length) and the number of particles it contains.

(b) **[IO1.2] - Implement periodic boundary conditions**
Implement and enforce periodic boundary conditions to the one dimensional box and each and every one of the particles contained in it.

(c) **[IO1.3] - Metropolis algorithm**
To develop the Metropolis algorithm, we need to implement a function that obtains the difference between contributions Δu, which provides information on the quotient between a candidate particle position probability and the current probability.

(d) **[IO1.4] - Derive equations to calculate the observables**
Deriving equations to calculate the different observables, that is the physical quantities in the simulated systems. Including, the pair distribution function $g(x)$ and the ground-state energy.

(e) **[IO1.5] - Implement the pair distribution function**
Implement the obtention of the pair distribution function $g(x)$ as an histogram.

(f) **[IO1.6] - Implement the drift force**
Implement a function that calculates the drift force from particles in the phase space. The drift force is necessary to calculate the ground-state energy.

(g) **[IO1.7] - Implement the ground-state energy**
Implement a function to obtain the ground-state energy of a system during the Metropolis algorithm.

2. **[IO2] - Code verification**

(a) **[IO2.1] - Verification calculating pair distribution function**
Prove its correctness simulating an ideal Fermi gas, calculating the pair distribution function $g(x)$ and comparing with analytical values.

(b) **[IO2.2] - Verification calculating the ground-state energy**
Prove its correctness simulating an ideal Fermi gas, calculating the ground state energy and comparing with analytical values.

(c) **[IO2.5] - Implement a function that measures the statistical error**
Implement a function able to obtain the statistical error bound to arise due to Metropolis algorithm making numerous independent measures of the energy.

(d) **[IO2.3] - Verification testing finite size effects**
Examine the finite size effects of our simulation and scrutinize if the effects come to a good agreement with their analytical equations, which are exact for ideal Fermi gas.

3. **[IO3]** - Simulate using the wave function corresponding to the experiment with dipolar atoms

   (a) **[IO3.1] - Define a new wave function**
   Define a new wave function with a new Jastrow term to operate with the new system.

   (b) **[IO3.2] - Implement a dichotomic search algorithm to solve the wave function**
   Implement a recursive dichotomic search to solve the Jastrow term as a function of new parameters, such as the scattering momentum and the phase shift.

   (c) **[IO3.3] - Apply the simulation using the new wave function and extract results**
   Use our Metropolis algorithm to find values of the energy and compare them with experimental results.

TR- Testing and reviewing

1. **[TR1] - Test the implementations**
   The appropriate tests have to be planned and carried out in order to scrutinize if the implementations are working properly. Any software error has to detected and rectified.

2. **[TR2] - Review and analyze the results**
   The analysis and review of the results is a process consisting on the inspection of the data in order to highlight useful information, to suggest conclusions, and to support decision making. This analysis is the conclusive part of an implementation, where the information has to be presented in an orderly and understandable way.

4.2 Human resources

The human resources expended on this project are the director and the author of the project. Some specific roles can be defined for both of us. For the author of the project, the roles of project manager, researcher, programmer and tester. As for the director, the role of specialist due to his insight as a recognised professional in the field.

1. **[PM] - The project manager**
   The project manager is responsible for ensuring that all stages of a project are fulfilled and the objective set is achieved. He has to know how to distribute tasks and be a facilitator. Consequently, maintain excellent communication with all those involved in the project. Although this case, even if we have assign multiple roles, there is only two people involved.

2. **[R] - Researcher**
   The researcher manages the development of the research in its various

stages. He has to come up with research proposals to enrich the project and make it more appealing. Also provides details the ease the current job of other roles.

3. **[P] - Programmer**
   The programmer uses the research made by the researcher to program efficient solutions for our tasks.

4. **[T] - Tester**
   Plan and carry out tests on our implementations to check whether they are working properly or have a faulty design.

5. **[A] - Analyzer**
   The analyzer makes an analysis and review of the results obtained by the execution of the programmers work once it is already tested. He has to make sense of the outcomes.

6. **[S] - Specialist**
   It is a role assigned to the director. The role of specialist is given to him since he is a recognised professional, has experience through practice and education on quantum physics and specifically in the field of the project.

4.3 Technical Resources

1. **[PC] - Personal computer**
   A personal computer used for programming and analytical purposes.

2. **[GNU] GNUplot**
   An application useful for graphical tasks and plotting.

3. **[GitHub] GitHub**
   A service used for version control of our project. Very useful for the tester to *debug* the program.

4. **[Latex] Latex**
   An online text editor service we use to carry out the documentation of the project.

5. **[G3] g3Data**
   A graphic tool which is used to extract data from graphs.

6. **[M] Wolfram alpha Mathematica**
   A mathematical tool to solve equations and check results.

4.4 Final time schedule

To sum up all the information in this section we have made a table Tab. 1 that shows for each task, the human and technical resources needed to complete it, as well as an added estimated time of completion (in hours).

Task table					
ID	**Task**	**Time** (hours)	**Dependency**	**Resources**	**Role**
PTM	Project time management	130	-	-	-
PTM1	Scope	10	PW1, PW2	-	PM
PTM2	Planification	10	PTM1	-	PM
PTM3	Sustainability Report	6	PTM1	-	PM
PTM4	Meetings	34	-	-	PM, S
PTM5	Documentation	60	PTM1..PTM3, IO	Latex	PM
PTM6	Preparation of the defense	10	PTM1..PTM5	-	PM
PW	Previous work	30	-	-	-
PW1	Research quantum physics	20	-	PC	R
PW2	Research Monte Carlo	8	-	PC	R
PW3	Create a working environment	2	PW1, PW2	PC	PM,P
IO1	Monte Carlo algorithm	140	-	-	-
IO1.1	Define 1D box	5	PW	PC	P
IO1.2	Periodic boundary conditions	5	IO1.1	PC	P
IO1.3	Metropolis algorithm	60	IO1.2	PC	P
IO1.4	Derive observable equations	10	PW	PC	P, R
IO1.5	Pair distribution function	20	IO1.3, IO1.4	PC	P, R
IO1.6	Drift force	20	IO1.3, IO1.4	PC	P, R
IO1.7	Ground-state energy	20	IO1.3, IO1.4	PC	P, R
IO2	Code verification	40	-	-	-
IO2.1	Test pair distribution function	5	-	PC	A
IO2.2	Test ground-state energy	15	IO1, IO2.1	PC, M	A
IO2.3	Statistical error	5	IO2.2	PC	P, A
IO2.4	Test finite size effects	15	IO2.3	PC	A
IO3	Dipolar system	60	-	-	-
IO3.1	Define new wave function	10	-	PC	R, S
IO3.2	Recursive dichotomic search	20	IO1, IO2, IO3.1	PC, G3	P, A
IO3.3	Apply the simulation	30	IO1, IO2, IO3.1, IO3.2	PC	P, A
TR	Testing and reviewing	50	-	-	-
TR1	Test	20	-	PC, M, GitHub	T
TR2	Review and analyze	30	-	PC, GNU	A
-	Total hours	450	-	-	-

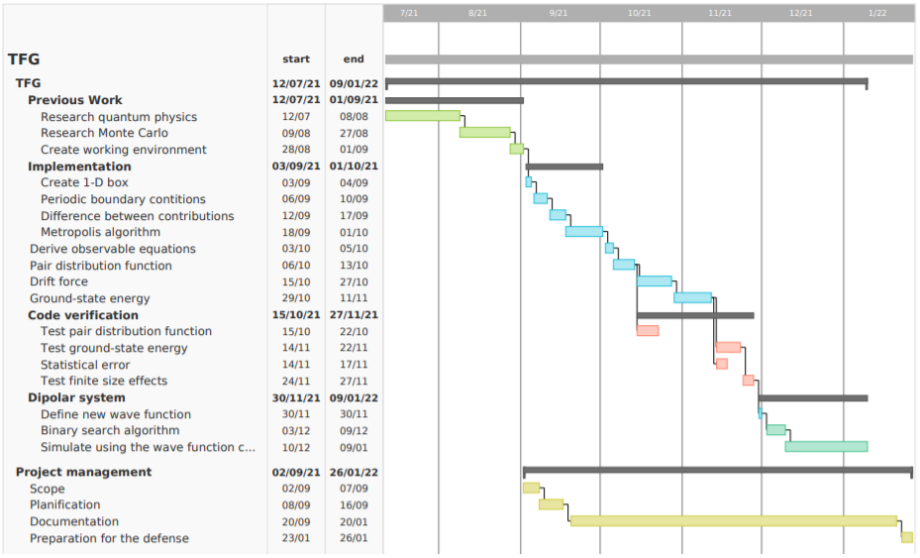
Table 1: Task table (own sources)

5 Risk management

It is well known saying that there is *"no such thing as a risk free investment"*. Well then, investing time in a project might not be answered with fruitful returns. To have a good risk management plan, an analysis of the risks has to

be made. The risks this project possesses is being able to develop the code, from start to end, for the Monte Carlo simulation within the assigned times in Section 4. If the implementation is delayed, this will lead to overwork in later phases of the project such as the code verification and the application on wave systems such as the experiments. Leading to problems with time management and deadlines.

6 Time allocations and Gantt chart



TFG	start	end
TFG	**12/07/21**	**09/01/22**
Previous Work	**12/07/21**	**01/09/21**
Research quantum physics	12/07	08/08
Research Monte Carlo	09/08	27/08
Create working environment	28/08	01/09
Implementation	**03/09/21**	**01/10/21**
Create 1-D box	03/09	04/09
Periodic boundary contitions	06/09	10/09
Difference between contributions	12/09	17/09
Metropolis algorithm	18/09	01/10
Derive observable equations	03/10	05/10
Pair distribution function	06/10	13/10
Drift force	15/10	27/10
Ground-state energy	29/10	11/11
Code verification	**15/10/21**	**27/11/21**
Test pair distribution function	15/10	22/10
Test ground-state energy	14/11	22/11
Statistical error	14/11	17/11
Test finite size effects	24/11	27/11
Dipolar system	**30/11/21**	**09/01/22**
Define new wave function	30/11	30/11
Binary search algorithm	03/12	09/12
Simulate using the wave function c...	10/12	09/01
Project management	**02/09/21**	**26/01/22**
Scope	02/09	07/09
Planification	08/09	16/09
Documentation	20/09	20/01
Preparation for the defense	23/01	26/01

Figure 1: Gantt chart (own elaboration using teamgantt.com web service)

7 Economic Management

In this section the economic aspect of the project will be discussed. We will begin by identifying the costs of human resources expended in the project, followed by the generic costs given our expended materials and tools. At the end of the section, we define a final cost budget applying a contingency method for expenses that can not be predicted.

7.1 Identifying the costs of human resources

Based on the project roles assigned in Section 4.2, and hours allocated Section 4.4, we can identify the costs of this project using the average salaries of each role. These salaries are obtained using a service of Glassdoor in Spain, which is a worldwide leader on insights about jobs and companies, and possesses data

on millions of job offers, per their sources. Table 2 provides the year salaries for the different roles, and those have been broken down to a cost per hour.

Human resources costs		
Role	**Yearly salary** (€)	**Cost per hour** (€)
Project Manager	47,500	22.83
Researcher	35,386	17.01
Programmer	27,695	13.31
Tester	23,671	11.38
Analyzer	31,427	15.10
Specialist	31,437	15.11

Table 2: Human resources costs table. Yearly salaries extracted from www.glassdoor.es/index

Once we have the costs per hour for every role, extracting the working hours from Tab. 1 and thus obtaining Tab. 3 which shows the total planned cost per role for this project .

Cost per role		
Role	**Hours worked** (€)	**Total cost** (€)
Project Manager	132	3.013,56
Researcher	108	1.837,08
Programmer	297	3.953,07
Tester	20	227,6
Analyzer	120	1.812
Specialist	44	664,84
-	-	11.544,15

Table 3: Costs per role (own sources)

At the end, we obtain the total cost for the human resorces $C_{human} = 11.544,15€$.

