Quantum Monte Carlo simulation for a many-body soliton in a waveguide

Arnau Cinca Roca

(Dated: June 21, 2021)

Thesis supervisor: Grigori Astrakharchik

GEP tutor: Marcos Eguiguren Huerta

Bachelor Degree in Informatics Engineering

Mentions in Computing Specialization

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

CONTENTS

I. Context 5
A. Introduction 5
B. Problem 6
C. Implied actors 7

II. Justification 8
A. Actual situation 8
B. Program selection 8

III. Scope & Obstacles 9
A. Objective 9
B. Sub objectives and indicators 9
1. Learn the theory 9
2. Program one-dimensional Monte Carlo simulator 9
3. Program the final simulator 9
C. Possible Obstacles and Risks 10

IV. Methodology & Rigour 11
A. Work Methodology 11
B. Rigour 11

V. Temporal Extension 12

VI. Project Phases 13

VII. Tasks Description 14
A. Previous Study 14
B. Design 14
C. Implementation 14
D. Experiments 14
E. Documentation 15
F. Meetings (M) 15

VIII. Tasks Dependency 16

IX. Time Duration 17

X. Task duration 18

XI. Gantt Diagram 18

XII. Risk Analysis 19

XIII. Budget 20

A. Costs identification 20

B. Staff costs 20

C. Energy costs 21

D. Others costs 22

E. Contingencies 23

F. Incidentals 23

G. Management control 23

XIV. Sustainability report 24

A. Economic dimension 24

B. Social dimension 24

C. Energetic dimension 24

XV. Monte Carlo Method 25

A. Introduction 25

B. Algorithm 25

1. Metropolis algorithm 26

XVI. Code 26

1. Coordinates 27

2. State 28

3. Monte Carlo 30

4. Measurements 31

XVII. Theory : one-dimensional systems 33

A. Model 33

B. Two-body problem: dimer in free space 33

C. Many-body problem: soliton mean-field solution 34
1. Gross-Pitaevskii equation 34
D. Many-body problem: soliton exact solution 35
1. Energy: McGuire solution 35
2. Density profile: Calogero solution 35
3. Pair distribution function: Castin solution 36

XVIII. Theory : three-dimensional systems 38
A. Model 38
B. Two-body problem: dimer in free space 38
C. Two-body problem: dimer in harmonic trap (Busch et al) 39
D. Two-body problem: dimer in a waveguide 39
E. Many-body problem: variational Monte Carlo solution 41

XIX. Results: one-dimensional system 43
A. Soliton: energy 43
B. Soliton: density profile 44
C. Soliton: pair distribution function 49

XX. Results: three-dimensional system 53
A. Soliton: energy 53
B. Soliton: density profile 53

XXI. Future work 56

XXII. Conclusions 56

XXIII. Bibliography 57

References 57

I. CONTEXT

A. Introduction

In this recent, years all the disciplines of science and technology have exponentially improved thanks, in part, to the improvement of computation power of our computers. This has allowed us to make such advances that were not even imagined in the past and has allowed us to simulate things that otherwise will costed a lot of resources and time.

The context of this work will be around physics, more precisely in quantum physics or mechanics, a relatively new field in physics that allowed us to really understand our world. Quantum mechanics are still in the research and there are a lot of unknowns we still do not resolve.

But quantum mechanics not only is useful in the theory, but many other technologies also exist thanks to quantum mechanics. Every one of us has at least one object with a processor (smartphone, laptop, etc), these processors can be built thanks to quantum mechanics. Other uses are in optical fiber, GPS, atomic clocks, MRI (magnetic resonance imaging), and many others. So it is very important to deepen our knowledge about quantum mechanics to get next-generation technologies.

B. Problem

In the quantum mechanics field, one of the main problems of the researchers is that the scale of the experiments is such small that are hard to carry out, normally they are expensive because they need specialized machines that are able to manipulate subatomic particles, set um a quantum field, and measure the states (in quantum mechanics a measurement can change the future result).

Another problem that make difficult the research in quantum mechanics is that quantum mechanics are not deterministic, setting up the same experiment can lead to different outcomes. This is because quantum mechanics work as a probabilistic system, and this forces the researchers to make the same experiment multiple times to estimate this probability, making the research slow and expensive.

During these years of studying quantum mechanics, McGuire derived an exact solution to the quantum problem[1]. The ground-state many-body wave function can be written explicitly as:

$$\psi(x_1, \cdots, x_N) = \prod_{i<j} \exp\left(-\frac{|x_i - x_j|}{a}\right) \tag{1}$$

where $a > 0$ is the one-dimensional s-wave scattering length corresponding to attraction between atoms.

But the real world is not one-dimensional, we only can produce a quasi-one-dimensional space, so this equation does not work properly, we need to find an exact equation that describes the real world in quasi-one-dimensional space.

C. Implied actors

This project is aimed to help quantum mechanics researchers to help them find a new and improved equation to describe quantum behaviors, and hopefully find the exact equation.

One other group that will be indirectly beneficiate are us (as costumers) and enterprises, because if the result of this final thesis leads to a good outcome, with the improvement of the knowledge in quantum mechanics, it may be possible to develop new technologies that will improve our lives.

II. JUSTIFICATION

A. Actual situation

In the actual situation of quantum mechanics, there is only known an exact formula for one-dimensional space, but our reality is built in alt leas three, therefore the exact equation for one-dimensional space does not work in the experiments.

In actuality there is an approximate formula for both one-dimensional and quasi-one-dimensional spaces, if given a Monte Carlo simulator and the approximate equation for one-dimensional space, the outcome of the simulator is the same as the exact equation, doing the same with the quasi-one-dimensional space must give the same result as the exact equation for quasi-one dimension space, which is unknown.

B. Program selection

After evaluating all possibilities of the programming language to use, we decide to use C (and standard libraries) as the main programming language because it is one of the fastest programming languages, it is stable and well tested during the years. C language also provides the capacity to adapt the code to C++ without any modification of the code in case of need of more modern structures or libraries.

III. SCOPE & OBSTACLES

A. Objective

The main objective of this project is to program a Monte Carlo simulator to simulate a given number of particles and other quantum physical parameters, to see its behaviors to find the exact equation of the quantum particles in a quasi-one-dimensional space.

B. Sub objectives and indicators

1. Learn the theory

The first objective is to learn, at least, the basic theory around quantum mechanics, to in the near-future apply it in the Monte Carlo simulator, and get a good result of it.

2. Program one-dimensional Monte Carlo simulator

To learn and structure the code it is easy to start with a known exact equation, such a McGuire, for one-dimensional movement.

This will give us a well build structure and a tested algorithm to adapt for a quasi-one-dimensional algorithm.

3. Program the final simulator

The final job to do is the actual project, to program the final version of a "Quantum Monte Carlo simulation for a many-body soliton in a waveguide".

C. Possible Obstacles and Risks

The possible obstacle to reach a good end of the project will be the capability to fully understand quantum mechanics in a short time.

Another possible obstacles it that our final program gives us a correct answer to the problem, and in the future help to find the exact solution of a quasi-one-dimensional problem.

Risks of losing the code or other "hardware" issues are practically nonexistent because during the course of this project the code will be secured in GitHub for any major change done.

IV. METHODOLOGY & RIGOUR

A. Work Methodology

This project is big and complex, to finish it, the author must learn about quantum mechanics and simulation, above this difficulty, to successfully finish the project the simulator must be capable to simulate a quasi-one-dimensional space for quantum particles and do some experiments to prove its correctness.

Scrum is a simple work methodology focused on big and complex projects and takes into account the unknowns of the project at the beginning, because of that, we consider this method as the base to decide our temporal extensions.

B. Rigour

To verify that the development follows the temporal estimations, there is a weekly meeting of at least one hour, in these meetings it will be presented the weekly job, and do some corrections of the code or explain the basic knowledge of quantum mechanics needed to finish the project.

V. TEMPORAL EXTENSION

The total temporal extension of this project is about 450 hours (18 ECTS and 25 hours/ECTS), the starting date is 15 of February of 2021, and the nearly end date is 28 of June of 2012, so we have more or less seventeen weeks to finish this project, obviously, we need some margin time in case of some incidence.

Assuming some margins, and maybe some rushes, the estimated daily time is around 6-7 hours, with this estimation we can calculate the finish date, is about fourteen weeks, more exactly the day 10 of May of 2021.

Given the date, we can more easily put some deadlines, of the phases described in the next section.

VI. PROJECT PHASES

This project is divided into six phases, four of them related to the project itself, and the other two are documentation and management of the project.

- **Previous Study:** In computer science courses do not teach quantum mechanics, so this phase is for learning all the theory needed to finish the project correctly.
- **Design:** As the code needed to complete the project is big and the concepts of the work are not easy, design the code and give a basic infrastructure is practically obligatory to finish this project correctly.
- **Implementation:** Like most computer science projects, this project needs a program to complete, in this case, a simulator, this phase is for finishing the result given by the previous phase.
- **Experiments:** To finish this project, some tests to verify the correctness of the code and experiments to get the results of the project are needed.
- **Documentation:** As with all the projects, this project needs documentation to share the work done and results, this documentation is for evaluating the project too.
- **Meetings:** During the course of this project, there are some obligatory meetings, this "phase" is not actually a phase, but it encompasses all the meetings.

VII. TASKS DESCRIPTION

This section describes all the tasks needed to complete the project divided by its phase, some tasks have dependencies, we will discuss these dependencies in the next section.

A. Previous Study

- **Quantum Theory (TQ):** As the name suggests, this task is for learning the quantum mechanics theory.
- **Simulator Theory (TS):** This task is to learning about the simulator used in this project, the Monte Carlo method, and the Manhattan algorithm.

B. Design

- **Design simple simulator (DS):** Once received some of the theory needed, the next step is to start the design and give an initial scheme of the code.
- **Design the metrics needed (DM):** To verify the code and get the results, some metrics are needed, this task is for deciding all the future metrics needed.

C. Implementation

- **Implement the one-dimensional simulator (IOS):** This task is for implementing the design of the task DS in one-dimensional space.
- **Implement the metrics needed (IM):** This task is to implement the metrics decided in the task DM.
- **Implement the final simulator (IFS):** The final objective of this project is to program a simulator of a quasi-one-dimensional space, this task is to adapting the code to work in the new space.

D. Experiments

- **Experiment with the simulator (ES):** This task is for getting the results of the simulator and verifying the correctness of it.

E. Documentation

This phase is much bigger than the others, to avoid big tasks, the phase is divided into more tasks, more or less one per phase.

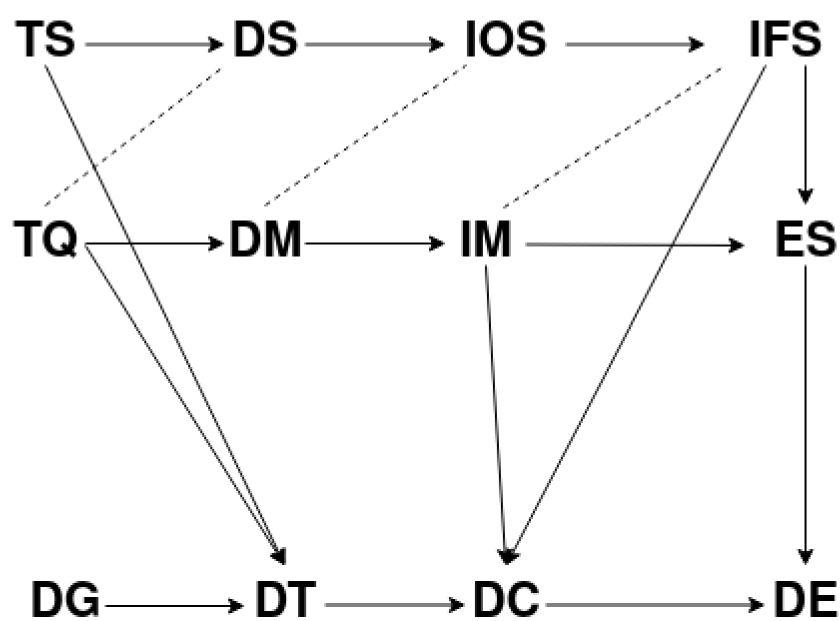
- **GEP (DG):** This task encompasses all the documentation done by the GEP.
- **Theory documentation (DT):** Once done all the theory Previous Study tasks, the final document must contain the information needed to finish the project.
- **Code documentation (DC):** The documentation directly or indirectly must explain the code implemented in this project, or at least its scheme.
- **Experiments documentation (DE):** The final task is to describe the final results of the experiments done in the Experiments phase.

F. Meetings (M)

During the course of this project, there are weekly meetings to keep track of the work and teach/guide about the theory and work done. These meetings have a duration of 1-2 hours.

VIII. TASKS DEPENDENCY

The dependency of the tasks are showed with arrows, and co-dependencies are marked with dotted lines, finally, left to right are shown an approximation of the time distribution of the project, being the left tasks at the beginning of the project and right the lasts tasks of the project. The meetings are not showed in this diagram because their periodicity and in part are included inside the other tasks.



