UNIVERSITAT DE BARCELONA

ICCUB & IN2UB

START: 01/07/2022, END:02/02/2023

Soliton dynamics in the Gross Pitaevskii equation

Author
Marc FARRERAS (Physics)

Supervisors
Dr. Carles CALERO
& Dr. Bruno JULIÁ

Summary

1 Introduction

This report contains all my experience and work during my internship with IC-CUB, carried out between July and September 2022. During the internship I worked with Eric Vidal Marcos in the project QuantumlabUB, supervised by Dr. Bruno Juliá Díaz and Dr. Carles Calero Borrallo.

1.1 The project

The QuantumlabUB is a project build by undergraduate students in Physics from the University of Barcelona, with the aim to develop applications to popularise aspects of quantum mechanics. In particular, my peer Eric developed an application to show the Spin Tunneling experiment, while I decided to redo an old project started by Ivan Morera and Laura Moreno to show soliton dynamics in the Gross Pitaevskii equation .

Even though the project carried out by Ivan Morera and Laura Moreno was completed, the code is not functional now due to changes in the python libraries used. For this reason and to make my experience more enlightening, Bruno pointed out that would be better to start from zero the project.

1.2 Goals

During our first reunion, Bruno explained the steps in order to solve the Gross Pitaevskii equation and study the soliton dynamics. The different goals set up to scheme the work were:

- Familiarize with the GPE and Split-step method.
- Study of the Fourier Transform and Discrete Fourier Transforms.
- Program the Split-step method and solve numerically the Schrödinger equation for free particle and harmonic oscillator.
- Solve numerically the GPE and compare with analytical results.
- Program the application

Due to lack of time to finish properly the application, the memorial was written with the objective to be as useful as possible for the next ones who continue the project. For this reason all the theoretical information, numerical comparisons and calculations are written in detail.

1.3 Methodology

One of the objective of the whole experience was to familiarize the students with the professional world. For this reason, the scheme work followed during the project, which mainly last all July and September, was similar to how the research groups work.

Two or three times a week, online meetings were arranged with our professors where Eric and myself exposed all the work that was done from the last meeting. This reunions where essential for us, it allowed us to discuss our results, problems and thoughts with our tutors. They in return could give us advise, helped us to build the next objective and lead us to a new paths whenever someone got stuck in a specific problem.

Between the weekly meetings, Eric and myself used to meet. This meetings acted like a rehearsal before our reunion with our professors, which not only let ourselves learn from the experiences of each other, but also it was essential to organize our thoughts and be able of present clarifying reports to our professors.

2 Theoretical background

2.1 Gross-Pitaevskii equation

The Gross-Pitaevskii equation, also known as GPE or nonlinear Schrödinger equation, has a wide variety of applications in nonlinear optics and quantum physics. In particular is used to describe the wave function of a Bose-Einstein condensate or BEC.

A BEC is formed when a gas of bosons trapped in and external potential is cooled to temperatures very close to $T \to 0\,K$. Due to bosons follow the Bose-Einstein statistics they can occupy the same state.

If the bosons do not interact with each other, each one of them occupies the ground state of the trap. The problem shows up when the bosons does interact between them, then the ground state occupied by the bosons is not the same as the ground state of the external potential and one is forced to find the ground state of the many body hamiltonian

$$\mathcal{H} = \sum_{i=1}^{N} \left[\frac{\vec{p}_i^2}{2m} + V_{ext}(\vec{r_i}) \right] + \frac{1}{2} \sum_{i \neq j} V(r_{ij}). \tag{1}$$

Using the variational method and the Hartree-Frock approximation the Hartree equation for bosons is obtained

$$\left[-\frac{\hbar^2\nabla^2}{2m} + V_{ext}(\vec{r}) + (N-1)\int d\vec{r'}|\varphi(\vec{r'})|^2 V(|\vec{r}-\vec{r'}|)\right]\varphi(\vec{r}) = \mu\varphi(\vec{r}). \qquad (2)$$

Where φ is the monoparticular state and μ is the chemical potential of the system. Solving this differential equation should give us the ground state of our N-bosons system.