7.2 Identifying the generic costs

The whole project can be made using a personal computer. Taking into account the monthly electric bill and internet expenses, as well as subscription services such as Wolfram Mathematica we create Tab. 4 showing an accurate representation of the generic costs of the project.

Generic costs		
Resource	**Monthly subscription type cost** (€/month)	**Total Cost for 7 months** (€)
Personal computer	-	500,00
Electric expenses	50	350,00
Internet expenses	50	350,00
Wolfram Mathematica	9	61,00
-	-	1.261,00

Table 4: Generic costs (own sources)

Thus, from Tab 4 we obtain total generic costs of $C_{generic} = 1.261,00€$.

7.3 Total cost and contingency

The total costs of the project obtained in this section are given in Euro (€) by the expression

$$C_{total} = C_{human} + C_{generic} = 12.805,15\ . \tag{4}$$

Applying to C_{total} a contingency method to provision for a possible event or circumstance, we fix 15% more of expenses. This translates into a decisive final planned cost of $C = 14.725,92€$.

8 Sustainability report

This section provides a sustainability report divided in a brief self-assessment, a description of the economic dimension of the project, followed by a discussion on the ambiental dimension, and lastly we conclude explaining the social dimension of this Bachelors Degree Thesis.

8.1 Self-assessment

During the course of the degree we have been repeatedly told about sustainability. Digging deep on the subject via courses and presentations. That said, the focus has mostly been on the environmental dimension of sustainability, due its obvious importance. To a lesser degree we have touched on the exploitation and viability of economic plans, the social impact and the risks inherent to both of them.

Sustainability has become one of the most important matters in the IT world nowadays. The sheer making of a useful product or tool is not enough anymore. In order to survive, businesses have had to adapt by adopting more economically and environmentally sustainable models. The social implications are also a big factor that companies can not ignore, thus for decades the social factors have been subject of studies for business purposes.

To sum it up, having a good sustainability strategy is almost mandatory for a business, a researcher, a programmer and any neighbour field to succeed and grow over time.

8.2 Economic dimension

The project has been put into production for economic purposes through the evaluation of the human and generic costs needed to achieve its goals. The roles involved have been clearly defined as a project manager, a researcher, a programmer, a tester, an analyzer and a specialist. The economic dimension has been well accommodated within a total cost of $C = 14.725,92€$, including the application of a preventive contingency method fixing 15% extra cost for unforeseen expenses (Section 7).

The viability plan for the project consists on an opportunity given by the lack of an exact many-body simulator to calculate the properties of a quantum system with a wave function defined to simulate experimental systems.

Current and future researchers might be able to use this implementation for their niche articles related to the quantum mechanics fields, especially bosonic gases applied with contact interactions.

8.3 Environmental dimension

Despite being a project without any major environmental repercussion, it is important remember that every project is fueled by the consumption of natural resources.

We opted for a rather low consumption design. The whole project can be carried out using a personal computer, therefore the consumption lays on its hardware materials, the electric consumption from the powering it, the internet connectivity and the services used, such as mathematical tools.

It is hard to quantify the impact of making this project possible has had on the environment, but it can be said that the final outcome of the project is a code that provides results and is contained within a script. As for the personal computer used to code it, it is still available for future use. Therefore, I assume that the most environmental consequences are due electric consumption.

8.4 Social dimension

At a personal level, this project is an important feat for a future computer scientist. It has helped me progress, not only on a technical level but also forcing me to work in a field rather new to me, the quantum mechanics. This has given me the confidence to take on any challenge. A computer scientist who works on real world cases will have to take on some problems which he has little knowledge about, and force himself to understand them to deep level. Hence, he will have to discover how to solve problems regardless their difficulty.

This project, has a multidisciplinary approach, combining computer science and quantum mechanics to achieve a common goal.

Quantum mechanics' researchers who may need this kind of simulations can use the outcomes of this project to work on their investigations.

Part III
Model, implementation and results

9 Introduction

In this part, we will cast light on the theoretical model followed in this project. The Monte Carlo algorithm will be explained, going into detail with our code implementation for the Metropolis algorithm, a Monte Carlo variant.

Following, the different observables in the project, i.e. the physical quantities that can be measured, will be explained thoroughly from a theoretical viewpoint.

Next, our Monte Carlo code will be verified through a simulation of an ideal Fermi gas, a gas whose particles are defined by an exact wave function. Thus, obtaining simulation results that can be analytically contrasted.

Lastly, we will do a practical application of the code to reproduce experimental results obtained in bosonic gases. And conclude with a simulation of the dipolar systems used in the experiments to reproduce their results [4].

10 Model

In quantum mechanics the delta potential is a potential well described by the Dirac delta function mathematically. Qualitatively, it relates to a potential $V(x)$ which is zero everywhere except at a single point, where it takes infinite value. This model can be used to simulate one dimensional situations where the particles are free to move horizontally within a box, and cannot escape. Simulating a box size L, the potential is zero

$$V(x) = 0 \, , \tag{5}$$

in the positions of the phase space inside the box, $0 < x < L$. And goes to infinity

$$V(x) = \infty \, , \tag{6}$$

at the walls of the box i.e for $x < 0$ or $x > L$. We assume infinite potential to impose that the particles have zero probability of leaving the box or even hitting the walls. The solutions to this problem gives possible values of the energy E and the wave function ψ, which when squared denotes the probability of locating a particle at a certain position within the box at a given energy level, that is

$$p(x) = |\psi|^2 \, . \tag{7}$$

Thus, our wave function has to fulfill that the probability of finding the particle at $x = 0$ or $x = L$

$$p(0) = p(L) = 0 \tag{8}$$

is zero.

The time-independent Schrödinger equation for a particle of mass m moving in one direction with energy E is

$$\left[-\frac{\hbar^2}{m}\frac{\partial^2}{\partial x^2} + V(x) \right] \psi(x) = E\psi(x) \, , \tag{9}$$

where $\hbar$ is the reduced Planck Constant ($\hbar = h2\pi$), $\psi(x)$ is the stationary time-independent wave function and $V(x)$ is the potential as a function of particle position. [9]

10.1 One-dimensional box

One-dimensional (1D) quantum systems have received much attention during the past many decades. This is due to the fact that quantum effects are more pronounced in reduced dimensions. [10]

In order to simulate an infinitely large system, we consider a finite-size homogeneous system in a box and apply periodic boundary conditions. We denote the size of a 1D box as L, so that partice positions are allowed to take any value between any two numbers, as $[0, L]$

$$L = \frac{N}{n} \, , \tag{10}$$

where N is the number of particles and n is linear density, that is the measure of a quantity of any characteristic value per unit of length.

10.2 Dimensionless units

For doing simulations with a computer one needs to introduce dimensionless units as the numbers stored by a computer are intrinsically dimensionless. Dimensionless units form a system of units where a physical quantity can be converted to an equivalent dimensionless quantity by way of scaling factor. For example, for expressing a distance x (measured in meters) one has to select unit length x_0 (also measured in meters) such that one can convert this to an equivalent dimensionless quantity x' (has no units)

$$x' = \frac{x}{x_0} \iff x = x' x_0 \, , \tag{11}$$

which is convenient for performing computer simulations in which can always be converted back to real physical units at any time.

In the following we use the inverse of the linear density $x_0 = n^{-1}$ as a unit of length. The energy values are expressed using dimensionless units of $\frac{\hbar^2 n^2}{m}$.

10.3 Combining these concepts

For clarity, we propose the following example. Given a linear density of $nx_0 = 1$ and a certain number of particles (for example $N = 10$) we obtain using Eq. (10) a box with an infinite amount of positions and a size of $L = 10n^{-1}$. The N particles have each position inside the box. The distance between any two particles with positions x, x' is $|x - x'|n^{-}1$.

10.4 Periodic boundary conditions

Our goal is to simulate an experiment with a rather large amount of particles which can not be directly simulated in polynomial time using Monte Carlo. To avoid this issue we make use of peridic boundary conditions to approximate a significantly major system using a smaller one. Thus, the particles are simulated in a box of size L with periodic boundary conditions.

In order to do so we ensure that any particle that would leave the 1D box from one side, will reenter the box from the other side. Having a box of length L the periodic boundary conditions can be defined by two constrictions.

First, we define the minimal distance with respect to image e.g. for two particles x, x' their distance applying periodic boundary conditions $d = |x - x'|$ has to be at most half the size of box

$$pbcDist(d) = \begin{cases} d > .\frac{L}{2}, & d - L \\ d < -.\frac{L}{2}, & d + L \ . \end{cases} \tag{12}$$

Second, enforcing periodic boundary conditions on particle movement. e.g let a particle with position x and a random real number ξ form a new position $x' = x + \xi$ for the particle. We must assure that it is inside the box

$$pbcMov(x') = \begin{cases} x' > L, & x' - L \\ x' < 0, & x' + L \ . \end{cases} \tag{13}$$

The implementation for these constraints can be read as a pseudo-code Alg. 1 and Alg. 2 for Eqs. (12,13) respectively.

11 Monte Carlo method

Monte Carlo simulations [11] are used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables.

The base idea is having a sequence $\{x_1, x_2, ..., x_N\}$ with the desired probability density function

$$p(x_1, x_2, ..., x_N) = \frac{|\psi(x_1, x_2, ..., x_N)|^2}{\int |\psi(x_1, x_2, ..., x_N)|^2} \ , \tag{14}$$

which will allow evaluation of integrals for the expectation value for any observable A

$$\langle A \rangle = \int A(x_1, x_2, ..., x_N) p(x_1, x_2, ..., x_N) dx \ , \tag{15}$$

by way of Monte Carlo sampling.

Since the probability of finding a quantum system in a given state is obtained by the square modulus of the wave function, we can apply Eq.(14) to calculate the average of a certain observable A (15) according to

$$\langle A \rangle = \frac{\int A(x_1, x_2, ..., x_N) \ |\psi(x_1, x_2, ..., x_N)|^2 \ dx}{\int |\psi(x_1, x_2, ..., x_N)|^2 \ dx} \ . \tag{16}$$

As argued before, the probability distribution $p(x_1, x_2, ..., x_N)$ can be sampled from using Monte Carlo which generates a sequence of configurations of the system distributed in consonance with the probability distribution function.

A collection of M configurations drawn from $p(x_1, x_2, ..., x_N)$ allows evaluation of any observable according to

$$\langle A \rangle \approx \frac{1}{M} \sum_{i=1}^{M} A(x_i) \ . \tag{17}$$

In conclusion, any observable of interest can be computed using Eq.(17) once the trial wave function has been chosen. This means the importance falls on choosing a trial wave function that allows us to obtain all the key observable values we may need.

A usual choice in the study of quantum bosonic fluids — such as the ultracold atomic gases used in recent experiments — is to consider a wave function with the form

$$\Psi(x_1, x_2, ...x_N) = \exp\left[U_1 + U_2, + ... + U_N\right] \ , \tag{18}$$

where U is some function for one-body, two-body or many-body terms. The approximate expression we use is called Jastrow wave function. It includes two-body correlations

$$\Psi(x_1, x_2, ...x_N) = \exp\left[\sum_{j<i} u_2(x_j - x_i)\right] \ . \tag{19}$$

Consistently, we apply periodic boundary conditions to $x_j - x_i$. Remark that, Jastrow form facilitates the computation of correlation functions by acting with the kinetic energy operator in the ground-state wave function. This way, one can recast the resulting terms in the form of a many-body Schrodinger Eq. (9) equation, thus identifying the parent Hamiltonian.