IX. TIME DURATION

The time duration can be distributed in four parts, GEP, theory & design, code, experiment, just as the documentation tasks are defined. The first two parts give more hours per day, as we can calculate with the next section, because if there is some problem it is better to find it as soon as possible, and have time to fix it later.

- **GEP:** This part has a length of four weeks and includes the tasks DG, TS, and TQ.
- **Theory & Design:** This part has a length of five weeks and includes the tasks DS, DM, DT, IOS, and IM.
- **Code:** This part has a duration of four weeks and includes the tasks of IFS, DC, and ES.
- **Experiments:** This last part has a duration of one week, and includes the task DE, and finish all the documentation.

X. TASK DURATION

Finally, in this section, there are described the exact time duration of each task.

Task Id	Task Name	Duration
TQ	Quantum Theory	35
TS	Simulator Theory	25
DS	Design simple simulator	30
DM	Design the metrics needed	30
IOS	Implement one-dimensional simulator	50
IM	Implement the metrics needed	40
IFS	Implement the final simulator	40
ES	Experiment with the simulator	25
DG	GEP	75
DT	Theory documentation	25
DC	Code documentation	25
DE	Experiments documentation	25
M	Meetings	25
	Total:	450

TABLE I. Task duration table

XI. GANTT DIAGRAM

The following Gantt diagram shows the distribution of tasks over the weeks, at the last row it shows the total time dedicated at the week, at the firsts six weeks the workload is about 40 hours, after these weeks the pork load is around 20 to 30 hours, that is because at the beginning of the semester there is less workload in the other courses of the university, and in case of some incident, have enough time to solve it adding hours at the end.

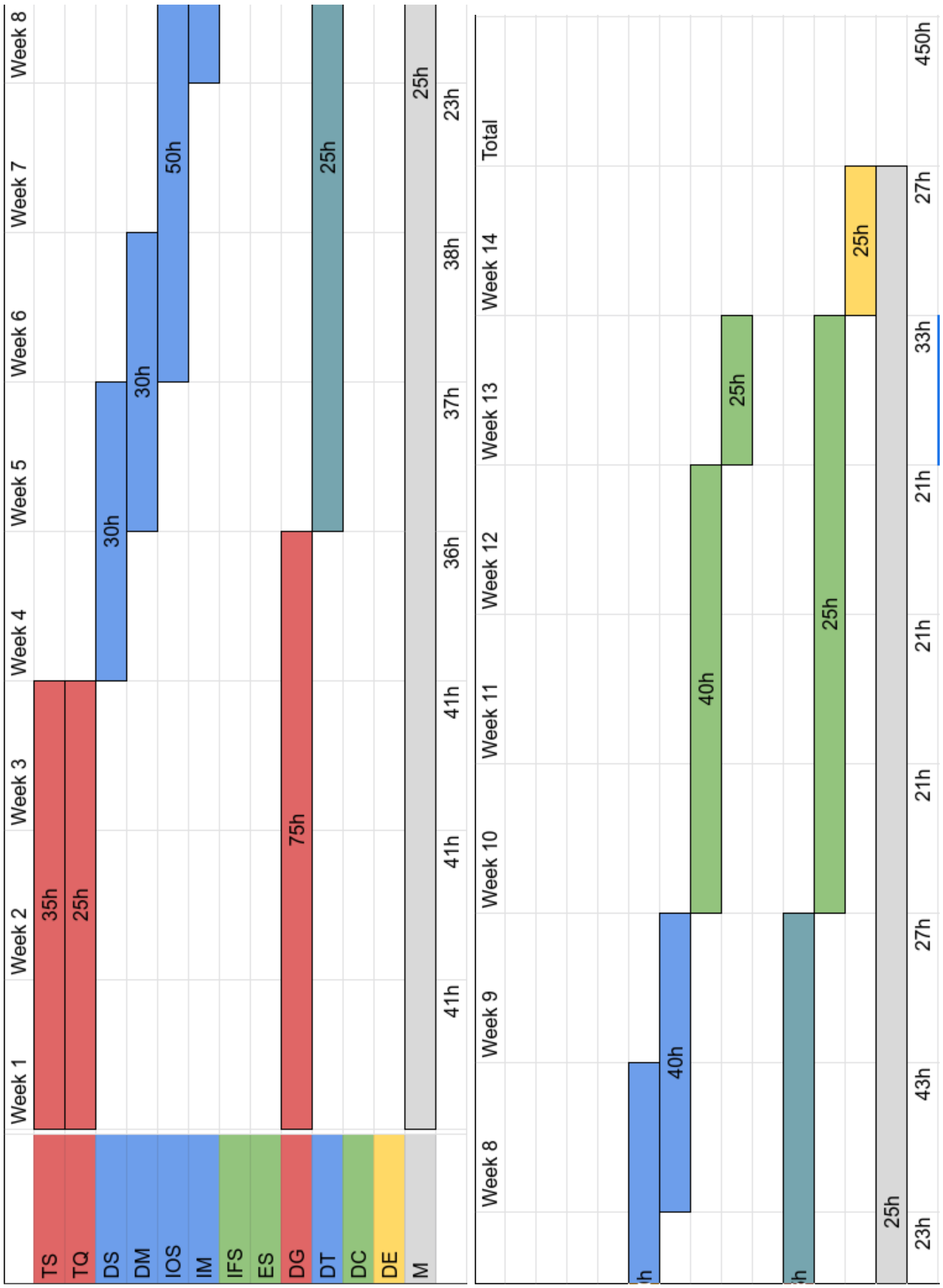


FIG. 1. Gantt Diagram

XII. RISK ANALYSIS

As mentioned in the "Possible Obstacles and Risks" there are practically no risks at all, one of the major risks is losing the code, but this is practically impossible because the code is upload to

Github.

XIII. BUDGET

A. Costs identification

The costs of the project can be divided into three parts:

- **Staff:** programmer, project director, teacher.
- **Energy:** electric consumption of the devices used during the development and tests.
- **Other:** personal computers to develop the project, books and auxiliary material to help the quantum mechanics learning, internet connection, space.

B. Staff costs

As seen in the previous section, this project requires at least one programmer, a project director, and a quantum mechanics teacher, quantum mechanics and project director in this project are the same and in this case only are one programmer, the role of a programmer, encompasses tasks like programming, testing, experimenting and writing documentation, and learning time.

According to my previous experience in *JediUPC*[4], as a programmer, the average price per hour in the projects are around 20€/h, as a project director, around 30€/h, and as a teacher, around 25€/h.

Role	Hours	€/h	Total(€)
Programmer	425	20	8500
Teacher	10	25	250
Project director	15	30	450
		Total:	9200

TABLE II. Costs divided by roles

Task	Role	Hours	€/h	Total(€)
TQ	Programmer	35	20	700
TS	Programmer	25	20	500
DS	Programmer	30	20	600
DM	Programmer	30	20	600
IOS	Programmer	50	20	1000
IM	Programmer	40	20	800
IFS	Programmer	40	20	800
ES	Programmer	25	20	500
DG	Programmer	75	20	1500
DT	Programmer	25	20	500
DC	Programmer	25	20	500
DE	Programmer	25	20	500
M(1)	Teacher	10	25	250
M(2)	Preject director	15	30	450
			Total:	9200

TABLE III. Costs divided by tasks

As shown in Table 2, The tasks TQ and TS are integrated with the meetings and self-learning and only apply to the programmer, the meetings are applied only to the project director and teacher.

C. Energy costs

According to *Endesa*[5], the electricity price is around 0.11 €/KWh, the power consumption of a laptop is around 200W, and the hours of work are 450 hours and at least 25 hours, in total around 10.45€ of electricity in the whole project.

$$0.11\frac{euros}{KWh}(0.2KW * (450 + 25)h) = 10.45euros \tag{2}$$

D. Others costs

There are not many materials needed to develop this project, only a laptop or PC to develop the project, the price of a ”correct” laptop to work costs at least 1000€.

To do the meetings and research there is needed a good internet connection, according to with the offers of the internet providers is around 50€ per month, by five month is around 250€.

To help the learning of the theory needed to develop the project, there is assigned a 100€ budget.

Finally, if there is a need for some space of work, a room in Barcelona is around 350€ per month, by five month, 1750€.

Name	Price(Monthly)	Total(€) five months
Laptop	200	1000
Internet	50	250
Auxiliary learning material	20	100
Working space	350	1750
	Total:	2100

TABLE IV. Costs divided by roles

E. Contingencies

As with all the projects, some unforeseen may occur during the development of the project, to avoid losings, to each cost there is added a percentage of the original costs, in this project due to the uncertainly of the final results, there is a 20% of contingency.

Cost	Price(Monthly)	contingency(€)
Staff	9200	1840
Energy	10.45	1,09
Others	2100	420
Total:	11310.45	2261.09

TABLE V. Costs divided by roles

F. Incidentals

As mentioned in the previous sections there are not many risks, the worst risk is losing the code, to avoid, all the code of the project is upload to GitHub, with this and local copies, the code practically impossible to lose, other risks that can occur are broken hardware (PCs), in this case, if there is not a replacement, the only solution is to buy another hardware.

G. Management control

The Management control is done in the weekly meetings, so it does not add any costs to the project, there is plenty of time to do it in the meetings, so it is not necessary to add extra costs in a parallel system, due to the meetings are each week is even better than a "normal" method of management control.

XIV. SUSTAINABILITY REPORT

A. Economic dimension

This project has a huge impact on the economic dimension, as it allows the scientific community to get a result that otherwise will cost even millions of euros to get, as to get the same results as this project, there is needed machinery to manipulate particles, and read its states, this in quantum mechanics is very hard and expensive.

If the results of this project are positive, it will help to find an exact formula for solitons in a wave function, this will allow the future to develop new technologies.

B. Social dimension

This project has not a huge impact on the social dimension, directly at most, as mentioned in the previous section, will allow the scientific community to do their job easier and more comfortably.

Also as mentioned in the previous section, if the project gives a positive outcome, it will help to develop new technologies, these technologies may change the life of millions of persons, giving new employment, and new products.

C. Energetic dimension

Finally, in the economic dimension, this project will allow avoiding experiments, running the code of this project probably is less expensive (in energy) than the current experiment, so all the experiments that can be avoided if using the simulator of the project gives a positive energetic outcome.

Also, the new technologies that may be developed if the results of this project are correct, may let us use less energy.

XV. MONTE CARLO METHOD

A. Introduction

Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. The underlying concept is to use randomness to solve problems that might be deterministic in principle. They are often used in physical and mathematical problems and are most useful when it is difficult or impossible to use other approaches. Monte Carlo methods are mainly used in three problem classes: optimization, numerical integration, and generating draws from a probability distribution.

Monte Carlo methods can be used to solve any problem having a probabilistic interpretation. By the law of large numbers, integrals described by the expected value of some random variable can be approximated by taking the empirical mean of independent samples of the variable. When the probability distribution of the variable is parametrized, mathematicians often use a Markov chain Monte Carlo (MCMC) sampler. The central idea is to design a judicious Markov chain model with a prescribed stationary probability distribution. That is, in the limit, the samples being generated by the MCMC method will be samples from the desired (target) distribution.

B. Algorithm

In this project, we will use this method to solve numerically the behaviour of particles in one-dimensional space and in a three-dimensional space with a harmonic trap, to do so we need some algorithm that can generate states from previous states, in our case it is completely random. As it is random the resulting state not always may be valid, to make sure that the state is valid, given a probability function (described in the quantum theory section), and with the **Metropolis algorithm** the program can decide if the state is valid and continue from that state or if it is invalid revert it.

Apart from

1. Metropolis algorithm

The **Metropolis algorithm** is an MCMC algorithm that we will use to try to solve the problem. A brief "pseudo-code" of it it will be like:

- Propose a movement $\vec{R}_i \to \vec{R}'$
- If $p(\vec{R}') \ > \ p(\vec{R}_i)$ then (accept movement)
  - $\vec{R}_{i+1} = \vec{R}'$
- else (accept with probability)
  - i.e. calculate $int\left(\frac{p(\vec{R}')}{p(\vec{R}_i)} + \xi\right)$
    $\xi \in (0;1)$ random number
    $(int == 0) \to$ reject $(\vec{R}_{i+1} = \vec{R}_i)$
    $(int == 1) \to$ accept $(\vec{R}_{i+1} = \vec{R}')$

XVI. CODE

All the code used in this project can be found here.
The program can be divided into two parts, the first one the Monte Carlo method itself, described above, and the second one, the measurements that are taken from it.

For the first part of the program, we need an algorithm that is easily modifiable from one-dimension to three-dimensions, to do so, the best way to do is to separate the coordinates system for the particles and state from the Metropolis algorithm.
As the programming language to do the code is C, but C is not mean to be an object-oriented language, but with structs and functions, can behave like one. In the following sections, we will describe these structs and their "methods".