2.1.1 Time independent Gross-Pitaevskii equation

For the interaction term can be used a pseudopotential $V = g\delta(\vec{r}_1 - \vec{r}_2)$. Using this pseudopotential and the limit of large N in eq. (2), the equation can be written as

$$-\frac{\hbar^2\nabla^2}{2m}\varphi(\vec{r}) + V_{ext}(\vec{r})\varphi(\vec{r}) + gN|\varphi(\vec{r})|^2\varphi(\vec{r}) = \mu\varphi(\vec{r}). \qquad (3)$$

This last equation is the $3D$ time independent Gross-Pitaevskii equation, which is equal to the Schrödinger equation with an additional nonlinear term $gN|\varphi|^2$.

2.1.2 Time dependent Gross-Pitaevskii equation

Moreover, one can obtain the time dependent GPE, which describes the better monoparticular state in each time.

$$i\hbar\partial_t\varphi(\vec{r},t) = -\frac{\hbar^2\nabla^2}{2m}\varphi(\vec{r},t) + V_{ext}(\vec{r})\varphi(\vec{r},t) + gN|\varphi(\vec{r},t)|^2\varphi(\vec{r},t). \qquad (4)$$

2.1.3 1D GPE in harmonic trap

The aim of this project is to solve the GPE with an harmonic potential in $1D$. In this case and using oscillator units the GPE and the time evolution corresponds to

$$-\frac{1}{2}\frac{\partial^2}{\partial\bar{x}^2}\bar{\varphi}(\bar{x}) + \frac{1}{2}\bar{x}^2\bar{\varphi}(\bar{x}) + \bar{g}|\bar{\varphi}(\bar{x})|^2\bar{\varphi}(\bar{x}) = \bar{\mu}\bar{\varphi}(\bar{x}),$$

$$i\partial_{\bar{t}}\bar{\varphi}(\bar{x},\bar{t}) = -\frac{1}{2}\frac{\partial^2}{\partial\bar{x}^2}\bar{\varphi}(\bar{x}) + \frac{1}{2}\bar{x}^2\bar{\varphi}(\bar{x},\bar{t}) + \bar{g}|\bar{\varphi}(\bar{x},\bar{t})|^2\bar{\varphi}(\bar{x},\bar{t}). \qquad (5)$$

2.1.4   Soliton solution

In mathematics and physics, a soliton or a solitary wave is a wave packet that maintains its shape while it propagates a constant velocity. This characteristics let them to have particle-like properties, for example interact between them without suffer interference effects.

The soliton solution is normally related to non-linear equations as the GPE. Specifically the GPE has two soliton-like solutions, the bright and dark soliton. Bright solitons are obtained for attractive interparticle interaction, consequently they are associated to a local maxima of the BEC density. On the contrary, the dark solitons appear in repulsive interparticle interaction and they correspond to a localized density suppression.

Regarding the GPE, the term which module the interparticle interaction is the coupling term g. Bright soliton solutions are obtained when $g < 0$, while $g > 0$ are dark soliton solutions. To learn more of GPE and soliton solution check [1] and [5].

2.2   Split step method

In order to simulate a system, one have to solve the differential equation which describes its dynamics. In the case of the GPE the numerical method used is the split-step method.

For pedagogical reasons, first will be showed how to solve the Schrödinger equation with the split-step method and then it will be applied to the GPE.

2.2.1   Numerical resolution for time dependent Schrödinger equation

Time dependent dependent Schrödinger equation corresponds to

$$i\hbar\partial_t\psi = H\psi = [T + V]\psi. \tag{6}$$

From the eq. (6) is directly derived that the wave function time dependent solution for $\psi(t, x)$ is:

$$\psi(t, x) = e^{-\frac{iHt}{\hbar}}\psi(t = 0, x) = e^{-\frac{i(T+V)t}{\hbar}}\psi(t = 0, x). \tag{7}$$