11.1 Metropolis algorithm

The Metropolis Algorithm [12] is Markov Chain Monte Carlo method that produces random samples from probability distributions that may otherwise be difficult to sample from directly. It gives us the opportunity to sample the distribution, so we can make histograms, or to compute an integral and obtain an expected value. For our Metropolis Algorithm we define a variable called difference between contributions Δu. Having a set of N particles in the box. Let a particle in the position x, we define its contribution $u_C(x)$ as

$$u_C(x) = \left[u_2(x_1 - x) + u_2(x_2 - x) + ... + u_2(0) + ... + u_2(x_N - x) \right] . \quad (20)$$

Considering a new candidate position x', the difference between contributions is given by

$$\Delta u = u_C(x') - u_C(x) . \quad (21)$$

Sampling using a wave function Ψ

$$\frac{p(x_1, ..., x', ..., x_N)}{p(x_1, ..., x, ..., x_N)} = \frac{\Psi(x_1, ..., x', ..., x_N)}{\Psi(x_1, ..., x, ..., x_N)} , \quad (22)$$

according to Eq.(19)

$$\frac{\exp\left[\sum_{j<i} u(x'_j - x'_i) \right]}{\exp\left[\sum_{j<i} u(x_j - x_i) \right]} = \exp\left[\sum_{j\neq i}^{N} u(x_j - x'_i) - u(x_j - x_i) \right] . \quad (23)$$

From Eqs. (21, 23) we derive

$$\Delta u = \sum_{j\neq i}^{N} \left[u(|x_j - x'_i|_{pbc}) - u(|x_j - x_i|_{pbc}) \right] . \quad (24)$$

The pseudo-code for the generation of Δu using a configuration of N points in the phase space $\vec{R}$, a current particle position $oldPos \in \vec{R}$ and a candidate position $newPos$ can be read in Alg. 4.

11.2 Metropolis algorithm implementation

Begin with a set of positions $\vec{R}$ of the N particles, $\vec{R} = \{x_1, ..., x, ..., x_N\}$, drawn randomly following a Gaussian distribution delimited between 0 and the length of the box L. That is generate, $\forall_{x_i \in \vec{R}},\ 0 < x_i < L$.

For any particle in any position x_i we introduce a random continuous variable $\xi \in (-\Delta t; \Delta t)$, where the amplitude of the displacement Δt is some floating point number. We create a new candidate position x_i' by adding ξ to the current position of the particle and applying periodic boundary conditions,

$$x_i' = pbcMov(x_i + \xi) \ . \tag{25}$$

Therefore, obtaining a new probability $\mathrm{p}(x_1, ..., x_i', ..., x_N)$ which in case is larger than the old probability,

$$p(x_1, ..., x_i', ..., x_N) > p(x_1, ..., x_i, ..., x_N) \ , \tag{26}$$

the move will be immediately accepted.

Expression (26) can be written as

$$\frac{p(x_1, ..., x_i', ..., x_N)}{p(x_1, ..., x_i, ..., x_N)} > 1 \ , \tag{27}$$

and expressed more conveniently for us by making use of Eqs. (22,23) as

$$\frac{p(x_1, ..., x_i', ..., x_N)}{p(x_1, ..., x_i, ..., x_N)} = e^{\Delta u} > 1 \ . \tag{28}$$

Observe $e^{\Delta u} > 1$ is fulfilled if, and only if $\Delta u > 0$. Thus, we simplify our first acceptance condition as: *"the move is to be always accepted when the difference between contributions is a positive value,* $\Delta u > 0$ *".*

Otherwise, our second acceptance condition assumes $\Delta u \leq 0$. In this case we consider a random floating point number $\xi' \in (0,1)$. One accepts a candidate position x_i' only if the integer part of $e^{\Delta u} + \xi'$ is equal to 1. Differently written, if

$$int[\frac{p(x_1, ..., x_i', ..., x_N)}{p(x_1, ..., x_i, ..., x_N)} + \xi] = 1 \ , \tag{29}$$

the move is to be accepted with the new probability $\frac{p(x_1,...,x_i',...,x_N)}{p(x_1,...,x_i,...,x_N)}$.

Any other outcome concludes with the position of particle remaining unchanged.

An iteration of the Metropolis algorithm is considered completed once we have repeated this process for each particle in the box. In order to add clarity to this procedure, a pseudo-code for one iteration of the Metropolis algorithm for a set of N particles contained in a vector structure $\vec{R}$, describing points in the phase space $\vec{R} = \{x_1, \cdots, x_N\}$, summarized in **Algorithm 4**.

11.3 Practical observations

The Metropolis algorithm is an iterative process to be repeated for as many iterations as desired, generating different configurations for the N particles. The cost of moving one particle is $\mathcal{O}(N)$, the cost of moving N particles is $\mathcal{O}(N^2)$.

Hence, the total cost of a complete execution of the Metropolis algorithm for M iterations is $\mathcal{O}(N^2M)$.

After each iteration the Metropolis algorithm promotes the current configuration $\vec{R}$ into a new configuration $\vec{R}'$. In order to obtain $\vec{R}'$ we only require a present set of points in the phase space $\vec{R}$. Hence, distribution of the next sample depends solely on the current coordenates. This kind of process is called a *Markovian* one, i.e. a process without memory. Consequently, the probability of making the move based on the present values is equal to the probability of the move, knowing the process' full history

$$p(x_1|x_0, x_{-1}, x_{-2}...) = p(x_1|x_0) \ . \tag{30}$$

12 Observables

An observable is a Hermitian linear operator that can operate on a system state providing information on the physical quantity. The real numbers that can be observed by measurement of an observable are encoded into the operator as the set of eigenvalues of that operator, which determine all the possible values that can be measured. In particular we are interested in the energy (corresponding operator is the Hamiltonian) and the pair distribution function $g(x)$.

12.1 Pair distribution function

12.1.1 Definition

The pair distribution function $g(x)$ is an important observable quantifying the correlations within a gas. It is defined as the distribution of distances between pairs of particles contained within a given volume. Physically, $g(x)$ is proportional to the probability of finding two particles separated by a distance x. The pair distribution function can be written as the following integral over the probability distribution,

$$g(x) = \frac{\int \sum_{i<j}^{N} \delta[x - pbcDist(x_i - x_j)]|\Psi(x_1, ..., x, ..., x_N)|^2 dx}{\int |\Psi(x_1, ..., x, ..., x_N)|^2 dx} \ . \tag{31}$$

Such an integral can be conveniently evaluated in the Metropolis algorithm. Indeed, $g(x)$ is obtained by calculating a histogram for the values of distance between particles of the system.

12.1.2 Implementation

For N particles, a linear density n and a box size L, we create a histogram representing the number of counts $\vec{h} = \{h_1...h_k\}$ with initially all values set to 0, that is, $\forall_{h_i \in \vec{h}}, h_i = 0$. The number of elements of $\vec{h}$ is given by the expression

$$N_{\vec{h}} = \frac{L}{2\Delta x} \ , \tag{32}$$

where Δx is the grid spacing of the x axis of the histogram.

For the M iterations of the Metropolis algorithm and for each pair of particles x_1, x_2, we find the bin number corresponding to its distance

$$N_{bin} = Int\left[\frac{|x_1 - x_2|_{pbc}}{n\Delta x}\right], \tag{33}$$

and increment its value by one, $\vec{h}[Nbin] = \vec{h}[Nbin] + 1$. The pseudo-code of this process applied to a configuration of points in the phase space $\vec{R}$ and a histogram $\vec{h}$ can be read in Alg. 5.

Once we have obtained the number of counts $\vec{h}$, the pair distribution function $g(x)$ is inferred from it by imposing the proper normalization. Applying to each $h_i \in \vec{h}$

$$g(x) = \frac{|\vec{h}_i|}{M\Delta x \frac{L}{2} \frac{N(N-1)}{2}(1 - \frac{1}{N})} . \tag{34}$$

12.2 Energy

Another observable of interest is the energy of the system. The total energy of a quantum mechanical system is defined as the expectation value of the Hamiltonian operator $\hat{H}$, which includes both kinetic and potential energies,

$$\hat{H} = E_{kin} + E_{pot} , \tag{35}$$

consequently making

$$E = \int \hat{H}\Psi(x_1, .., x_N)p(x_1, .., x_N)dx \tag{36}$$

the expression for its expectation value. By making use of Eq. (14) we obtain

$$E = \frac{\int \Psi(x_1, x_2, ..., x_N)\hat{H}\Psi(x_1, x_2, ..., x_N)\ dx}{\int |\Psi(x_1, x_2, ..., x_N)|^2\ dx} , \tag{37}$$

where

$$p(x_1, .., x_N) = \frac{\Psi(x_1, x_2, ..., x_N)dx}{\int |\Psi(x_1, x_2, ..., x_N)|^2\ dx} \tag{38}$$

is the probability distribution and

$$E(x_1, \cdot, x_N) = \frac{\hat{H}\Psi(x_1, .., x_N)}{\Psi(x_1, .., x_N)} \tag{39}$$

is the locale energy, which depends on N coordinates. In order to obtain the ground-state energy of the system we have to obtain the drift force to calculate the kinetic energy.

12.2.1 Drift force

The drift force of a particle F_i is proportional to the gradient of the wave function at position x_i, and the drift force is important for the calculation of the kinetic energy.

It is defined as a vector pointing in the direction of the gradient of the wave function Ψ, in other words in the direction at which the wave function Ψ increases at a maximal rate in the phase space composed of $(x_1, \cdots, x_N)$ variables. Its i-th element is obtained as a logarithmic derivative with respect to x_i,

$$F_i = \frac{\nabla_i \Psi(x_1, ..., x_i, ...x_N)}{\Psi(x_1, ..., x_i, ...x_N)} = \frac{\nabla_i \exp\left[\sum_{i<j}^{N} u_2(|x_i - x_j|)\right]}{\exp\left[\sum_{i<j}^{N} u_2(|x_i - x_j|)\right]} \tag{40}$$

By applying first derivative operator $\nabla_i = \frac{\partial}{\partial x_i}$ we obtain

$$F_i = \sum_{i<j}^{N} u_2'(|x_i - x_j|) \frac{x_i - x_j}{|x_i - x_j|} \ , \tag{41}$$

where $u_2'(x)$ is given by the first logaritmic derivative of the Jastrow term.

The drift force F_i is used to obtain the expectation value of the kinetic energy. Algorithm 6 exhibits the pseudocode for the calculation of the particle drift force during any iteration of the Metropolis algorithm, given a configuration $\vec{R}$ formed by N points in the phase space $x_1, \ldots, x_N$.

12.2.2 Kinetic energy

The kinetic energy is the energy a particle possesses from its quantum motion.

$$E_{kin}(x_1, \cdots, x_N) \ , \tag{42}$$

where $E_{kin}(x_1, \cdots, x_N)$ denotes the local kinetic energy. The kinetic energy operator E_{kin} is defined in terms of the Laplacian, which is an operator corresponding to the second derivative with respect to (x, y, z). In one dimension it can be simplified to $\Delta = \frac{\partial^2}{\partial x^2}$. Therefore, we adapt expression (39) for our simulations to

$$E_{kin}(x_1, \cdots, x_N) = \frac{\Delta \Psi(x_1, .., x_N)}{\Psi(x_1, .., x_N)} \ . \tag{43}$$

The ground-state energy is then given by the expectation value of the local energy

$$E = \langle E_{kin}(x_1, \cdots, x_N) \rangle \ . \tag{44}$$

The kinetic energy can be calculated using two different estimators. The first one

$$E_{kin}^{(1)} = -\frac{\hbar^2}{2m} \left\langle \sum_{i=1}^{N} F_i^2 + \sum_{j \neq i} u_2''(|x_i - x_j|) \right\rangle \ , \tag{45}$$

is obtained by applying directly the Laplacian to the many-body wave function, where $u_2''(|x_i - x_j|)$ is given by the second logarithmic derivative of the Jastrow term.