1. Coordinates

The most basic struct are the coordinates that define the position of the particles.

```
typedef struct coords{
    double x;
#if TRIDIM == 1
    double y;
    double z;
#endif
} coords;
// returns a coordinates that are right at the origin {0}
// ({0,0,0} in three-dimensions).
coords orig();

// returns the euclidean distance between two particles (i and j).
double dist(coords i, coords j);

// initiate the coordinates c
// with a random value between -initial_dispersion and initial_dispersion.
void coordsRandomInit(coords* c, double initial_dispersion);

// moves the particle c increasing (or decreasing)
// its values with a random number.
void randomMove(coords* c);
```

As we can see, there is a define ($TRIDIM$) to indicate in the compiling time if we want to work with a one-dimensional space or a three-dimensional space. This define is also used in other parts of the code as we will see in the next sections.

2. State

The state struct is the state of all the particles at one instant, every iteration the state can change and became a new state with the Metropolis algorithm.

```
typedef struct state{
        unsigned int rejected;
        unsigned int tried;
        coords* particle_coords;
} state;

// returns the ratio of acceptance of the Metropolis algorithm
// a good ratio must be between 0.1 and 0.9.
double acceptanceRatio(state* s);

// initializes all the particles in the state s
// using the randomInit method from the coordinates struct
// with initial_dispersion.
void initState(state* s, double initial_dispersion);

// applies the Manhattan algorithm over the state
// to generate the next state, and it overrides the state s with it.
void nextState(state* s);

// returns the center (mean) of all the positions from the particles.
// 1D: x; 3D: z
double centerOfMases(state* s);

// returns the energy of the state.
double getEnergy(state *s);
```

The first two fields in the struct are only to determine how often the Metropolis algorithm accepts the state change, the third field is the coordinates (position) of all the particles in the experiment.

To determine the acceptance of the new state, the division of the probabilities must give more than one, as the result of the probabilities is exponential, if the inside of this exponential ($\triangle u$) is smaller than 0 it means a division of the probabilities is less than one. $\triangle u$ is given by:

$$\triangle u = u_1(\vec{r_i'}) - u_1(\vec{r_i}) + \sum_{i \neq j} u2(|\vec{r_i'} - \vec{r_j}|) - u2(|\vec{r_i} - \vec{r_j}|) \tag{3}$$

where u_1 is 0 in one-dimension and the harmonic trap of the Eq. 24 for the wave-guide, and u_2 is $-r/a$ for one-dimensional space and Eq. 37 for the wave-guide.

The energy is calculated by the following equation:

$$E = -\frac{\hbar^2}{2m}\left[\sum_{i=1}^{N}\sum_{\alpha=\{x,y,z\}} \vec{F}_{i\alpha}^2 + \sum_{i=1}^{N}\left(\frac{\partial^2}{\partial x_i}, \frac{\partial^2}{\partial y_i}, \frac{\partial^2}{\partial z_i}\right) u_1(\vec{r_i}) + 2\sum_{i<j}\left\{u_2''(r_{ij}) + \frac{D-1}{r_{ij}} u_2'(r_{ij})\right\}\right], \tag{4}$$

where the drift force is:

$$\vec{F_i} = \left\{\frac{\partial u_1}{\partial x_i}, \frac{\partial u_1}{\partial y_i}, \frac{\partial u_1}{\partial z_i}\right\} + \sum_{i \neq j} u_2'(r_{ij}) \frac{\vec{r_i} - \vec{r_j}}{r_{ij}} \tag{5}$$

The montecarlo struct is a simple struct to separate the Monte Carlo method from the measurements and the main code.

```
typedef struct montecarlo{
        state* state;
} montecarlo;

// initializes the montecarlo struct and then initializes the first state.
montecarlo* montecarloInit(double initial_dispersion);

// executes the Metropolis algorithm one iteration.
void runOneStep(montecarlo* mc);

// executes the Metropolis algorithm steps iterations.
void runNSteps(montecarlo* mc, unsigned int steps);
```

The state field from the montecarlo struct is the actual state of the experiment.

4. Measurements

The last struct, measurements, contains all the information needed for the experiment to plot the results.

```
typedef struct measurements {
        unsigned int size;
        double delta_x;
        double range; //[-range, range]
        unsigned int iterations;
        unsigned int* histoX;
#if TRIDIM == 1
        unsigned int* histoY;
        unsigned int* histoZ;
#else
        unsigned int** heatmap;
#endif
} measurements;

// initializes and calculates all the fields in the struct.
measurements* measurementsInit(double range, unsigned int size);

// add to the graphics the state s, and increments by one the iterations field.
void addIteration(measurements* m, state *s);

// print at the file fp the desity profile of the x axis.
void printDensityProfileX(measurements* h, FILE *fp);
#if TRIDIM == 1

// print at the file fp the desity profile of the y axis.
void printDensityProfileY(measurements* h, FILE *fp);

// print at the file fp the desity profile of the z axis.
void printDensityProfileZ(measurements* h, FILE *fp);
```

```
#else

// print at the file fp the heat-map of the pair distribution function en the x ax
void printDensityProfile2D(measurements* h, FILE *fp);

// print at the file fp the [i,-i] diagonal of the heat-map.
void printDensityProfile2DDiag1(measurements* h, FILE *fp);

// print at the file fp the [i,i] diagonal of the heat-map.
void printDensityProfile2DDiag2(measurements* h, FILE *fp);
#endif
```

The first three fields of the struct define the shape of the graphics, all the graphics have *size* points between $-range$ and $range$, each point of the graphic represents a $delta_x$ space range.
The fourth field counts all the times that a state is added to the graphics, is used to normalize when printing the result, histoX is the histogram of the density profile in the x axis, histoY and histoZ are similar, but corresponding to the axis y and z respectively, this two are only used if we work in a three-dimensional space, finally, we have the heat-map, only when it is in one-dimensional space, that represents the pair distribution function.

XVII. THEORY : ONE-DIMENSIONAL SYSTEMS

A. Model

The Schrödinger equation can be rewritten with a Hamiltonian equation:

$$\hat{H} = \sum_{i=1}^{N} -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \sum_{i<j} g\delta(x_i - x_j) \tag{6}$$

Resulting in this Schrödinger equation:

$$i\hbar\frac{\partial \psi}{\partial t} = \left(\sum_{i=1}^{N} -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \sum_{i<j} g\delta(x_i - x_j)\right)\psi \tag{7}$$

B. Two-body problem: dimer in free space

For two particles with coordinates x_1 and x_2, the Schrödinger equation can be explicitly solved in the center of mass frame in terms of the relative coordinate, $x = x_1 - x_2$. The resulting wave function $\psi(x)$ is

$$\psi(x_1 - x_2) = \exp\left(-\frac{|x_1 - x_2|}{a}\right), \tag{8}$$

where $a > 0$ is the s-wave scattering length.

The density profile is than given by the square of the absolute value of the wave function (8), $n(x) = |\psi(x)|^2$ resulting in

$$n(x_1 - x_2) = \frac{2}{a}\exp\left(-\frac{2|x_1 - x_2|}{a}\right), \tag{9}$$

The density profile should be measured as the distance calculated from center of mass position $x = x_1 - R$ with $R = (x_1 + x_2)/2$. One notes that $x_1 = R + x$, $x_2 = R - x$, so that $x_1 - x_2 = 2x$ Thus, the density profile of a dimer is

$$n(x = x_1 - R) = \frac{2}{a}\exp\left(-\frac{4|x|}{a}\right), \tag{10}$$

C. Many-body problem: soliton mean-field solution

1. *Gross-Pitaevskii equation*

$$i\hbar\frac{\partial}{\partial t}\psi(x,t) = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\psi(x,t) + g|\psi(x,t)|^2\psi(x,t) \tag{11}$$

In the case of large number of particles N, the density profile of the soliton can be obtained in the mean-field theory explicitly and is given by

$$n_{MF}(x) = \frac{N-1}{2a}\frac{1}{\cosh^2\left[\frac{(N-1)x}{a}\right]} \tag{12}$$

It can be noted, that the number of particles N and the s-wave scattering length enters in the following combination

$$P_{MF} = \frac{N-1}{a} \tag{13}$$

That is, different systems having same value of P_{MF} will have similar density profiles.

The chemical potential corresponding to solution (12) is quadratic with respect to the number of particles N

$$\mu(N) = -N^2\frac{\hbar^2}{2ma^2} \tag{14}$$

and the energy is

$$E = \int_0^N \mu(N')dN' = -N^3\frac{\hbar^2}{6ma^2} \tag{15}$$

D. Many-body problem: soliton exact solution

1. Energy: McGuire solution

The ground-state energy is the sum of the kinetic and potential energies in the system at zero temperature. The ground-state energy of the many-body problem described by model (**??**) was calculated McGuire and is given by

$$E_{1D} = -\frac{mg^2}{\hbar^2}\frac{N(N^2-1)}{24} \tag{16}$$

with

$$g = -2\frac{\hbar^2}{ma} \tag{17}$$

the resultant equation is:

$$E_{1D} = -\frac{4\hbar^2}{ma^2}\frac{N(N^2-1)}{24} \tag{18}$$

after simplification results in:

$$E_{1D} = -\frac{\hbar^2 N(N^2-1)}{6ma^2} \tag{19}$$

2. Density profile: Calogero solution

The probability of finding a particle at a certain distance from the center of mass position is quantified by the density profile. For model (**??**) it was calculated by Calogero and Degasperis in Ref. [2]

$$n(x) = \frac{N!^2}{N\xi}\sum_{k=0}^{N-2}\frac{(-1)^k(k+1)}{(N-2-k)!(N+k)!}e^{-(k+1)|x|/\xi}, \tag{20}$$

in terms of the width of the Gross-Pitaevskii solution

$$\xi = \frac{\hbar^2}{m|g|N} = \frac{a_s}{2N} \tag{21}$$

It is normalized in such a way that the integral over x over the whole space is equal to N.

3. Pair distribution function: Castin solution

The pair distribution function $\rho(x, y)$ quantifies the probability of simultaneously finding one particle at position x and another particle at position y. Its value for the model described by Hamiltonian (**??**) has been explicitly calculated by Yvan Castin in the limit of a large number of particles N in Ref. [3]. The main contribution to it comes from the uncorrelated product of the density profiles, $\rho(x, y) \approx \rho(x)\rho(y)$. The most interesting effect is that of the non-trivial correlations,

$$\delta\rho(x, y) \equiv \rho(x, y) - \rho(x)\rho(y) \tag{22}$$

which are absent in the mean-field description.

The result obtained by Yvan Castin reads as

$$\delta\rho(x, y|0) \simeq -\frac{N}{16\xi^2}\frac{(2 + |X - Y|)\sinh(X/2)\sinh(Y/2) + 2\sinh(|X - Y|/2)}{[\cosh(X/2)\cosh(Y/2)]^3} \tag{23}$$

where the distances are calculated in the units of the soliton size, $X = x/\xi$ and $Y = y/\xi$. A contour plot of the pair distribution function is like the Figure 2.

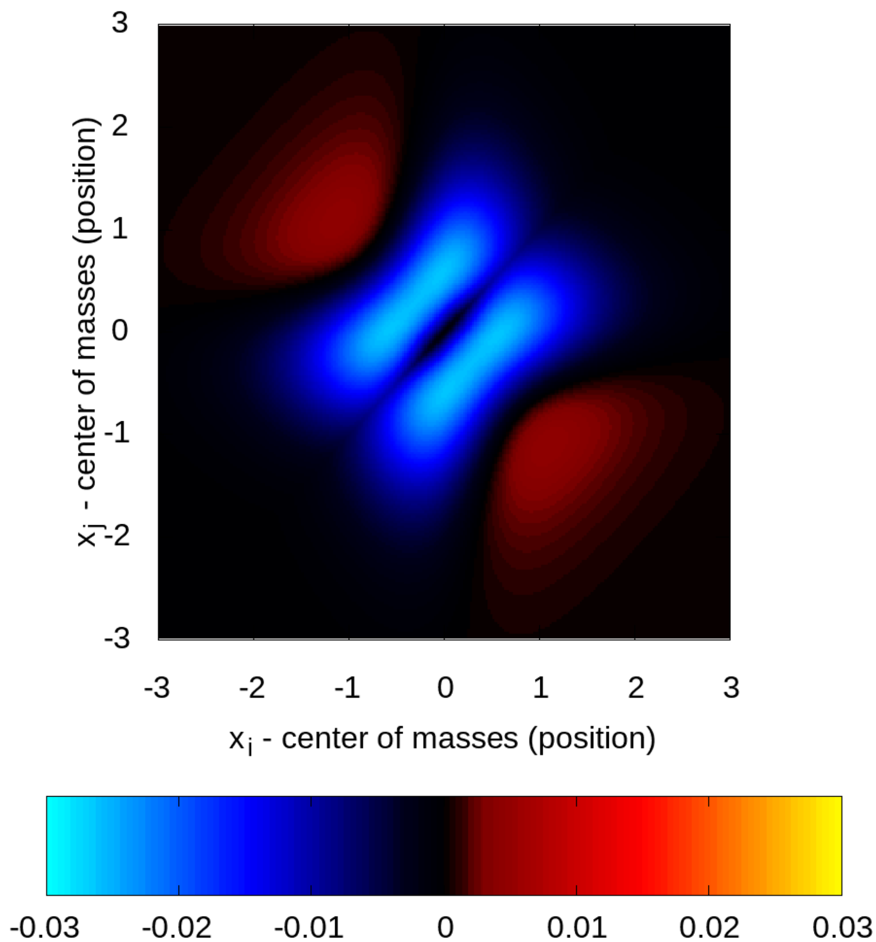


FIG. 2. Pair distribution $\delta(x, y)$ function as given by Castin's approximation, Eq. 23, shown with a heatmap. The parameters are chosen to be $N = 10$ particles and a s-wave scattering length of $a = 9$, which corresponds to the mean-field parameter (13) $(N-1)/a = 1$ and relatively large number of particles where Eq. 23 is expected to be valid.