Alternatively, the kinetic energy can be approximated as the average value of the square of the drift force,

$$E_{kin}^{(2)} = \frac{\hbar^2}{2m} \left\langle \sum_{i=1}^{N} F_i^2 \right\rangle . \tag{46}$$

although this estimator has a much larger variance.

The pseudo-codes for calculation of the ground-state energy in units of $\frac{\hbar^2 n^2}{m}$ using estimators $E_{kin}^{(1)}$ and $E_{kin}^{(2)}$ are provided in Algs. 7 and 8, respectively. Here $\vec{R}$ is a vector representing a configuration and $driftForce$ is a vector representing the drift force of the system obtained a from a call to Alg. 6.

13 Code verification

Once the Metropolis algorithm is implemented it is important to verify the correctness of the implementation by running a certain test. For such a test we consider the simulation of an ideal Fermi gas in one dimension. An ideal Fermi Gas is a gas formed by fermions with no interaction. Fermions are quantum particle, which have half-integral spin and follow the statistical description given by Fermi-Dirac statistics. For ideal particles the interaction with other particles is neglected. In order to simulate the ideal Fermi gas we note that the Jastrow term given in Eq.(19) is known exactly [13] in that case and can be written as,

$$u(x) = \ln \left| \sin \frac{\pi x}{L} \right| , \tag{47}$$

and will be used for our wave function to simulate an ideal Fermi gas system in a one-dimensional box.

The drift force can be obtained through Metropolis sampling using Eq. (41) where $u_2'(x)$ is given by the first logaritmic derivative of the Jastrow term for the ideal Fermi gas Eq. (47),

$$u_2'(x) = \frac{\pi}{L \tan\left(\frac{\pi x}{L}\right)} . \tag{48}$$

As for the kinetic energy, it can be obtained during the simulation for both kinetic estimators Eq. (45,46), where $u''(x)$ is given by the second derivative of the Jastrow term function. In the case of the ideal Fermi gas that is

$$u_2''(x) = \left(\frac{\pi}{L}\right)^2 . \tag{49}$$

13.1 Pair distribution function of the ideal Fermi gas

For the ideal Fermi gas the pair distribution function is known exactly [13] and can be written explicitly as

$$g(x) = 1 - \left(\frac{\sin \pi n x}{\pi n x}\right)^2 . \tag{50}$$

We performed a simulation for $N = 10$ particles, a linear density of $n = 1$ and range of distance between points in the histogram of $\Delta x = 0.1$ obtaining Fig. 2. The expectation values for $g(x)$ of our Metropolis algorithm implementation are represented by the purple points while the blue line is obtained through the evaluation of Eq. (50).

We can see how the expectation values of our implementation follow correctly the behaviour of the explicitly calculated ones.

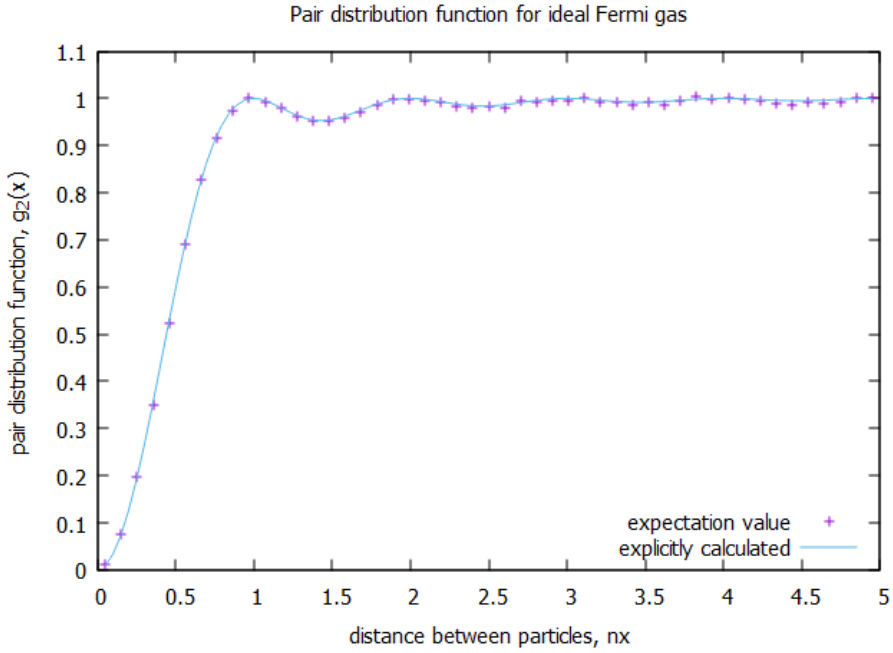


Figure 2: Plot obtained through the simulation of an ideal Fermi gas system with $N = 10$ particles, a linear density of $n = 1$, a grid space of $\Delta x = 0.1$, and a box size $L = 10$. The expectation values for $g(x)$ by way of Metropolis algorithm are represented by the purple points while the blue line is obtained through the evaluation of the ideal Fermi gas' exact pair distribution function given by Eq. (50).

13.2 Statistical error for the energy given independent measurements

Given M iterations of our Metropolis algorithm with independent measures of the kinetic energy operator E_{kin}, we have to take into account a statistical error. The expectation value of the kinetic energy becomes a value averaged over M different iterations, $E_1 \cdots E_M$, that is:

$$\langle E \rangle \approx \frac{1}{M} \sum_{i=1}^{M} E_i \equiv \overline{E}_{num} \ , \tag{51}$$

as a consequence

$$E_{kin} = \overline{E}_{num} \pm E_{Err} \tag{52}$$

becomes our energy estimator over independent measurements. It is very important to estimate the statistical error

$$E_{Err} \approx \frac{\sigma}{\sqrt{M}} \tag{53}$$

in the resulting energy, where

$$\sigma^2 = \langle (E_{kin} - \overline{E}_{num})^2 \rangle \approx \frac{1}{M} \sum_{i=1}^{M} E_i^2 - \overline{E}_{num}^2 \tag{54}$$

having a mean-square variance σ.

13.3 Kinetic energy results

The energy per particle of an ideal Fermi gas system can be calculated analytically,

$$\frac{E}{N} = \frac{\pi^2}{6} \frac{\hbar^2 n^2}{m} \tag{55}$$

in the thermodynamic limit.

For ideal fermions that do not interact with each other, the potential energy is equal to zero Eq. (35),

$$E_{pot} = 0, \tag{56}$$

making the total energy of the system equal to its energy of quantum motion,

$$E(x_1, \cdots, x_N) = E_{kin}(x_1, \cdots, x_N) \; . \tag{57}$$

Now, to prove the correctness of our Metropolis algorithm implementation we will compare the results obtained with it with the ones obtained analytically using Eq.(55). As mentioned before we can calculate the total energy using two different estimators Eq. (45,46) taking into account statistical error. For $N = 10$ particles and a linear density of $n = 1$, we obtain analytically $E/N = 1.64493 \frac{\hbar^2 n^2}{m}$. For $M = 100000$ iterations of Metropolis algorithm, we calculated the first estimator $E_{kin}^{(1)}$, obtaining an average value of $E/N = 1.64 \frac{\hbar^2 n^2}{m}$. Within the same execution the second estimator $E_{kin}^{(2)}$ obtained $E/N = 1.60 \frac{\hbar^2 n^2}{m}$. The statistical error given by the evaluation of Eq.(53) added up to $E_{Err} \approx 0.04 \frac{\hbar^2 n^2}{m}$, so that the second estimator is compatible with the exact result within the errorbars.

Figure 3 shows a typical example of the calculations of the two estimators for the energy per particle. The purple line corresponds to the first estimator $\langle E_{kin}^{(1)} \rangle$ which is constant as the used wave function is exact. The blue points represent the second estimator $\langle E_{kin}^{(2)} \rangle$ which shows large fluctuations which, when averaged, result in the same expectation value for the ground-state energy within the statistical error.

13.4 Finite-size effects

Another interesting test consists in calculating the energy per particle E/N as a function of the density n. We calculated the energy per particle numerically

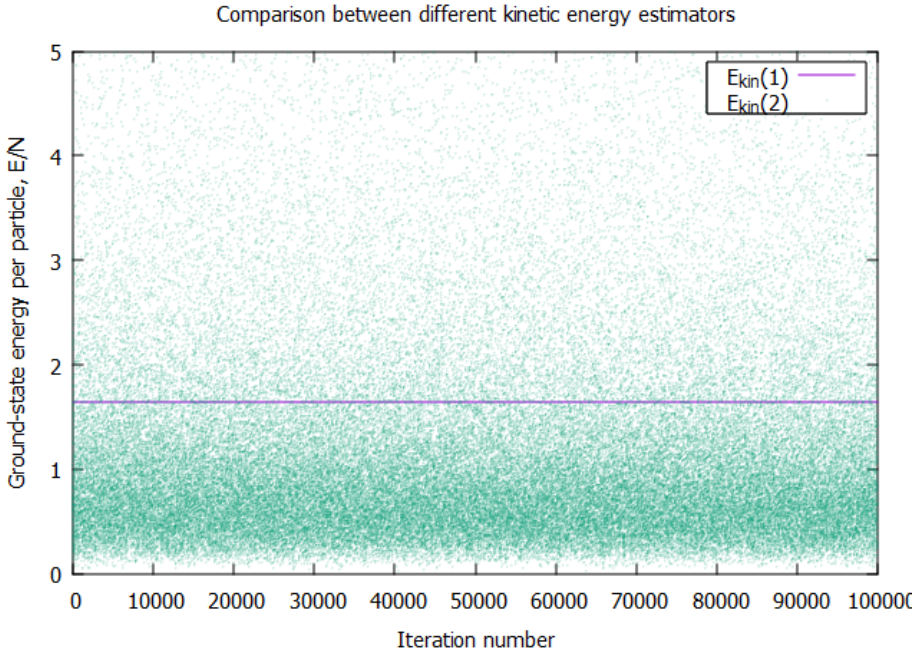


Figure 3: Plot obtained by doing $M = 100000$ iterations, and sampling on each iteration the kinetic energy using estimators $E^{(1)}_{kin}$ and $E^{(2)}_{kin}$. The system consisted of $N = 10$ particles and had a linear density of $nx_0 = 1$. The purple line correspond to the independent values that make up $\langle E^{(1)}_{kin} \rangle$ while the blue points represent the ground-state energy per particle for calculations using $\langle E^{(2)}_{kin} \rangle$. The first estimator $E^{(1)}_{kin}$ showed an average value for the ground-state energy per particle of $E/N = 1.64\hbar^2 n^2/m$. Within the same execution the second estimator $E^{(2)}_{kin}$ obtained $E/N = 1.60\hbar^2 n^2/m$. The statistical error given by the evaluation of Eq. (53) added up to $E_{Err} \approx 0.04\hbar^2 n^2/m$, so that the second estimator is compatible with the exact result within the errorbars.

using the first estimator $E^{(1)}_{kin}$ in a wide range of densities, $1 \ldots 10$. The obtained result are shown in Fig. 4 where the purple points represent the estimated values for the kinetic energy using Metropolis algorithm and the blue line is given by the analytical expression for the energy per particle of an ideal Fermi gas Eq. (55). We find a perfect agreement between the numerical data and the exact analytical theory.