XVIII. THEORY : THREE-DIMENSIONAL SYSTEMS

A. Model

A waveguide is a structure that guides waves, with minimal loss of energy by restricting the transmission of energy to one direction. Without the physical constraint of a waveguide, wave amplitudes decrease according to the inverse square law as they expand into three-dimensional space.

In quasi-one-dimensional space we have an external potential in a form of a harmonic oscillator:

$$V(x, y) = \frac{1}{2} m w^2 (x^2 + y^2) \tag{24}$$

The Schrödinger equation equation can be rewrited with a Hamiltonian equation with an harmonic oscilator in the x and y axis:

$$\hat{H} = \sum_{i=1}^{N} -\frac{\hbar^2}{2m}\left[\frac{\partial^2}{\partial x_i^2}, \frac{\partial^2}{\partial y_i^2}, \frac{\partial^2}{\partial z_i^2}\right] + \sum_{i=1}^{N} V(x_i, y_i) + \sum_{i<j} g\delta(x_i - x_j) \tag{25}$$

B. Two-body problem: dimer in free space

For two particles with coordinates x_1, y_1, z_1 and x_2, y_2, z_2 in a free space, we use the following wave function:

$$\psi(x_1, y_1, z_1, x_2, y_2, z_2) = \frac{\exp\left(-\frac{|\mathbf{r}_1 - \mathbf{r}_2|}{a}\right)}{|\mathbf{r}_1 - \mathbf{r}_2|}, \tag{26}$$

for the bound state where $a > 0$ is the s-wave scattering length. This state has a negative binding energy

$$E = -\frac{\hbar^2}{ma^2} \tag{27}$$

The wave function of a free-scattering state ($E = 0$) is given by

$$\psi(x_1, y_1, z_1, x_2, y_2, z_2) = 1 - \frac{a}{|\mathbf{r}_1 - \mathbf{r}_2|} \tag{28}$$

corresponding to the negative s-wave scattering length, $a < 0$.

C. Two-body problem: dimer in harmonic trap (Busch et al)

For two particles with coordinates x_1, y_1, z_1 and x_2, y_2, z_2 in a parabolic, spherically symmetric trapping potential, we use the following wave function:

For $a > 0$

$$\psi(x_1, y_1, z_1, x_2, y_2, z_2) = \exp\left(-\frac{1}{2}(x_1^2 + y_1^2 + z_1^2 + x_2^2 + y_2^2 + z_2^2)\right) \frac{\exp\left(-\frac{|\mathbf{r}_1 - \mathbf{r}_2|}{a}\right)}{|\mathbf{r}_1 - \mathbf{r}_2|}, \tag{29}$$

For $a < 0$

$$\psi(x_1, y_1, z_1, x_2, y_2, z_2) = \exp\left(-\frac{1}{2}(x_1^2 + y_1^2 + z_1^2 + x_2^2 + y_2^2 + z_2^2)\right)\left(1 - \frac{a}{|\mathbf{r}_1 - \mathbf{r}_2|}\right), \tag{30}$$

D. Two-body problem: dimer in a waveguide

For two particles with coordinates x_1, y_1, z_1 and x_2, y_2, z_2 in a waveguide, we use the following wave function:

$$\psi(x_1, y_1, z_1, x_2, y_2, z_2) = \exp\left(-\frac{1}{2}(x_1^2 + y_1^2 + x_2^2 + y_2^2)\right) \frac{\exp\left(-\frac{|\mathbf{r}_1 - \mathbf{r}_2|}{a}\right)}{|\mathbf{r}_1 - \mathbf{r}_2|}, \tag{31}$$

where $a > 0$ is the s-wave scattering length. The problem of scattering of two particles in a wave guide, i.e. in an external field $V(x, y, z) = m\omega^2(x^2 + y^2)/2$ described by oscillator length $a_{ho} = \sqrt{\hbar^2/(m\omega)}$, was solved by Olshanii. There is an explicit relation between the three-dimensional s-wave scattering length a_{3D} and the one-dimensional one a_{1D}, as [4, 5]

$$a_{1D} = -a_{ho}(a_{ho}/a_{3D} - A) \tag{32}$$

where $A = 1.0326$ is a dimensionless constant.

There is a resonance when $a_{3D} \approx a_{ho}$. Instead, when $|a_{3D}| \ll a_{ho}$, the resonant behaviour can be neglected and one can use the mean-field relation

$$a_{1D} = -a_{ho}^2/a_{3D} \tag{33}$$

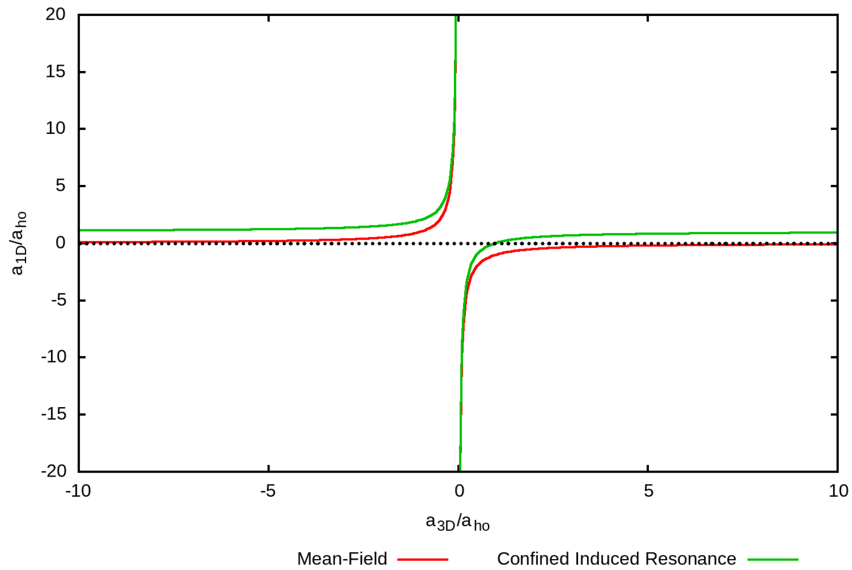


FIG. 3. One-dimensional s-wave scattering length a_{1D} for two atoms in a waveguide of width a_{ho} as a function of the three-dimensional scattering length a_{3D}. Red line, mean-field prediction given by Eq. (33). Green line, exact prediction given by Eq. (32) and describing the Confined-Induced Resonance (crossing with the black dashed line) for $a_{3D} \approx a_{1D}$.

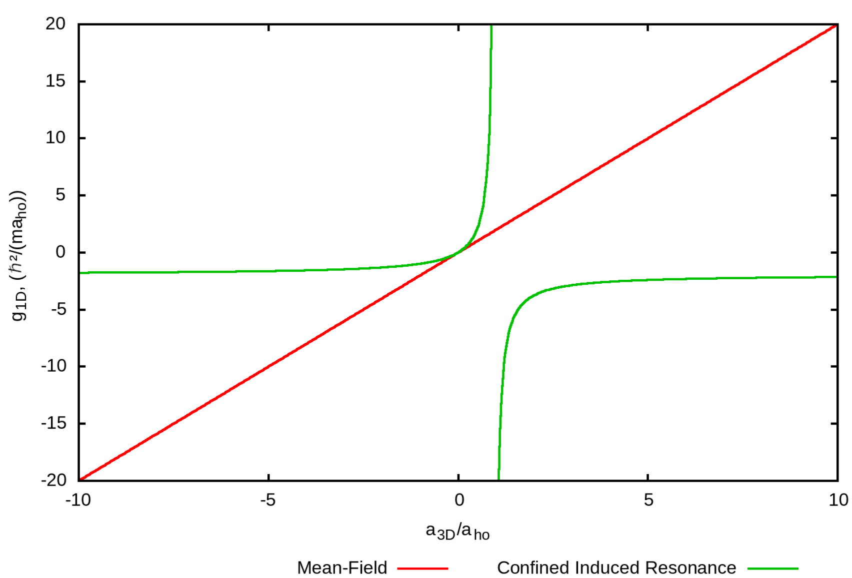


FIG. 4. One-dimensional coupling constant $g_{1D} = -2\hbar^2/(ma_{1D})$ for two atoms in a waveguide of width a_{ho} as a function of the three-dimensional scattering length a_{3D}. Red line, mean-field prediction given by Eq. (33). Green line, exact prediction given by Eq. (32) and describing the Confined-Induced resonance (crossing with the black dashed line) for $a_{3D} \approx a_{1D}$.

E. Many-body problem: variational Monte Carlo solution

The following simple choice of the many-body wave function

$$\psi(\mathbf{r}_1, \cdots, \mathbf{r}_N) = \prod_{i=1}^{N} \exp\left(-\frac{1}{2a_{ho}^2}(x_i^2 + y_i^2)\right) \prod_{i<j}^{N} \left(1 - \frac{a_{3D}}{|\mathbf{r}_i - \mathbf{r}_j|}\right) \tag{34}$$

where $a_{3D} < 0$ satisfies the Bethe-Peierls boundary condition. That is for any two particles with indices i and j approaching each other, $r = r_i - r_j \to 0$,

$$\frac{1}{\psi(r)r}\frac{\partial(\psi(r)r)}{\partial r} = -\frac{1}{a_{3D}} \tag{35}$$

A more advanced choice of the Jastrow terms adds information about the one-dimensional s-wave scattering length a_{1D} and an additional degree of freedom in terms of a variational parameter A_{par}

$$f_2(r) = \left(1 - \frac{r}{a_{3D}}\right) \exp\left(-\frac{r^2}{a_{1D}(A_{par} + r)}\right) \tag{36}$$

The optimal value of the variational parameter is obtained by minimizing the energy. In particular, for $a_{3D} = -0.01a_{ho}$, we get $A_{par} = 5$

$$u_2(r) = \ln\left(1 - \frac{r}{a_{3D}}\right) - \frac{r^2}{a_{1D}(A_{par} + r)} \tag{37}$$

In the limit of weak interactions, $a_{3D} \to -0$, we expect to recover the energy of the McGuire soliton E_{1D} as given by Eq. 19 with the addition of the energy of quantum fluctuations in a harmonic oscillator if the waveguide,

$$E_{3D} = N\hbar\omega + E_{1D} \tag{38}$$

Figure 5 shows the difference between the energy from one-dimensional space, and three-dimensional space, for different particles, the plot show that the three-dimensional energy increase faster than the one-dimensional energy while incriminating the number of particles, as the E_{3D} has a factor of the number of particles added. Also if the value of a_{3D}/a_{ho} was near zero the difference between the energies will be bigger as a_{1D}/a_{ho} will increase near $-\infty$ as shown in Figure 3.

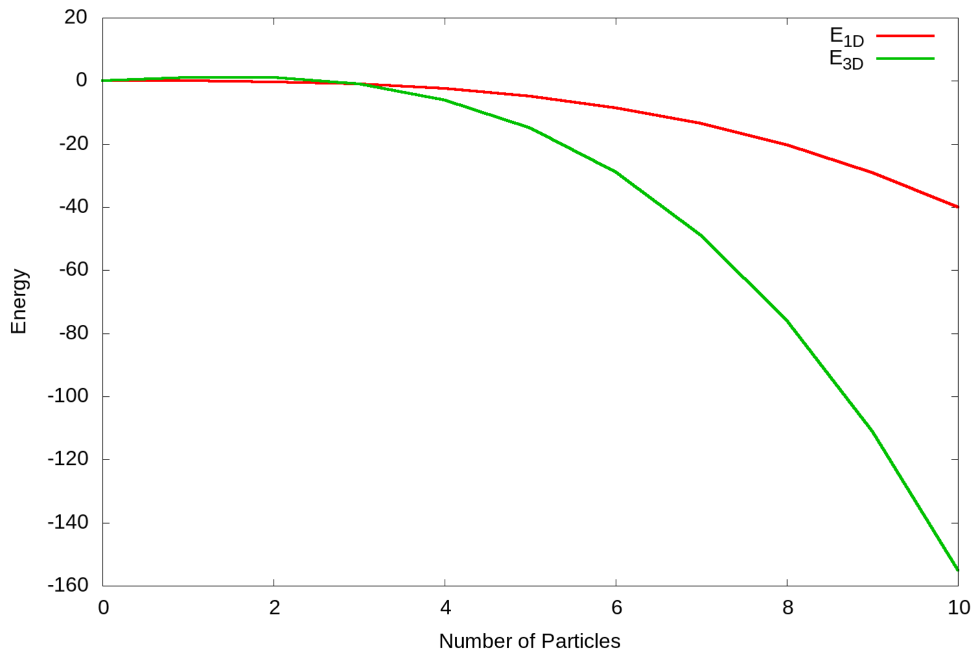


FIG. 5. Energy, using $a_{ho} = 1.0$, $a_{3D}/a_{ho} = -1.0$ and a_{1D} as a result of Eq 32. Red line, McGuire solution, Eq 19, with $a = a_{1D}$. Green line, results of Eq 38 divided by the number of particles, with $a = a_{3D}$.