We also test the finite-size correction in the energy. The calculations are performed in a box of a finite size, with a limited number of particles. As a result, there is a certain difference in the energy as opposed to the thermodynamic one. Analytically, it is possible to calculate the finite-size correction to the ground-

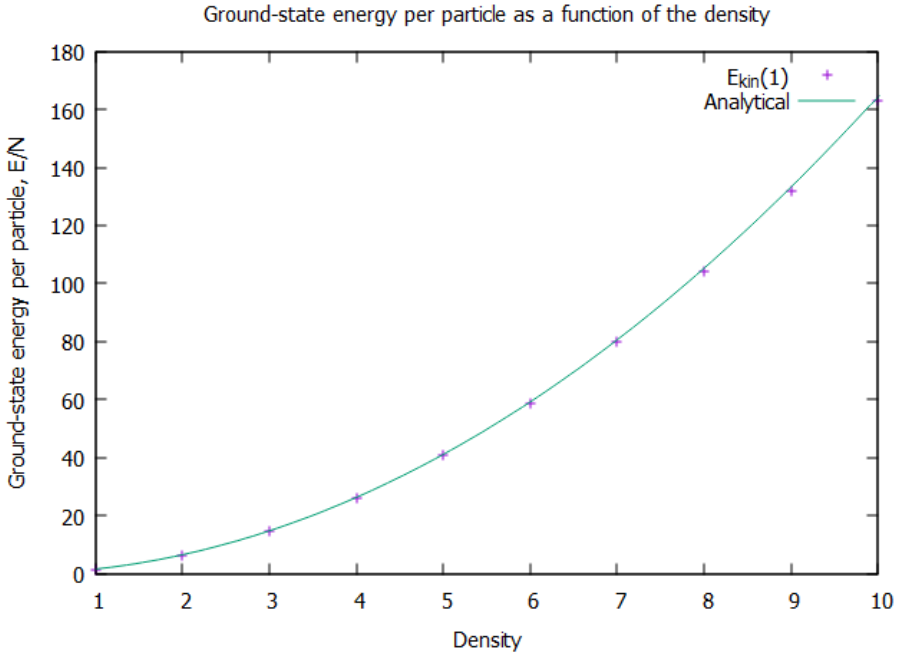


Figure 4: Plot representing the energy per particle E/N with units of $\hbar^2 n^2/m$ in function different values of the linear density n. For number of particles $N = 10$, the purple points are obtained through the calculation of $E^{(1)}_{kin}$ using Metropolis algorithm, while the blue line given by the evaluation of the analytical expression for the energy per particle of an ideal Fermi gas, Eq. (55).

state energy, Eq. (55)

$$\frac{E}{N} = \frac{\pi^2}{6}\frac{\hbar^2 n^2}{m}\left(1 - \frac{1}{N^2}\right) \tag{58}$$

which has $1/N^2$ convergence.

Figure 5 shows an example of the finite-size dependence of the ground-state energy E/N as a function of the inverse number of particles, $1/N$. The thermodynamic value, $N \to \infty$, corresponds to the limit of the zero argument, $1/N \to 0$. The purple points in Fig. 5 represent the values obtained using Metropolis algorithm while the blue line corresponds to the exact result, as provided by Eq. (58). We find a perfect agreement thus validating the implementation of the algorithm.

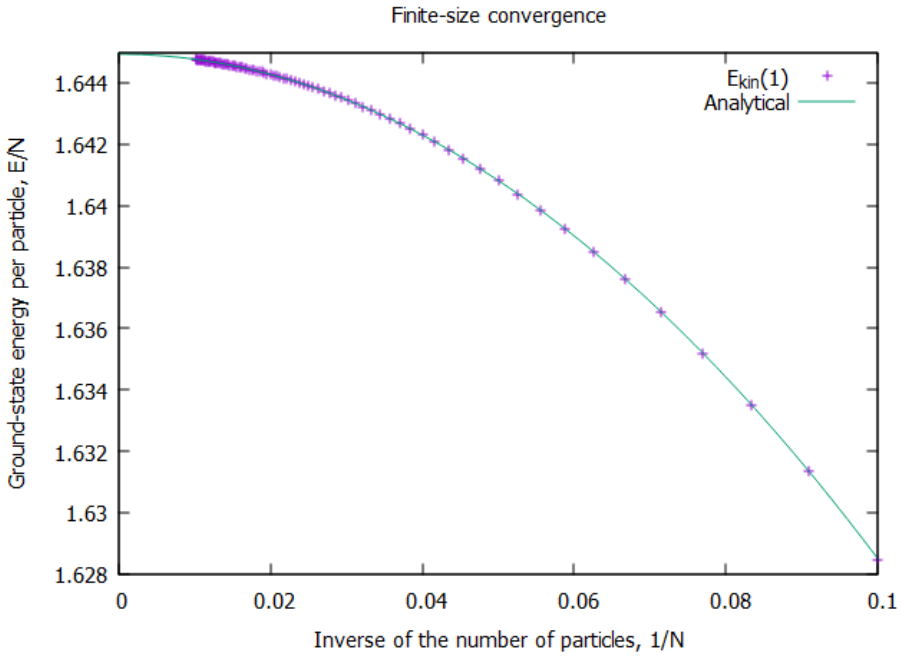


Figure 5: Plot representing the finite size convergence for the ideal Fermi gas using the energy per particle E/N with units of $\hbar^2 n^2/m$ in function the inverse of the number of particles $1/N$. For number of particles $N = 10$ and density $n = 1$, the purple points are obtained through the calculation of $E^{(1)}_{kin}$ using Metropolis algorithm, while the blue line given by the evaluation of the analytical expression for the finite-size correction with a convergence of $1/N^2$, Eq. (58)

14 Wave function corresponding to the experiment with dipolar atoms

Once we have proved the correctness of our implementation of the Metropolis algorithm, we want to reproduce experimental result obtained in dipolar systems. In order to do so we have to work with an appropriate wave function.

We construct the Jastrow function as a solution of the two-body Schrödinger equation,

$$\left[-\frac{\hbar^2}{m}\frac{\partial^2}{\partial x^2} + V(x) \right] \psi(x) = E\psi(x) \tag{59}$$

We assume contact interaction potential, $V(x) = g\delta(x)$, where $\delta(x)$ is Dirac's delta function. For $|x| > 0$ one has $V(x) = 0$ and free particle differential equation applies, $-(\hbar^2/m)\psi'' = E\psi$. Thus, the solution can be written as a sum of plane waves [13]

$$\psi(x) = A\cos kx + B\sin kx \ , \tag{60}$$

where the scattering momentum k is related to the two-body scattering energy E according to $E = \hbar^2 k^2/m$. Equation (60) can be conveniently written in terms of the phase shift δ

$$\psi = C\cos(kx + \delta) \ , \tag{61}$$

and scattering momentum k.

The many-body wave function is than given by the Jastrow pair-product form,

$$\Psi(x_1, x_2, ...x_N) = \exp\left[\sum_{j<i} u(x_j - x_i) \right] \tag{62}$$

Note that $\delta = \frac{\pi}{2}$ and $k = \frac{\pi}{L}$ for the ideal Fermi gas giving us same results as Eq. (47).

For a particle position x we have to find k and δ to solve Ψ. In order to do this we will rely on two boundary conditions.

At short distances, the Bethe-Peierls boundary condition [14] applies

$$\frac{\Psi'}{\Psi}\Big|_{x=0} = -\frac{1}{a} \ , \tag{63}$$

where a is the s-wave scattering length with units of $a = \tilde{a}n$. And the scattering length $a = \frac{2}{g}$ where

$$g = -\frac{2\hbar^2}{ma} \tag{64}$$

is the coupling constant.

At large distances, the periodic boundary condition must be satisfied,

$$\Psi'\left(\frac{L}{2}\right) = 0 \ , \tag{65}$$

so that the wave function is maximal at half size of the box, $\Psi(\frac{L}{2})$.

Therefore, there are two unknown numbers to be defined, i.e the scattering momentum k and phase shift δ, and their values can be inferred from Eqs. (63, 65) for any value of the s-wave scattering length and the size of the box: $k(a, L), \delta(a, L)$.

14.1 Deriving the scattering momentum and the phase shift

In order to obtain k and δ from Eqs. (63, 65) we start with the Jastrow term

$$f(x) = \cos\left(kx + \delta\right) , \tag{66}$$

and differntiate it in order to obtain its first derivative

$$f'(x) = -k \sin\left(kx + \delta\right) , \tag{67}$$

and its logarithmic derivative

$$\frac{f'(x)}{f(x)} = -k \tan\left(kx + \delta\right) . \tag{68}$$

which enters into the expression of the drift force. Lastly the second logarithmic derivative, needed for the calculation of the ground-state energy is given by

$$(\frac{f'(x)}{f(x)})' = -k^2 - \left[- k \tan\left(kx + \delta\right)\right]^2 . \tag{69}$$

Periodic boundary condition (65) at $x = \frac{L}{2}$,

$$\frac{f'(L/2)}{f(L/2)} = -k \tan\left(k\frac{L}{2} + \delta\right) = 0 , \tag{70}$$

is satisfied when $\tan(k\frac{L}{2} + \delta) = 0$ that is if, only if

$$k\frac{L}{2} + \delta = \ell\pi , \tag{71}$$

where an integer number ℓ is the ramification index. Its specific value $\ell = 1, 2, ...$ together with the sign of a defines the topological number n_t, i.e. the number of nodes of the Jastrow function $f(x)$.

Isolating k, we obtain the first condition for the scattering momentum

$$k = \frac{2}{L}(\pi\ell - \delta) . \tag{72}$$

On the other hand applying Bethe-Peierls boundary condition given by Eq. (63) to our wave function

$$\frac{f'(x)}{f(x)}\big|_{x=0} = -k \tan\delta = -\frac{1}{a} , \tag{73}$$

we can also isolate the scattering momentum, obtaining second condition for the scattering momentum

$$k = \frac{1}{a \tan \delta} \ . \tag{74}$$

The system of Eqs. (72,74),

$$k = \frac{2}{L}(\pi\ell - \delta) = \frac{1}{a \tan \delta} \ , \tag{75}$$

define a single equation on the value of the phase shift δ

$$\frac{2a}{L}(\delta - \pi\ell) + \cot \delta = 0 \ , \tag{76}$$

where $0 \leq \delta \leq \pi$.

In the limit of strong repulsion, $g \to +\infty$ or $a \to -0$, the phase shift is equal to

$$\delta = \frac{\pi}{2} \tag{77}$$

and the scattering momentum is

$$k = \pi(2\ell - 1)/L \tag{78}$$

In the vicinity of that point, when $|a|/L \ll 1$, the phase shift can be expressed in terms of a small correction around its limiting value (77) as

$$\delta = \frac{\pi}{2} + \tilde{\delta} \tag{79}$$

where $|\tilde{\delta}| \ll \delta$. Thus, the cotangent function can be expanded around this point, $\cot \delta = \cot(\pi/2 + \tilde{\delta}) = -\tilde{\delta} + \mathcal{O}(\tilde{\delta}^3)$, allowing to express explicitly change in the phase shift due to a finite value of a/L, according to

$$\tilde{\delta} = -\frac{\pi}{2} \frac{2a}{L - 2a}(2\ell - 1) \tag{80}$$

The phase shift itself is than can be expressed as

$$\delta = \frac{\pi}{2} \left[1 - \frac{2a}{L - 2a}(2\ell - 1) \right] \tag{81}$$

14.2 Implementation of a dichotomic search for the phase shift

A dichotomic search is a type of divide and conquer algorithm which operates by selecting one of two alternatives at each step.

For a given s-wave scattering lenght a and a ramification index ℓ related to the topological number representing the number of nodes of the wave n_t, we have apply a dichotomy on Eq. (75) to find a phase shift δ. Inspired in Eq. (76) we made a function

$$PDD(\delta) = -\frac{2a}{L}(\pi\ell - \delta) + \frac{1}{\tan\delta} \ , \tag{82}$$

which returns a floating point number. Note that,

$$PDD(0) = \infty \text{ and } PDD(\pi) = -\infty \ . \tag{83}$$

Making use of this we build a dichotomic search function $dS(min, max, auxAnt)$, where min, max are the minimal and maximal boundaries of δ, respectively. The variable $auxAnt$ saves $PDD(\delta_{-1})$, where δ_{-1} is the candidate value of the phase shift obtained in the previous level of the recursion. Let δ_0 be the candidate value of the phase shift obtained in the current level of the recursion. If $PDD(\delta_0) = PDD(\delta_{-1})$ then

$$PDD(\delta_0) \approx 0 \tag{84}$$

satisfying (76). Else if $PDD(\delta_0) > 0$ we set the minimal boundary of δ to δ_0, calling $dS(\delta_0, max, PDD(\delta_0))$. Otherwise, $PDD(\delta_0) < 0$, and we set the maximal boundary of δ to δ_0, calling $dS(min, \delta_0, PDD(\delta_0))$.