XIX. RESULTS: ONE-DIMENSIONAL SYSTEM

A. Soliton: energy

Figure 6 shows the comparison between the energy of the Monte Carlo results and the ground-state energy solution from the McGuire equation, Eq. 19, as the figure shows, there is no difference between the two solutions. The McGuire energy is cubic for the number of particles N. The data showed in the figure approach a straight line for $N \gg 2$. The reason for that is that the mean-field parameter (13) is used to relate the s-wave scattering length and the number of particles. Combined with the McGuire equation, the cube of N of the equation divided y the square of N form s-wave scattering length gives a linear equation which, indeed, is obtained in the mean-field theory, Eq. 15.

As it can be seen, the Monte Carlo is in perfect agreement with the exact result.

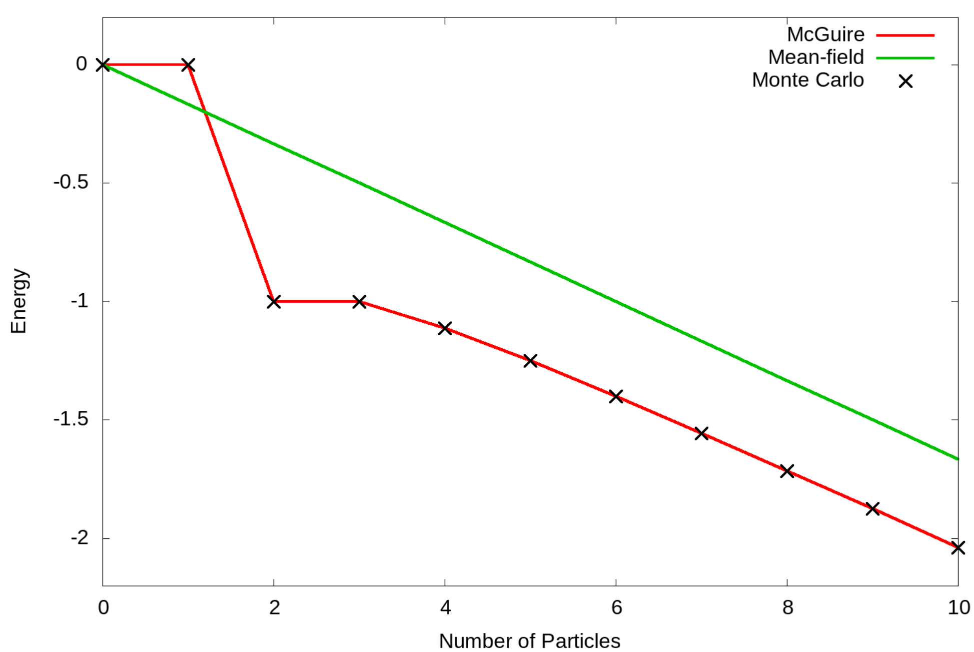


FIG. 6. One-dimensional energy for calculated for $N = 2 - 10$ particles, for different values of the s-wave scattering length a while fixing the mean-field parameter (13) to 1. Black crosses ($\times$), Monte Carlo results. result, Eq. 19. Red solid line, McGuire exact many-body. Green solid line, mean-field energy, Eq. 15.

B. Soliton: density profile

Figure 7 shows the comparison between the density profile obtained in Monte Carlo simulations and the density profile obtained from the Calogero equation. As it shows in the graphic, for the same parameters (N and a), the density profile of both methods follows the same shape and scale.

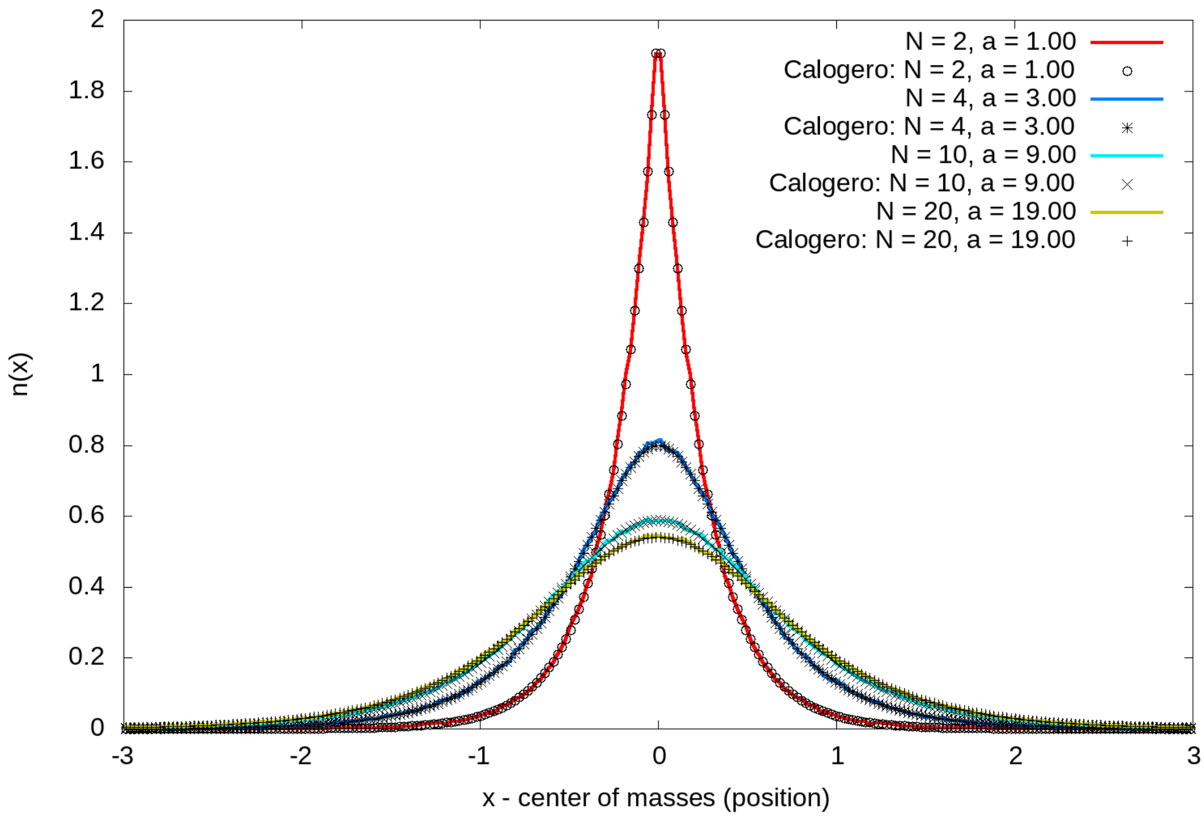


FIG. 7. One-dimensional density profiles for two, four, ten and twenty particles, for different values of the s-wave scattering length a while fixing the mean-field parameter (13) to 1. Color solid lines, Monte Carlo results. Black symbols, Calogero density profiles for different N, Eq. 20. Unit normalization of the density profiles is used, $\int_{-\infty}^{\infty} n(x)dx = 1$.

Figure 8 shows the relation between N and a, instead of fixing the mean-field parameter (13) the s-wave scattering length a is fixed. The results show that with more particles the curve is sharper than with fewer particles, that makes sense since with more particles attracting each other with the same force they will come closer, as the total of attraction is bigger. This figure is to clarify the results in the other figures as in the majority of the other figures the results with fewer particles are sharper than the ones with more particles, that is because in the other figures the mean-field parameter (13), instead of fixing the s-wave scattering length, and while the number of particles N increase, the s-wave scattering length a increase too.

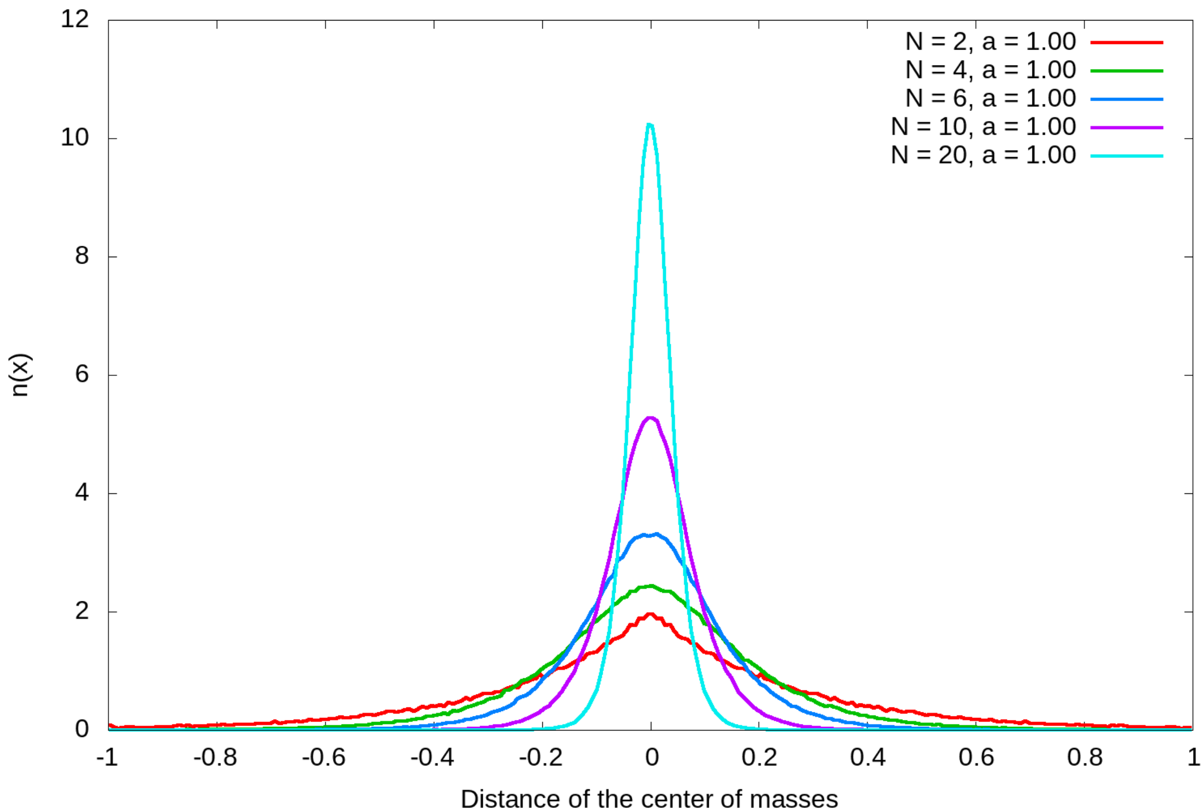


FIG. 8. One-dimensional density profiles for two, four, six, ten and twenty particles, while fixing the s-wave scattering length a to 1. Color solid lines, Monte Carlo results. Unit normalization of the density profiles is used, $\int_{-\infty}^{\infty} n(x)dx = 1$.

Figure 9 shows the density profile obtained in Monte Carlo simulations. For $N = 2$ particles, the density profile coincides with that of the dimer solution (Eq. 10). For large number of particles ($N \gtrsim 100$) the mean-field result, Eq. 12, is recovered.

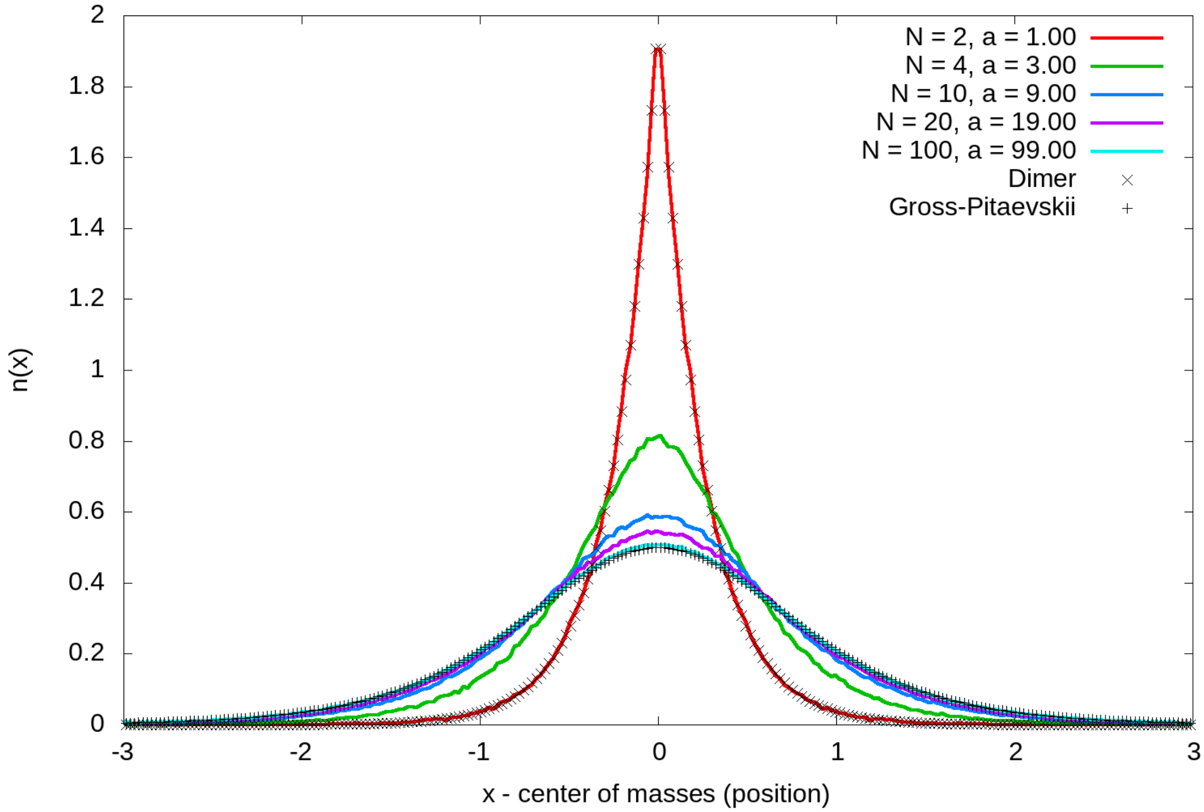


FIG. 9. One-dimensional density profiles for $N = 2, 4, 10, 20, 100$ particles, for different values of the s-wave scattering length a while fixing the mean-field parameter (13) to 1. Color solid lines, Monte Carlo results. Black crosses ($\times$), dimer density profile valid for $N = 2$, Eq. 10. Black pluses ($+$), Gross-Pitaevskii density profile valid for large N, Eq. 12. Unit normalization of the density profiles is used, $\int\limits_{-\infty}^{\infty} n(x)dx = 1$.

Figure 10 reports the comparison of the Monte Carlo result for the density profile obtained for several characteristic parameters of the mean-field parameter P_{MF}. It can be noticed that while spatial size changes, the shape of the curves calculated with a different number of particles are similar. The reason for that is that there is a single unit of length in the system, defined by the s-wave scattering length.

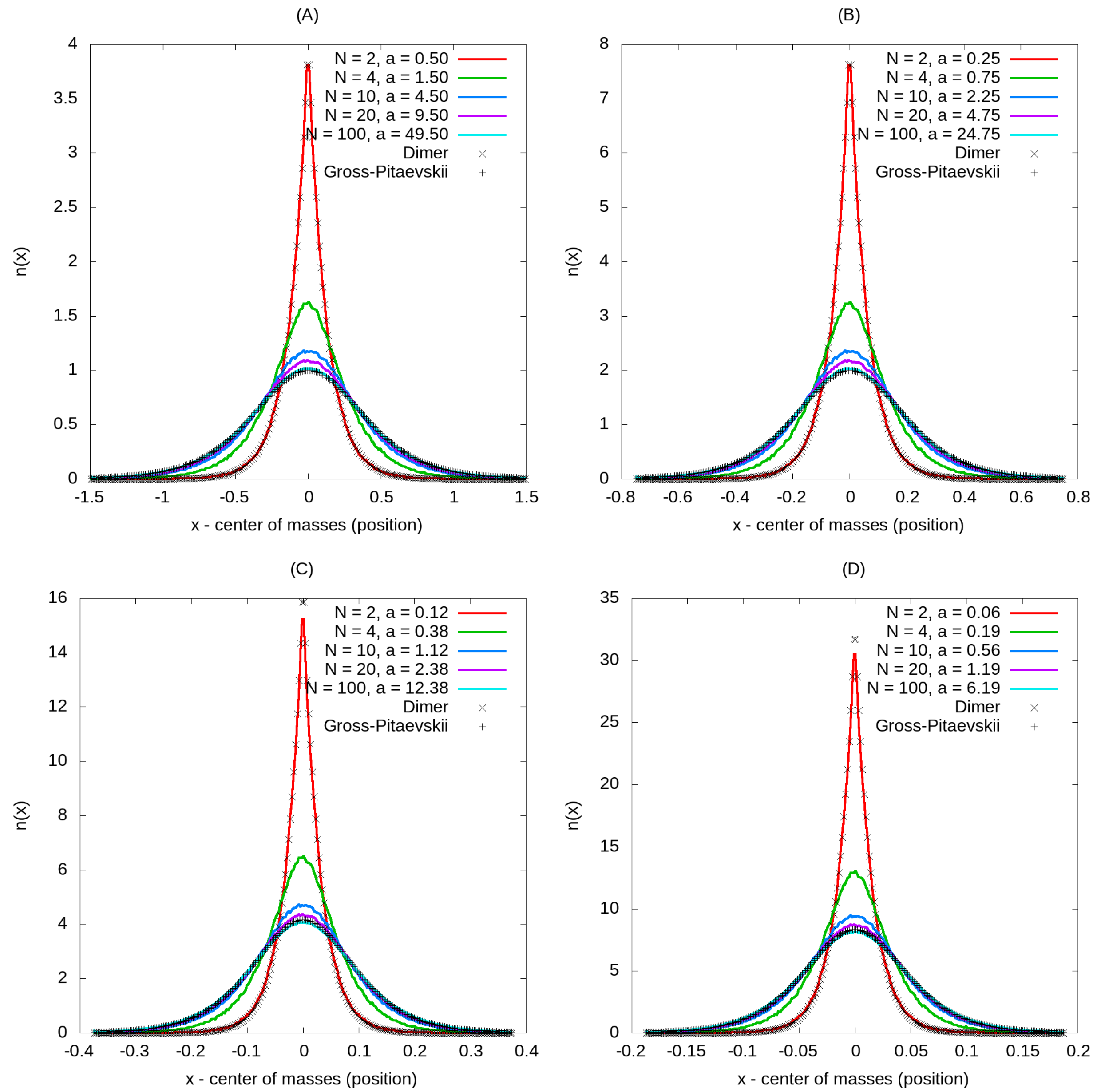


FIG. 10. One-dimensional density profiles for $N = 2, 4, 10, 20, 100$ particles, for different values of the s-wave scattering length a while fixing the mean-field parameter (13) to 2(A), 4 (B), 8 (C), and 16 (D). Color solid lines, Monte Carlo results. Black crosses ($\times$), dimer density profile valid for $N = 2$, Eq. 10. Black pluses ($+$), Gross-Pitaevskii density profile valid for large N, Eq. 12. Unit normalization of the density profiles is used, $\int_{-\infty}^{\infty} n(x)dx = 1$.

In order to quantify the differences between the many-body and MF theories, we integrate the difference between the exact profile $n(x)$ and the mean-field prediction $n_{MF}(x)$, given by Eq. (12),

$$\epsilon = \frac{\int\limits_{-\infty}^{\infty} |n(x) - n_{MF}(x)|\; dx}{\int\limits_{-\infty}^{\infty} n(x)\; dx} \tag{39}$$

Figure 11 shows the error (Eq. 39) as a function of the number of particles. For $N = 100$ particles the error is $\epsilon \approx 0.012$.

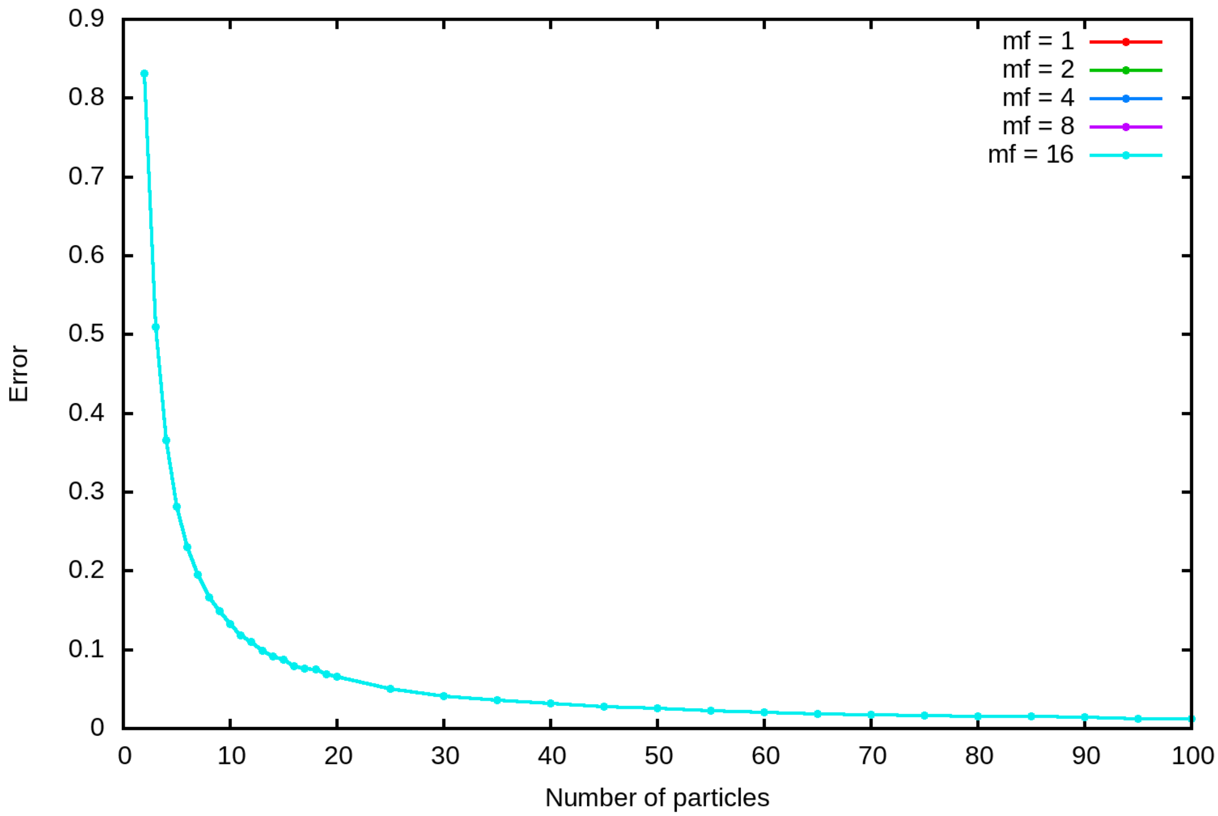


FIG. 11. Error (Eq. 39) for the different mean field parameters (13) with different number of particles, from 2 particles to 100 particles.

C. Soliton: pair distribution function

The mean-field theory neglects two-body correlations and it is interesting to study them. In order to quantify the two-body correlations, we measure the pair distribution function.

$$g_2(x_1, x_2) = \langle n(x_1)n(x_2)\rangle \tag{40}$$

The mean-field prediction for $g_2(x)$

$$g_2^{MF}(x_1, x_2) = n^{MF}(x_1)n^{MF}(x_2) \tag{41}$$

Figure 12, shows three plots obtained with two particles and with the mean-field parameter (13) fixed to 1., the first one is the total pair-distribution from Monte Carlo results, where only the counter-diagonal has values different from zero, this make sense as the center of masses will be always in the middle of two particles, when one particle is positioned "x" coordinates to the right of the center, the other will be always at "$-x$" coordinates. The second plot shows the non-trivial contribution to the pair-distribution function, Eq. 22, that in theory it can be approximated with the Eq. 23, displayed in the third plot, but as the plots show there are totally different, this difference only occurs with two particles.