In order to avoid problems with the compilator dividing by zero, we define a small continuous variable $\epsilon = 10^-6$ to delimit δ . Since $0 < \delta < \pi$ we approximate to $\epsilon \leq \delta \leq \pi - \epsilon$.

The pseodo-code for the execution of $dS(min, max, auxAnt)$ function can be seen in Alg. 9, working on logarithmic time.

Once we obtained δ we can derive k by solving Eq. (72). Hence, we can now solve the Jastrow function $f(x)$.

14.3 Fitting the scattering momentum of the two-body problem

The two-body scattering problem is used for constructing the Jastrow wave function, as explained in the previous Sections. Furthermore, the scattering momentum k can be used to approximate the many-body energy E for small values of a. In particular, the dependence of the scattering momentum k on the s-wave scattering length a can be deduced numerically and we use the following

fit to approximate the obtained dependence

$$k_1 L = \pi + c_1 \frac{a}{L} \tag{85}$$

$$k_2 L = 3\pi + c_2 \frac{a}{L}, \tag{86}$$

$$k_n L = (2\ell - 1)\pi + c_n \frac{a}{L}, \tag{87}$$

where the linear coefficients c_1 and c_2 are obtained from the fitting procedure. Here, as before, k_1 correspond to the state with a ramification index $\ell = 1$, k_2, to the state with $\ell = 2$.

We can clearly observe the dependence scattering momentum has on the s-wave scattering length with the representation for values of kL as a function of a/L shown in Fig. 6. Through our implemented dichotomic search dS we obtained for $-0.5 < a < 0.5$ different values of k for both ramification indexes $\ell = 1$ and $\ell = 2$. Then in a fitting range of $[-0.1, 0.1]$ we obtained a value for our linear coefficients $c1$ and $c2$ with their asymptotic standard errors, $c1 = 6.28 \pm 0.21$ and $c2 = 16.26 \pm 0.41$.

Note that, for a s-wave scattering length $a = 0$ and a ramification index of $\ell = 1$, $kL = \pi$. While for $\ell = 2$, $kL = 3\pi$.

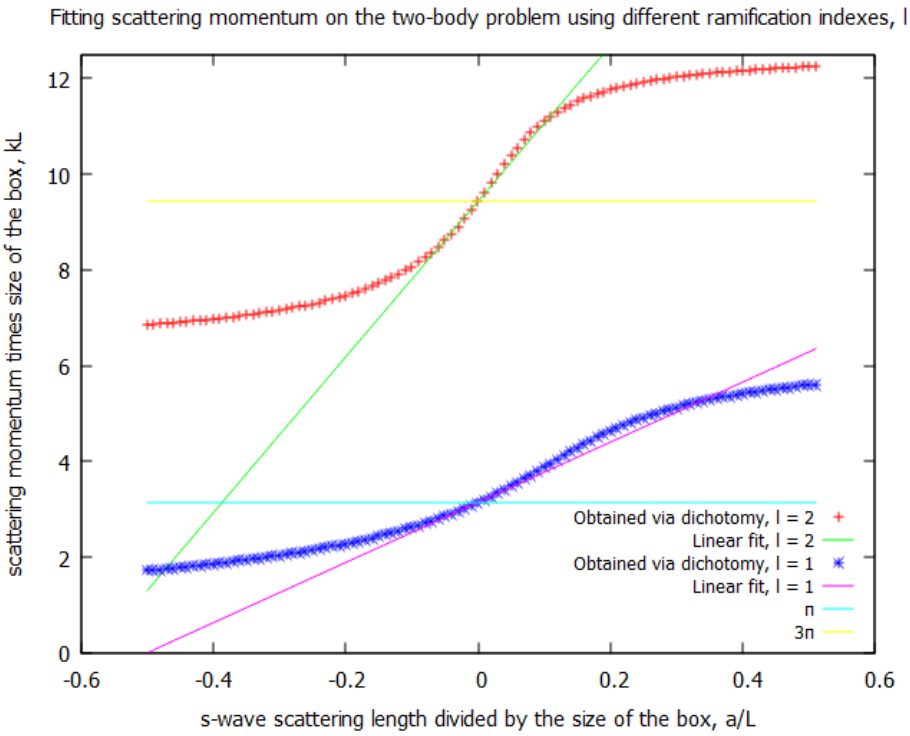


Figure 6: Scattering momentum k times the size of the box L as a function of the quotient of the s-wave scattering length a and L. The blue points are obtained via dichotomy for a ramification index $\ell = 1$, and the pink line corresponds to its linear fit using Eq. (85) where $c_1 = 6.28 \pm 0.21$. The red points are obtained via dichotomy for $\ell = 2$, and the green line represents its linear fit using Eq. (86) in the range $[-0.1, 0.1]$ where $c_2 = 16.26 \pm 0.41$. We can also observe how for $a = 0$ and $\ell = 2$ we obtain $kL = 3\pi$ while for $\ell = 1$ the results are $kL = \pi$.

15 Results

15.1 Topological excitations: dependence on the s-wave scattering length a

We have seen that the scattering momentum k and the phase shift δ that build our wave function depend on the s-wave scattering length a and the ramification index ℓ. Furthermore, a is also related to the coupling constant g which decides the strength in which the particles in the box excite or repulse each other.

We show in Fig. 7 typical examples of the Jastrow wave function for different values of s-wave scattering length a (or the coupling constant g).

- Repulsive interactions, $g > 0$, correspond to negative values of a and the

ground-state solution is nodeless. That is, for $a < 0$, the Jastrow function $f(x)$ has no nodes.

- For an infinite repulsion, $g = +\infty$, the bosonic particles become impenetrable and the wave function vanishes wherever two particles meet. In terms of the Jastrow function this corresponds to a zero value at the contact, $f(0) = 0$. In this limit, the infinitely-repulsive bosons share the energy and correlation functions with ideal fermions.

- Instead, for attractive interaction, $g < 0$, that is positive values of the s-wave scattering length, the bosonic nodeless ground-state solution corresponds to an exponentially bound state. Instead, the scattering state we are interested in, has one node, at a distance similar to $x = a$. That is, $f(x = a) \approx 0$. Thus, the solution we show in Fig. 7 for that case corresponds to an excited bosonic state.

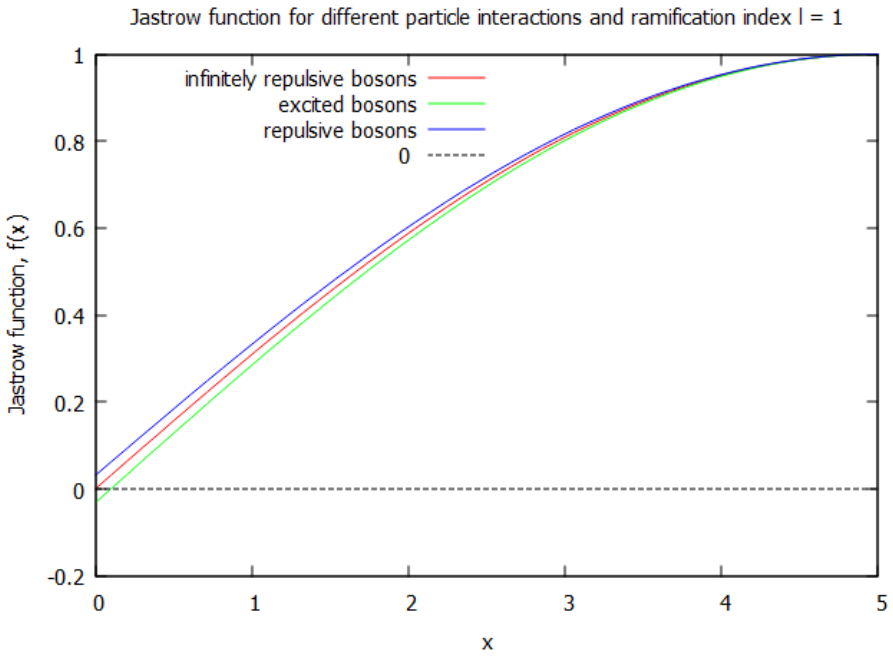


Figure 7: Jastrow function obtained using Eq. (62) for ramification index $\ell = 1$ and different values of the s-wave scattering length a: repulsive bosons ($a < 0$); infinitely repulsive bosons or fermions ($a = 0$) and excited bosons ($a > 0$). The topological number counts the number of nodes and is equal to $n_t = 0$ for $a < 0$ case and $n_t = 1$ for $a = 0$ and (excited) $a > 0$ cases.

Figure 8 provides a number of typical examples of the Jastrow function for a higher ramification index, $\ell = 2$. As a result, the number of nodes, specified

by the topological number n_t, is higher in that case.

- Indeed, for the until now nodeless repulsive interactions, the Jastrow function $f(x)$ has now to one node, $nt = 1$.
- For the infinite repulsion interactions the topological number has increased too, $nt = 2$. One of the nodes is still decided at the position $x = 0$ where for s-wave scattering length $a = 0$ the phase shift $\delta = \frac{\pi}{2}$, hence $f(0) = 0$.
- Lastly, excited bosons for $\ell = 2$ have also seen their number of nodes increased to two, $nt = 2$. One of the nodes remains given by $f(x = a) \approx 0$.

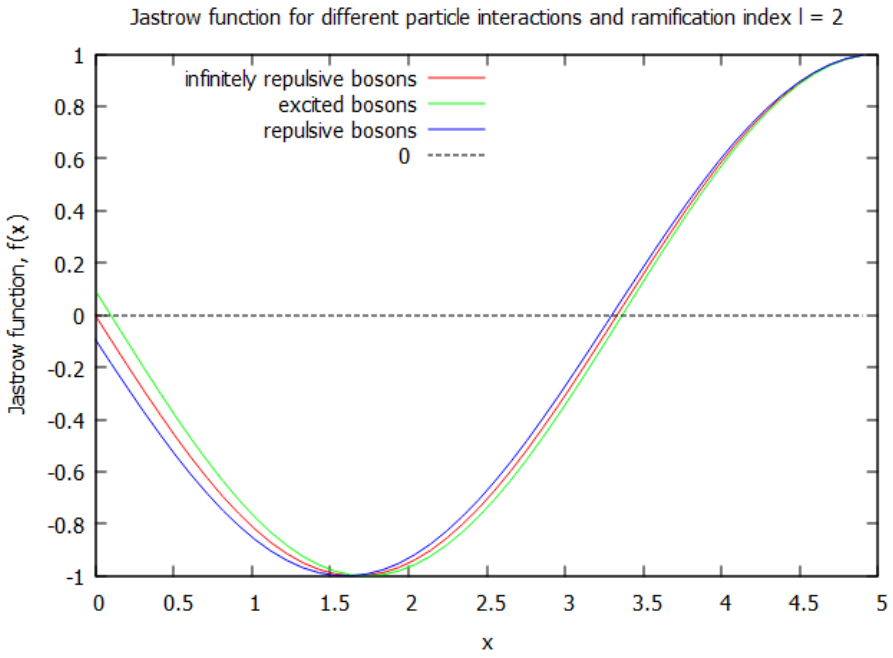


Figure 8: Jastrow function obtained using Eq. (62) and a ramification index $\ell = 2$. For repulsive bosons $a < 0$, infinitely repulsive bosons or fermions $a = 0$ and excited bosons $a > 0$. The topological number counts the number of nodes and is equal to $n_t = 1$ for $a < 0$ case and $n_t = 2$ for $a = 0$ and (excited) $a > 0$ cases.

15.2 Topological excitations: dependence on the ramification index ℓ

In this section we will discuss topological excitations fixing a value of the s-wave scattering length and having the ramification index ℓ as a variable.

We show in Fig. 9 different panels with typical examples of Jastrow function $f(x)$ for different ramification indices $\ell = 1; 2; 3$ for characteristic values of the s-wave scattering length.

- (a) $a < 0$, i.e. ground-state of repulsive Bose gas ($\ell = 0$) and its topologically excited states $\ell > 1$.
- (b) $a = 0$, impenetrable bosons, or ideal Fermi gas.
- (c) $a > 0$, i.e. excited states of an attractive Bose gas.

Using different ramification indices ℓ we can see how the number of nodes, hence the topological number n_t changes accordingly. The topological number n_t counts the number of nodes (when $f(x) = 0$) and its values depends on ℓ and the sign of a. For $a \geq 0$ the value of the ramification index is equal to the value of topological number, $\ell = n_t$.

15.3 Excluded volume corrections

The many-body energy can be calculated explicitly in the case of an ideal Fermi gas according to

$$\frac{E}{N} = \frac{\pi^2\hbar^2}{6m}\frac{N^2}{L^2}. \tag{88}$$

The same ground-state energy is recovered in a Bose gas with an infinite repulsion between atoms, $g = \infty$, or $a = 0$. It turns our that result (88) can be generalized[13] to hard-rod Bose gas, that is impenetrable particles interacting with potential such that $V(x) = +\infty, |x| < a$ and $V(x) = 0$, otherwise with $a > 0$. In that case, free solution applies whenever $|x_{ij}| > a$, but the phase space is reduced by Na, i.e. by the size of the excluded volume $L \to L - Na$. The ground-state hard rod energy is obtained by using the substitution $L \to L - Na$ in Eq. (88)

$$\frac{E}{N} = \frac{\pi^2\hbar^2}{6m}\frac{N^2}{L^2}\frac{1}{(1-\frac{Na}{L})^2}. \tag{89}$$

It is expected that the same energy approximately applies to the gas with contact interactions when

$$n|a| \ll 1 \tag{90}$$

that is when the value of the s-wave scattering length is small compared to the mean interparticle distance n^{-1}.

In order to generalize Eq. (89) by including a possibility of topological excitations we assume that $kL = \pi$, that is

$$L = \frac{\pi}{k}. \tag{91}$$

Thus, we derive the following prediction,

$$\frac{E}{N} = \frac{\pi^2\hbar^2N^2k^2}{6\pi^2m}\frac{1}{(1-\frac{Nka}{\pi})^2}. \tag{92}$$

For small values of a, such that $n|a| \ll 1$, we can use the linear fit expression (87) for the momentum and effective volume L according to

$$L_1' = \frac{\pi}{k_1} = \frac{\pi}{\pi + c_1 \frac{a}{L}} L = \frac{L}{1 + \frac{c_1}{\pi} \frac{a}{L}} \tag{93}$$

$$L_2' = \frac{\pi}{k_2} = \frac{\pi}{3\pi + c_2 \frac{a}{L}} L = \frac{L}{3 + \frac{c_2}{\pi} \frac{a}{L}} \tag{94}$$

Thus we derive the following analytical approximation for the many-body energies in function of the ramification index.

$$\frac{E_\ell}{N} = \frac{\pi^2 \hbar^2}{6m} n^2 \left((2\ell - 1) + \frac{c_1}{\pi} \frac{a}{L} \right)^2 \frac{1}{(1 - na((2\ell - 1) + \frac{c_1}{\pi} \frac{a}{L}))^2} \tag{95}$$

In particular, for the lowest two branches we have

$$\frac{E_1}{N} = \frac{\pi^2 \hbar^2}{6m} n^2 \left(1 + \frac{c_1}{\pi} \frac{a}{L} \right)^2 \frac{1}{(1 - na(1 + \frac{c_1}{\pi} \frac{a}{L}))^2} \tag{96}$$

$$\frac{E_2}{N} = \frac{\pi^2 \hbar^2}{6m} n^2 \left(3 + \frac{c_2}{\pi} \frac{a}{L} \right)^2 \frac{1}{(1 - na(3 + \frac{c_2}{\pi} \frac{a}{L}))^2} \ , \tag{97}$$

that is for $\ell = 1, 2$.

15.4 Comparison with the experiment

This section contains the main results of the Thesis. The Monte Carlo predictions for the energy are confronted with experimental results as summarized in two figures, shown in different units.

First figure, see Fig. 10, reports the ground-state energy per particle E/N as a function of a dimensionless parameter na, product of the linear density n and the s-wave scattering length. The horizontal axis has a linear scale, so that the perturbative regime of analytical expressions, $n|a| \ll 1$, is clearly visible.

The following observations apply to the shown data:

- We obtain the values of the bosonic ground-state energy are obtained numerically in Monte Carlo simulation. For ramification indices of $\ell = 1$ the energies are represented in the figure as red points and green points for ramification indices $\ell = 2$.

- The analytically obtained values using the approximation for the many-body energy in function of the ramification indices Eqs. (96,97) are represented by the blue line ($\ell = 1$) and the pink line ($\ell = 2$).

- The ground-state hard-rod energy obtained analytically Eq. (92) is depicted by the black line for the case of $\ell = 1$ and the red line for a ramification index of $\ell = 2$.

- The turquoise and yellow points are the attraction and repulsion values, respectively, extracted from the experimental data using g3data.

We find an excellent agreement between our numerical results and the experimental values for small s-wave scattering lengths, such that $n|a| \ll 1$ for the lowest value of the ramification index, $\ell = 1$. The Monte Carlo results for the ground-state energies for ramification index $\ell = 2$ are qualitatively similar to the experimental data but show quantitative differences. At the same time the experimental results do not follow the analytical limits for $a \to 0$.

The second figure, see Fig. 11, shows the dimensionless coupling constant in the horizontal axis, $|g| = 2/(n|a|)$, on a logarithmic scale. In this units, weak repulsion corresponds to $|g| \ll 1$ and strong repulsion to $|g| \gg 1$. The vertical axis portrays the quotient between the ground-state energy per particle E/N and ideal Fermi gas, $\frac{pi^2\hbar^2n^2}{6m}$. The experimental results are represented by red symbols while the values obtained through Monte Carlo are shown as blue symbols. In such units, it becomes evident that for the lowest ramification index there is an excellent agreement, and for $|g| \to \infty$ the energy is well reproduced by the ideal Fermi gas value, $E/N = 1(\pi^2\hbar^2/6m)$. For the upper branch, Monte Carlo results approach the analytic limiting value, $E/N = 9(\pi^2\hbar^2/6m)$, for $|g| \to \infty$ while the experimental data is slightly off.

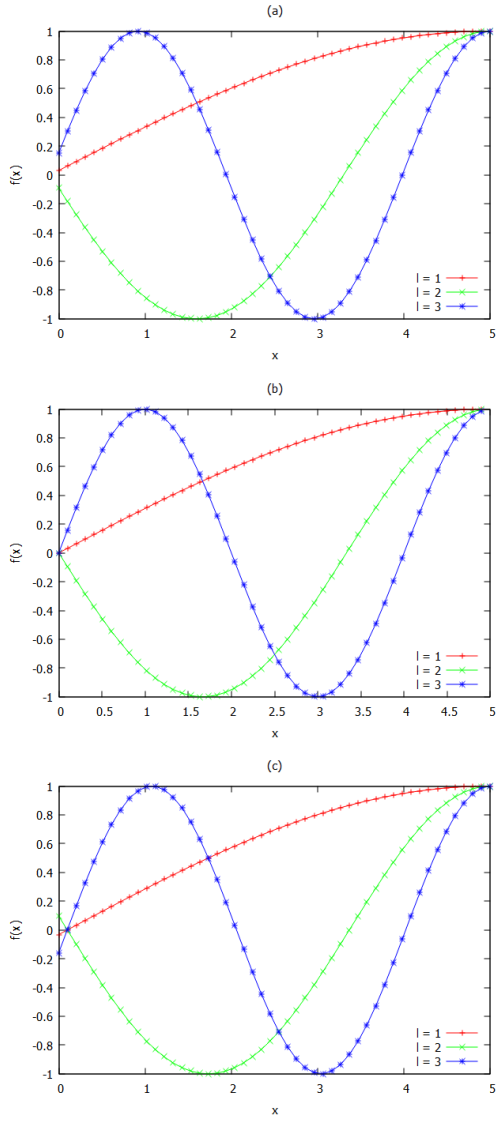


Figure 9: Typical examples of Jastrow function $f(x)$ for different ramification indices $\ell = 1; 2; 3$ for characteristic values of the s-wave scattering length. Panels: (a) $a < 0$, i.e. ground-state of repulsive Bose gas ($\ell = 0$) and its topologically excited states $\ell > 1$ (b) $a = 0$, impenetrable bosons, or ideal Fermi gas (c) $a > 0$, i.e. excited states of an attractive Bose gas. Using different ramification indices ℓ we can see how the number of nodes, hence the topological number n_t changes accordingly. The topological number n_t counts the number of nodes (when $f(x) = 0$) and its values depends on ℓ and the sign of a. For $a \geq 0$ the value of the ramification index is equal to the value of topological number, $\ell = n_t$.

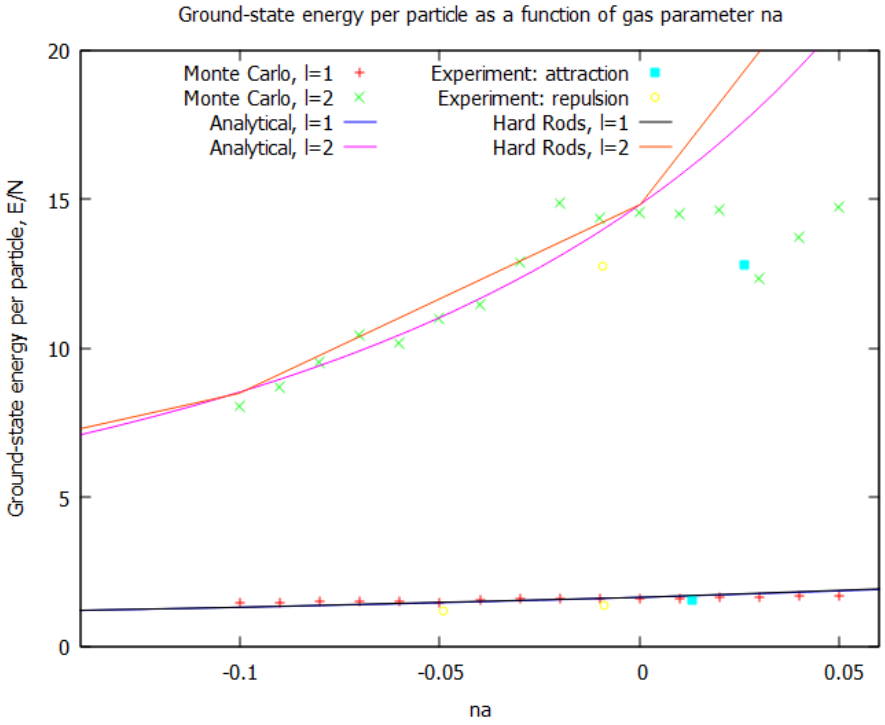


Figure 10: Ground-state energy per particle E/N as a function of gas parameter na. Red and green symbols: the values of the bosonic ground-state energy are numerically obtained in Monte Carlo simulation for ramification indices of $\ell = 1$ (red points) and $\ell = 2$ (green points). Turquoise and yellow points: experimental data obtained for attractive and repulsive interactions, respectively. Lines: analytical expressions (96,97) we derived are represented by the blue line ($\ell = 1$) and the pink line ($\ell = 2$). The ground-state hard-rod energy obtained analytically Eq. (92) is depicted by the black line for the case of $\ell = 1$ and the red line for a ramification index of $\ell = 2$.