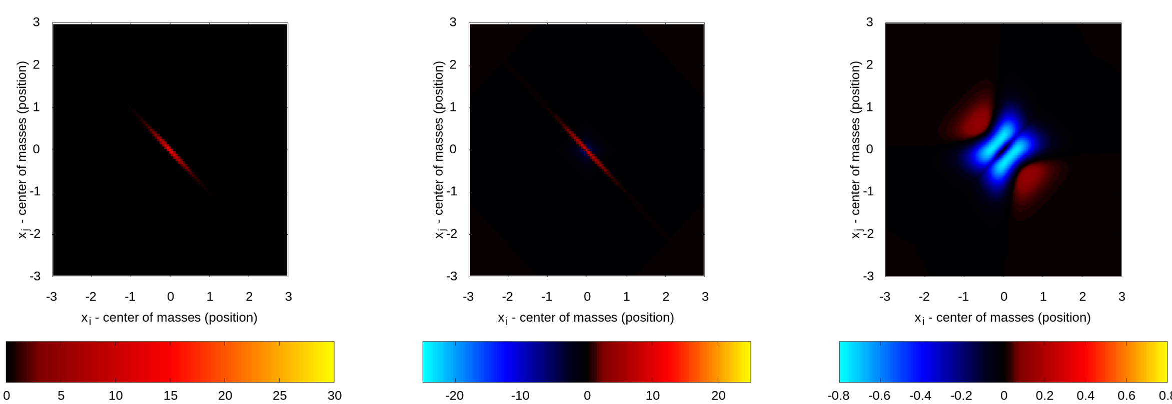


FIG. 12. From left to right (a) total pair-distribution function $g_2(x, y)$ (41), Monte Carlo result (b) non-trivial contribution to the pair-distribution function $dg_2(x, y) = g_2(x, y) - n(x)n(y)$ as predicted by Castin Eq. 22. Monte Carlo, with exact Calogero density profile $n(x)$ subtracted, and (c) the results of Castin approximation, Eq. 23. For $N = 2$ particles, with the mean-field parameter (13) fixed to 1.

Figures 13, 14, 15 and 16, shows the same as Figure 12, but with five, ten, forty and eighty particles. In the first plots in comparison with the first plot for two particles, the particles are more spread between the diagonal, instead of only stay in the counter-diagonal, as a result of this, the other two plots are similar enough to consider the third plots as an approximation of the second ones, obtaining the results given by Castin.

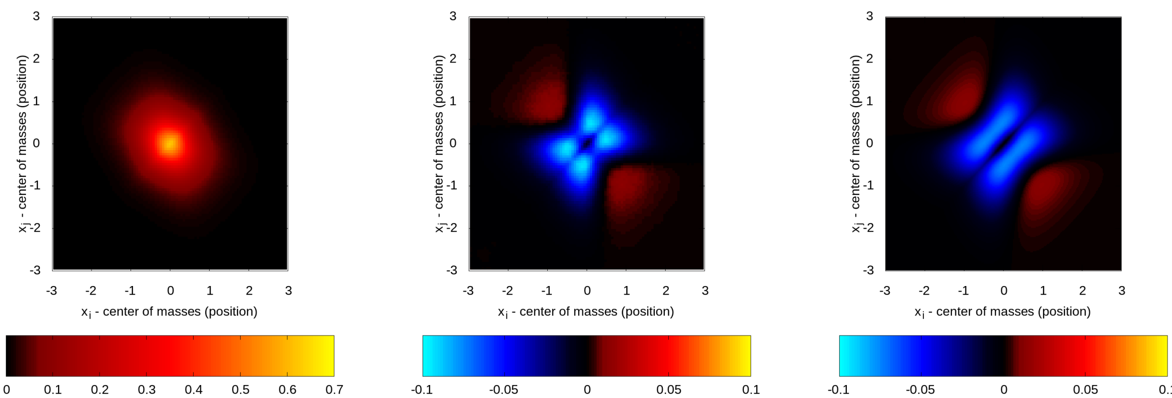


FIG. 13. From left to right (a) total pair-distribution function $g_2(x,y)$ (41), Monte Carlo result (b) non-trivial contribution to the pair-distribution function $dg_2(x,y) = g_2(x,y) - n(x)n(y)$ as predicted by Castin Eq. 22. Monte Carlo, with exact Calogero density profile $n(x)$ subtracted, and (c) the results of Castin approximation, Eq. 23. For $N = 5$ particles, with the mean-field parameter (13) fixed to 1.

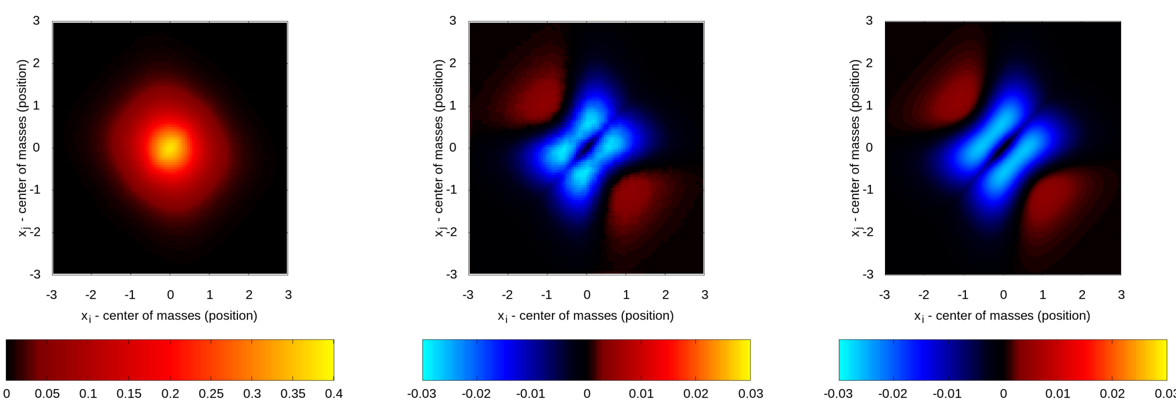


FIG. 14. From left to right (a) total pair-distribution function $g_2(x,y)$ (41), Monte Carlo result (b) non-trivial contribution to the pair-distribution function $dg_2(x,y) = g_2(x,y) - n(x)n(y)$ as predicted by Castin Eq. 22. Monte Carlo, with exact Calogero density profile $n(x)$ subtracted, and (c) the results of Castin approximation, Eq. 23. Eq. 22, and the results of Castin approximation, Eq. 23. For $N = 10$ particles, with the mean-field parameter (13) fixed to 1.

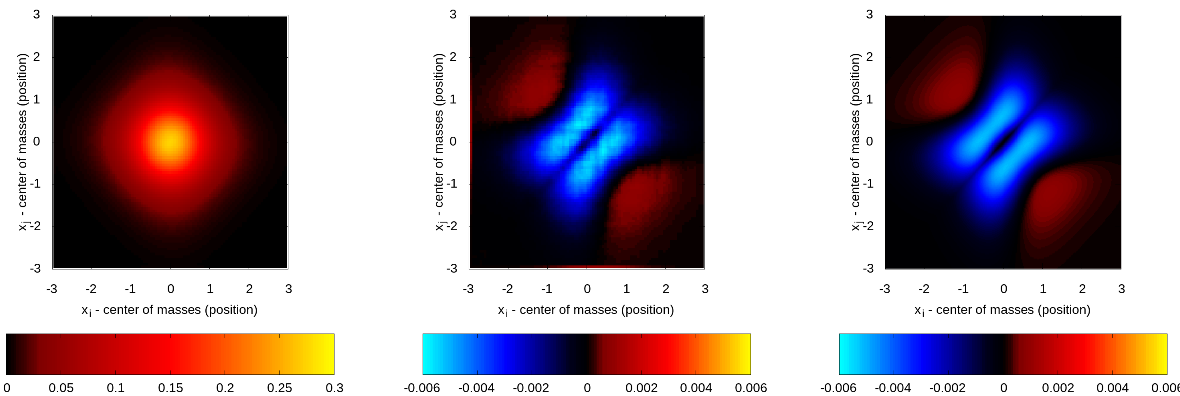


FIG. 15. From left to right (a) total pair-distribution function $g_2(x, y)$ (41), Monte Carlo result (b) non-trivial contribution to the pair-distribution function $dg_2(x, y) = g_2(x, y) - n(x)n(y)$ as predicted by Castin Eq. 22. Monte Carlo, with exact Calogero density profile $n(x)$ subtracted, and (c) the results of Castin approximation, Eq. 23. For $N = 40$ particles, with the mean-field parameter (13) fixed to 1.

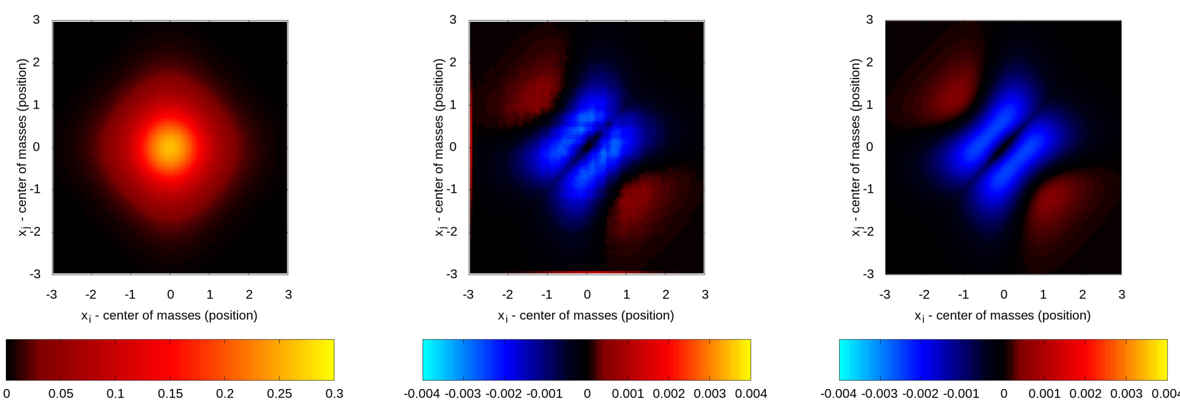


FIG. 16. From left to right (a) total pair-distribution function $g_2(x, y)$ (41), Monte Carlo result (b) non-trivial contribution to the pair-distribution function $dg_2(x, y) = g_2(x, y) - n(x)n(y)$ as predicted by Castin Eq. 22. Monte Carlo, with exact Calogero density profile $n(x)$ subtracted, and (c) the results of Castin approximation, Eq. 23. For $N = 80$ particles, with the mean-field parameter (13) fixed to 1.

Figures 17 and 18 shows the counter-diagonal and diagonal from the previous figures, the results show that with small number of particles Monte Carlo results have a similar shape, but different scale, but with a larger number of particles, there are more similarity between the Monte Carlo results and Castin approximation, Eq 23.

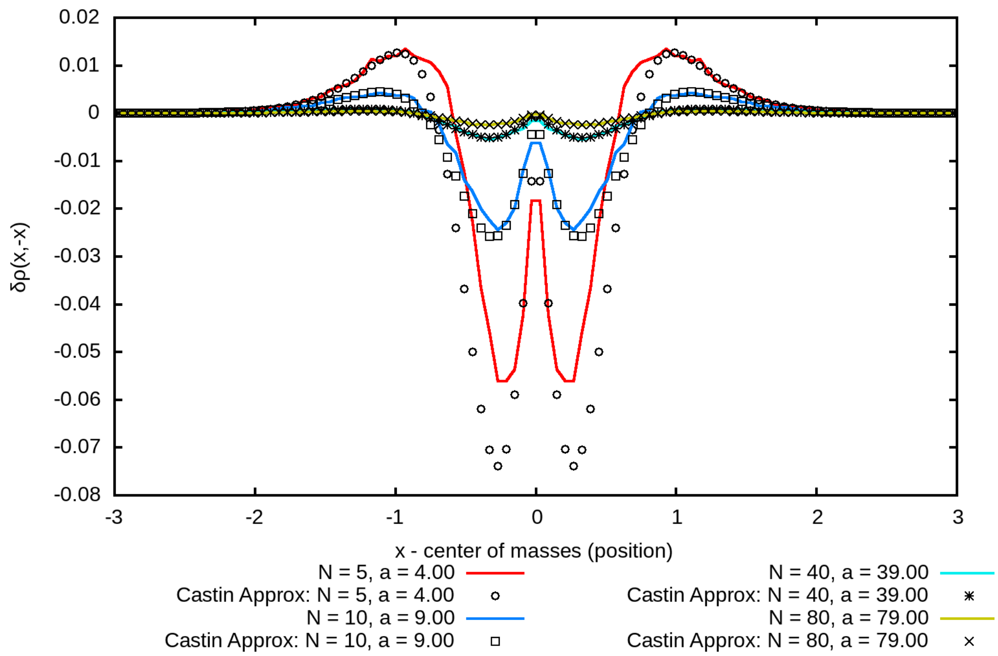


FIG. 17. Counter-diagonals of the pair-distribution function. Color solid lines, Monte Carlo result, Eq. 22. Black symbols, Castin approximation, Eq. 23. For $N = 5, 10, 40, 80$ particles, with the mean-field parameter (13) fixed to 1.