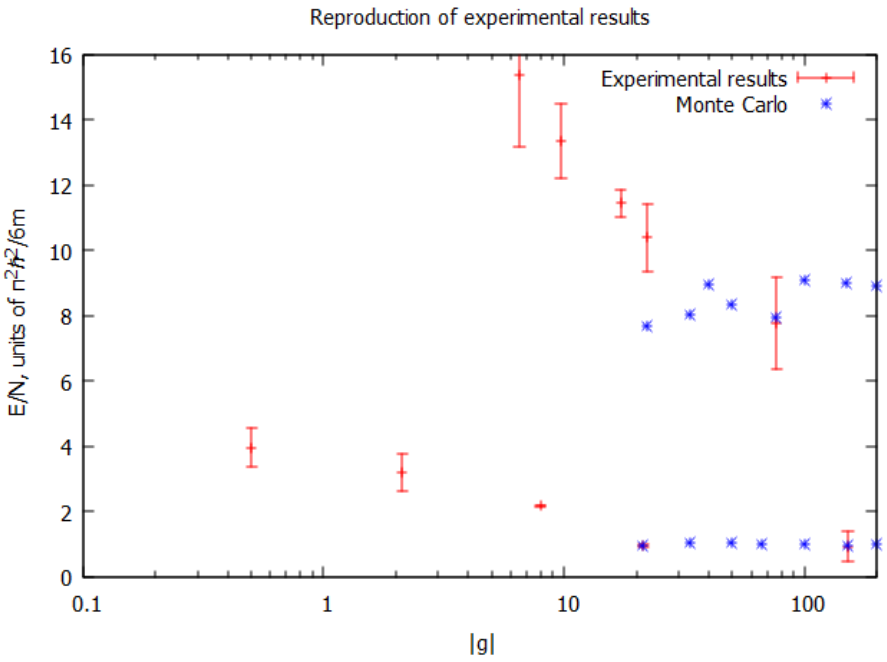


Figure 11: Energy plot presented in experimental units. The horizontal axis represents the dimensionless coupling constant g, while the vertical axis portrays the quotient between the ground-state energy per particle E/N and corresponding value in an ideal Fermi gas energy $\pi^2\hbar^2n^2/6m$. The experimental results are represented in red while the values obtained through our Monte Carlo implementation are given in blue.

Part IV
Conclusions

To conclude, we developed an *ab-initio* Monte Carlo code, tested it and used it to explain experimental results starting directly from a microscopic model. The structure of the project contains four parts (i) development of the code (ii) verifying and testing the code (iii) development of the model and analytical theory (iv) obtaining Monte Carlo data, analysing it and making a comparison with the experimental data of Ref. [4].

(i) **Code development.** We have accomplished the implementation "from scratch" of a Monte Carlo method capable of sampling the system properties and computing observables of interest. Listing the parts that lead to the achievement this success,

- we have defined a microscopic model consisting of particles interacting via contact potential (modelled by Dirac delta function) in a box with periodic boundary conditions
- we implemented a Metropolis algorithm to perform quantum Monte Carlo calculations and to quantify the correlations within a gas via calculation of the pair distribution function and the ground-state energy of the system.

(ii) **Code verification** After the code development, our Monte Carlo implementation has been subjected to multiple tests consisting on contrasting various observables found via simulation of an ideal Fermi gas system, that is a gas where the particles do not interact, and the analytically explicit results. In the testing, we calculated a pair distribution function and found an excellent agreement between our simulated values and the exact ones. For the ground-state energies, we have used two estimators to compute the kinetic energy of our simulations. The first estimator was constant as the wave function for the ideal Fermi gas is exact, while the second was fluctuation, but once averaged was found to be in agreement with the ground-state energy within a statistical error. Then, we compared the finite size effects of Monte Carlo simulation of ideal Fermi gas with the theoretical ones, reaching an outstanding agreement between the two.

(iii) **Microscopic model** Regarding the practical application of the code, we successfully managed to reproduce experimental results obtained in bosonic gases. This was accomplished by choosing a new Jastrow function, in function of the scattering momentum k and phase shift δ, solving it. In order to find k and δ we used a recursive dichotomic search algorithm implemented making use of the boundary conditions for short (Bethe-Peierls) and large (periodic boundary conditions) interparticle distances, which are in function of two know parameters: the s-wave scattering length a and

the ramification index ℓ, which is related to the topological number n_t that expresses the number of nodes in the Jastrow function. Afterwords, a linear fitting procedure was applied to describe the dependence of the scattering momentum on the s-wave scattering length a. This has allowed to construct an analytical prediction for the many-body energy for small values of a. In addition, we generalized the excluded-volume corrections in expression of the many-body energy to include the possibility of topological excitations.

(iv) **Comparison with experiment** We compare the predictions of the derived analytical expressions both with the Monte Carlo data obtained in simulations and with the experimental results. The results obtained for the lowest ramification index, $\ell = 1$, show an excellent agreement with the analytical and experimental results. For the ramification index $\ell = 2$ a fair agreement was obtained between Monte Carlo and experimental values. It is appropriate to mention that for that branch the experimental values do not satisfy the analytical limits. Possible explanation for such a discrepancy might be

   (i) on the theoretical side, limitations of the model we used.

   (ii) on the experimental side, the experiment not being entirely in one-dimensional regime.

As a future work, it might be interesting to predict the behavior of correlation functions (pair correlation function $g(r)$, momentum distribution, etc) in a gas with topological excitations.

16 Bibliography

References

[1] Beatrice Ellerhoff Oscar Garcia Montero, Alex Schuckert. What is many body physics?, 2020. Their webpage, *https://manybodyphysics.com/*.

[2] Shane Dooley. Robust quantum sensing in strongly interacting systems with many-body scars. *PRX Quantum*, 2:020330, May 2021.

[3] Marcus Woo. Quantum machine appears to defy universe's push for disorder. *Quanta Magazine*, 2019. Available at *https://www.quantamagazine.org/quantum-scarring-appears-to-defy-universes-push-for-disorder-20190320/*.

[4] Wil Kao, Kuan-Yu Li, Kuan-Yu Lin, Sarang Gopalakrishnan, and Benjamin L. Lev. Topological pumping of a 1d dipolar gas into strongly correlated prethermal states. *Science*, 371(6526):296–300, Jan 2021.

[5] Peter Bonate. A brief introduction to monte carlo simulation. *Clinical pharmacokinetics*, 40:15–22, 02 2001.

[6] Paul Dirac. The principles of quantum mechanics. *Oxford University Press*, 1930.

[7] David Hubber. How to do units and scaling right (or at least how to not do them wrong). 2015.

[8] Jing Yang and Adolfo del Campo. One-dimensional quantum systems with ground-state of jastrow form are integrable, 2022.

[9] Raymond Chang. Physical chemistry for the biosciences. *CA: University Science*, 2005.

[10] Li Yang Liming Guan and Han Pu. Strongly interacting quantum gases in one-dimensional traps. 2015.

[11] G. E. Astrakharchik. Quantum monte carlo study of ultracold gases (phd thesis), 2014.

[12] J. Boronat and J. Casulleras. Monte carlo analysis of an interatomic potential for he. *Phys. Rev. B*, 49:8920–8930, Apr 1994.

[13] M. Girardeau. Relationship between systems of impenetrable bosons and fermions in one dimension. *Journal of Mathematical Physics*, 1(6):516–523, November 1960.

[14] Peng Zhang, Long Zhang, and Youjin Deng. Modified bethe-peierls boundary condition for ultracold atoms with spin-orbit coupling. *Physical Review A*, 86(5), Nov 2012.

[15] Colin Kelley Thomas Williams. Gnuplot an interactive plotting program. Available at *http://www.gnuplot.info/docs_ 5.4/Gnuplot_ 5_ 4.pdf*.

Part V
Annex

Algorithm 1: $pbcDist(x)$

Input: **float** x
Output: **float** x'
if $x > L/2.$ **then**
  return $x - L$
if $x < -L/2.$ **then**
  return $x + L$
return x

Algorithm 2: $pbcMov(x)$

Input: **float** x
Output: **float** x'
if $x > L$ **then**
  return $x - L$
if $x < 0.$ **then**
  return $x + L$
return x

Algorithm 3: $difContributions(newPos, oldPos, \vec{R}, index)$

Input: **float** $newPos$, **float** $oldPos$, **vector<float>** $\vec{R}$, **int** $index$
Output: **float** Δu

```
1 float newContribution = 0. ;                          /* u_C(x') */
2 float oldContribution = 0. ;                          /* u_C(x) */
3 for i = 0; i < N ; i + 1 do
4     if index ≠ i then
5         newContribution+ = u(pbcDist(R[i] − newPos));
6         oldContribution+ = u(pbcDist(R[i] − oldPos));
7 return newContribution − oldContribution ;        /* Eq. ( 21) */
```

Algorithm 4: $metropolisAlgorithm(\vec{R})$

Input: **vector<float>** $\vec{R} = [x_0, ..., x..., x_{N-1}]$

```
1 for i = 0; i < N ; i + 1 do
2     float x'_i = pbcMov(R[i] + randomFloat(−Δt, Δt));
3     float Δu = difContributions(x'_i, R[i], R, i);
4     if ((int)((exp(2. ∗ Δu) + randomFloat(0, 1)) then
5         R[i] = x'_i;
```

Algorithm 5: $g2(\vec{h}, \vec{R})$

Input: **vector<float>** $\vec{h}$, **vector<float>** $\vec{R}$

```
1 for i = 0, i < N, i = i + 1 do
2     for j = i + 1, j < N, j = j + 1 do
3         float dist = fabs(pbcDist(R[j] − R[i]));
4         h[(int)(dist)/(nΔx)]++
```

Algorithm 6: $getDriftForce(\vec{R})$

Input: **vector<float>** $\vec{R}$
Output: **vector<float>** $driftForce$

```
1 vector<float> driftForce(N, 0.) ; /* Declare a vector with size
   N and set to 0.  */
2 for i = 0, i < N, i = i + 1 do
3     for j = 0, j < N, j = j + 1 do
4         float dist = pbcDist(R[j] − R[i]);
5         driftForce[i]+ = u'(dist);
6 return driftForce
```

Algorithm 7: $getKineticEnergy1(\vec{R}, driftForce)$

Input: **vector**<**float**> $\vec{R}$, **vector**<**float**> $driftForce$
Output: **float** $E_{kin}^{(1)}$

```
float auxE;
float F^2;
for i = 0, i < N, i = i + 1 do
    F^2 = driftForce[i] * driftForce[i];
    for j = i + 1, j < N, j = j + 1 do
        float dist = pbcDist(R[j] - R[i]);
        auxE+ = u''(dist);
return -(auxE + F^2 * 0.5)
```

Algorithm 8: $getKineticEnergy2(driftForce)$

Input: **vector**<**float**> $driftForce$
Output: **float** $E_{kin}^{(2)}$

```
float F^2;
for i = 0, i < N, i = i + 1 do
    F^2 = driftForce[i] * driftForce[i];
return F^2 * 0.5
```

Algorithm 9: $dS(min, max, auxAnt)$

Input: $min = \epsilon, max = \pi - \epsilon, auxAnt = \infty$
Output: δ

```
δ_0 = (min + max)/2;
aux ← PDD(δ_0);
if aux == auxAnt then
    return δ_0 ;                    /* If PDD(δ_0) == PDD(δ_-1) */
else if aux > 0 then
    return dS(δ_0, max, aux);
return dS(min, δ_0, aux);
```