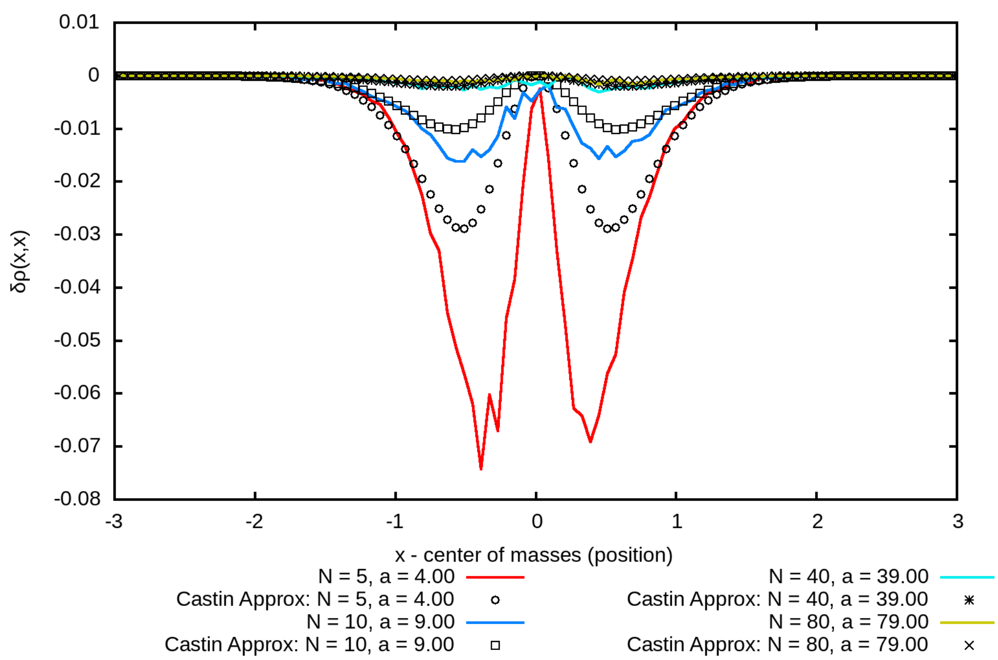


FIG. 18. Diagonals of the pair-distribution function. Color solid lines, Monte Carlo result, Eq. 22. Black symbols, Castin approximation, Eq. 23. For $N = 5, 10, 40, 80$ particles, with the mean-field parameter (13) fixed to 1.

XX. RESULTS: THREE-DIMENSIONAL SYSTEM

A. Soliton: energy

The relation between the a_{3D} and a_{1D} is not implemented correctly in the code, so the results cannot be compared with the one-dimensional results or one-dimensional dependent equations, still are useful results as even if it cannot be compared they show how the system behaves in a waveguide.

Figure 19 shows the comparison between the expected energy and the Monte Carlo energy, as the plot shows the energy is in a very different scale as the relation between a_{1D} and a_{3D} is not correct, but still follows the expected shape.

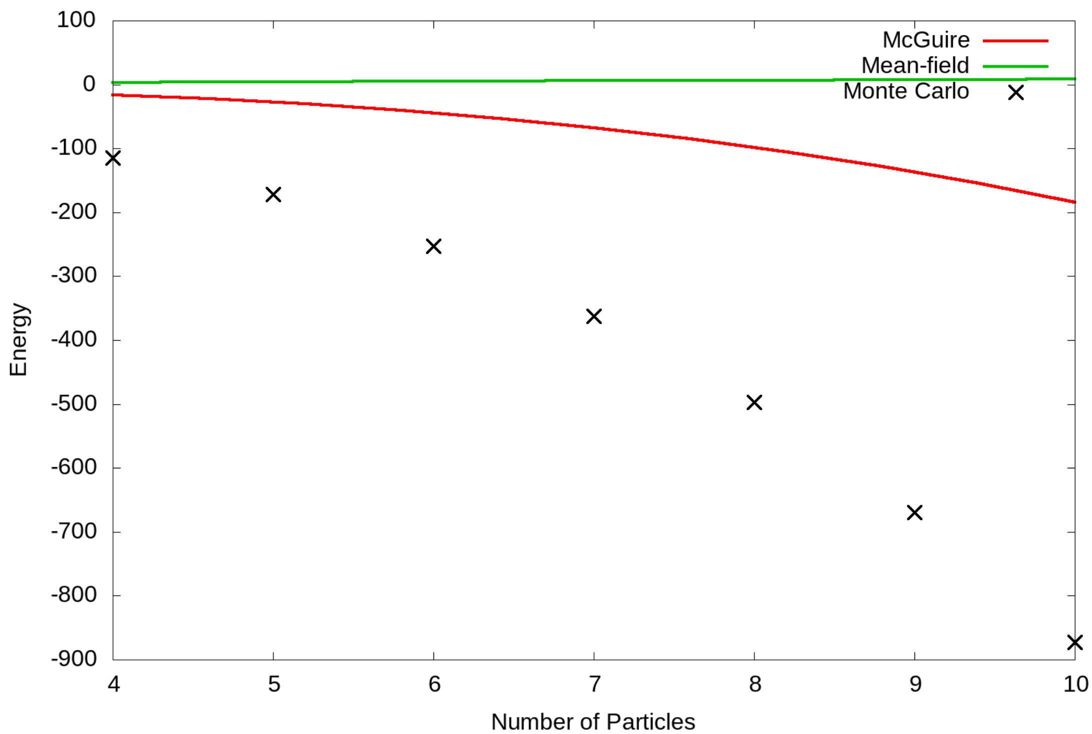


FIG. 19. Three-dimensional energy for $N = 4 - 10$ particles, for different values of the s-wave scattering length a while fixing the mean-field parameter (13) to 1. Color lines, expected energy from two equations, Eq 38, with E_{1D} from equations 19 and 15. Black crosses ($\times$), Monte Carlo results.

B. Soliton: density profile

Figure 20 shows the comparison between density profiles for a different number of particles in the x and y axis where there is a harmonic trap, the two plots are very similar, as should be, as the trap is the same on both axes. The plots show the density profile centered in zero where the harmonic trap forces to stay, also it can be noticed that as the number of particles increase, the density profile gets sharper.

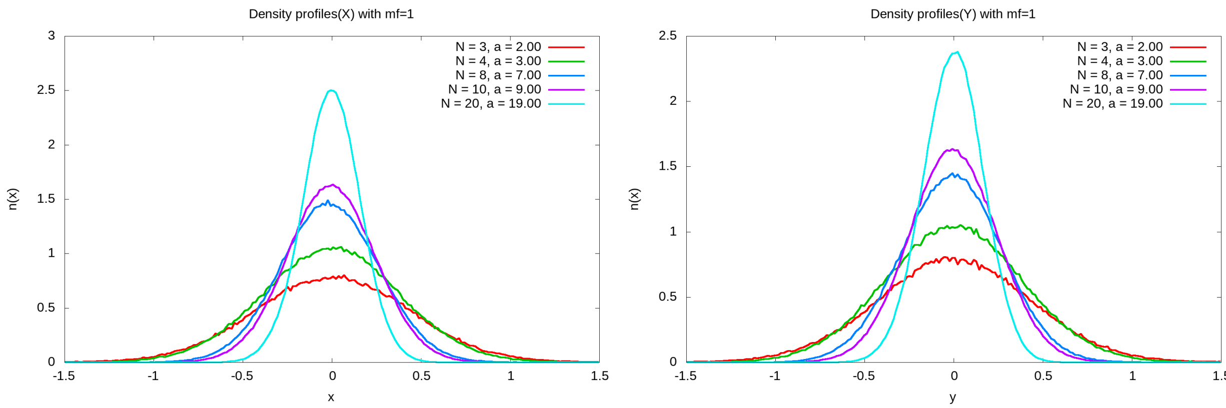


FIG. 20. Three-dimensional density profiles of the x and y axis, for $N = 3, 4, 8, 10, 20$ particles, for different values of the s-wave scattering length a while fixing the mean-field parameter (13) to 1. Color solid lines, Monte Carlo results. Unit normalization of the density profiles is used, $\int_{-\infty}^{\infty} n(x)dx = 1$.

Figure 21 shows the density profile in the free axis (z) respect to the center of masses, the results, in this case, follow a similar shape to the trapped axis from the previous figure, it shows that with a bigger number of particles, the density profile gets sharper, this behaviour is different of the one-dimensional density profile, where the density becomes less sharp with more particles.

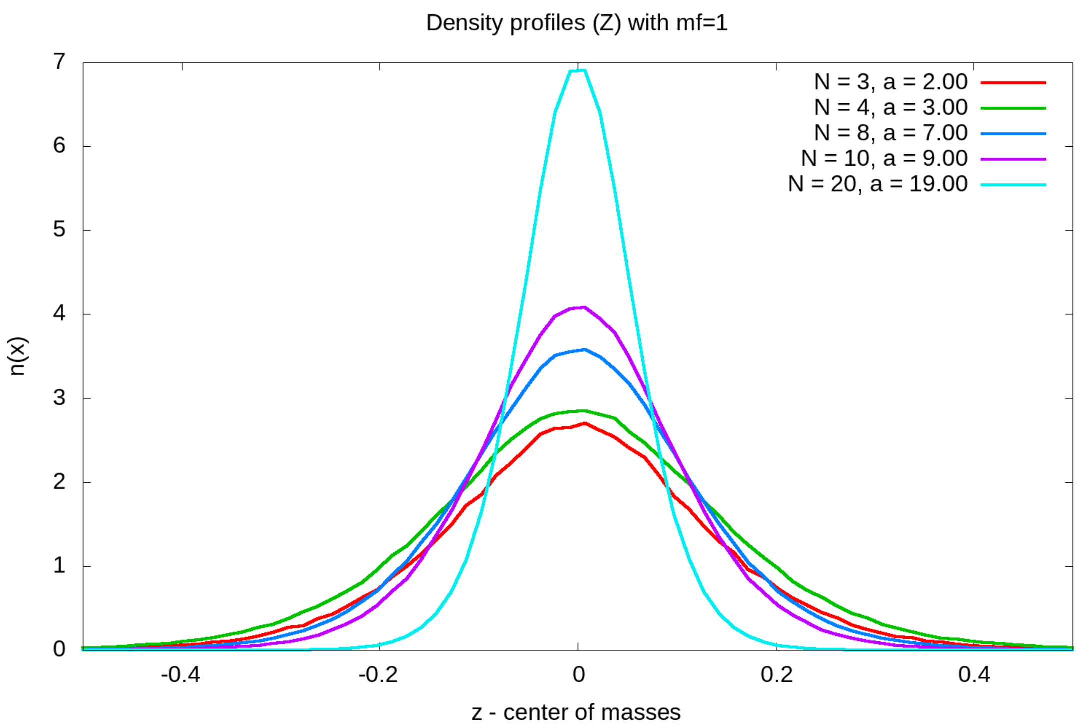


FIG. 21. Three-dimensional density profiles of the z respect the center of masses, for $N = 3, 4, 8, 10, 20$ particles, for different values of the s-wave scattering length a while fixing the mean-field parameter (13) to 1. Color solid lines, Monte Carlo results. Unit normalization of the density profiles is used, $\int_{-\infty}^{\infty} n(x)dx = 1$.

XXI. FUTURE WORK

The code is finished for one-dimensional space, but still need some work to do in a three-dimensional system and a harmonic trap. As the relation between a_{1D} and a_{3D} is not done correctly as shows the energy experiment in a three-dimensional system. To finish it is necessary to implement the parameter a_{ho} to find a_{1D} correctly.
Once finished, the experiments can continue, to find solutions that can be compared with already known equations to prove the correctness of three-dimensional results.
Finally, when all the results of the code are checked, the final part is to explore in-depth the result and extract equations to define the system.

XXII. CONCLUSIONS

For this project is needed a lot of physics background that is hard to get in four months, but still with the help of Grigori, we reach the majority of the objectives set at the beginning of the project.

The code is practically finished, even if at the end we couldn't check the correctness of results for three-dimensional space, we checked the correctness of one-dimensional space, as well as the correctness of the code structure, with different experiments, showing that in all the cases we get the same results as the already proven equations.

The three-dimensional space is nearly done, only needs to finish the implementation of the relation between a_{1D} and a_{3D}, to compare correctly with one-dimensional solutions, but is already working.

Finally, the code also permits the addition of a harmonic trap in the free axis, in one and three-dimensional systems, so it can explore more solutions for a different experiment.

XXIII. BIBLIOGRAPHY

[1] J. B. McGuire, "Study of exactly soluble one-dimensional n-body problems," Journal of Mathematical Physics **5**, 622–636 (1964).

[2] F. Calogero and A. Degasperis, "Comparison between the exact and hartree solutions of a one-dimensional many-body problem," Phys. Rev. A **11**, 265–269 (1975).

[3] Y. Castin, "Internal structure of a quantum soliton and classical excitations due to trap opening," The European Physical Journal B **68**, 317–328 (2008).

[4] M. Olshanii, "Atomic scattering in the presence of an external confinement and a gas of impenetrable bosons," Phys. Rev. Lett. **81**, 938–941 (1998).

[5] T. Bergeman, M. G. Moore, and M. Olshanii, "Atom-atom scattering under cylindrical harmonic confinement: Numerical and analytic studies of the confinement induced resonance," Phys. Rev. Lett. **91**, 163201 (2003).