So for a time step from t to $t + dt$ it can be written formally:

$$\psi(t + dt, x) = e^{-\frac{i(T+V)dt}{\hbar}}\psi(t, x). \tag{8}$$

If the exponential could be separated respectively into one exponential with only momentum operator arguments and another with only spatial operators arguments, it would be straightforward to calculate the differential equation

numerically due to the fact that each operator is an scalar in their own space

Position space (x): momentum space (p):

$$e^{-\frac{idt}{\hbar}V} \rightarrow e^{-\frac{idt}{\hbar}V(x)} \qquad\qquad e^{-\frac{idt}{\hbar}V} \rightarrow e^{-\frac{idt}{\hbar}V(i\hbar\frac{\partial}{\partial p})} \qquad (9)$$

$$e^{-\frac{idt}{\hbar}T} \rightarrow e^{\frac{i\hbar dt}{2m}\frac{\partial^2}{\partial x^2}} \qquad\qquad\qquad e^{-\frac{idt}{\hbar}T} \rightarrow e^{-\frac{idt}{\hbar}\frac{p^2}{2m}}$$

The problem is that the T and V operators do not commute, so we can not calculate it directly. Here is where the split-step methods comes handy. Split-step methods try to avoid the non-commutative problem by approximating:

$$e^{\lambda}(A + B) \approx e^{\lambda\beta_n B}e^{\lambda\alpha_n A}...\, e^{\lambda\beta_1 B}e^{\lambda\alpha_1 A}. \qquad (10)$$

The philosophy of split-operator lies on pick the coefficients α_i and β_i so the right hand part of the equation approximate the left side to as high order in λ wanted. In [3] a coefficient table can be found with the error associated to coefficient combination. For our project an error of $\mathcal{O}(\lambda^3)$ is enough, so the respectively coefficients are $\alpha_1 = \alpha_2 = \frac{1}{2}$ and $\beta_1 = 1$. Finally the eq. (8) approximated is:

$$\psi(t + dt, x) = [e^{\frac{-idt}{2\hbar}V(x)}e^{-\frac{idt}{\hbar}T}e^{\frac{-idt}{2\hbar}V(x)}]\psi(t, x) + \mathcal{O}(\lambda^3). \qquad (11)$$

As mentioned before the exponential operators $1D$ in their respective spaces correspond to scalars. Taking advantage from the Fourier properties, the simplest and direct way to compute it would be:

$$\begin{aligned}
\psi_0(x) &= \psi(t_0, x), \\
\tilde{\psi}_1(p) &= \mathcal{F}[e^{\frac{-idt}{2\hbar}V(x)}\psi_0(x)], \\
\psi_2(x) &= \mathcal{F}^{-1}[e^{-\frac{idt}{\hbar}\frac{p^2}{2m}}\tilde{\psi}_1(p)], \\
\psi(t_0 + dt, x) &= e^{\frac{-idt}{2\hbar}V(x)}\psi_2(x).
\end{aligned} \qquad (12)$$

2.2.2 Imaginary time propagation method

Another really interesting feature of the split-step method is how it enables us not only to solve the time evolution of our wave function, but also can solve the Schrödinger eigenvalue problem.

To demonstrate how, one takes the time dependent Schrödinger equation, which corresponds to eq. (6), and change real time for an imaginary $t = -i\tau$. Now the Schrödinger equation becomes a diffusion equation

$$\begin{aligned}
-\hbar\partial_\tau\psi(x, \tau) &= H\psi(x, \tau), \\
\psi(x, 0) &= \psi_0(x).
\end{aligned} \qquad (13)$$

6

Analytical solution to this differential equation corresponds to an exponential

$$\psi(x,\tau) = e^{-\frac{\tau H}{\hbar}}\psi_0(x). \tag{14}$$

If one now expand the initial wave function ψ_0 in the base of eigenfunctions ϕ_i and the apply diffusion operator

$$\psi_0(x) = \sum_i <\phi_i|\psi_0> \phi_i = \sum_i c_i\phi_i,$$
$$\psi(x,\tau) = e^{-\frac{\tau H}{\hbar}}\sum_i c_i\phi_i, \tag{15}$$
$$\psi(x,\tau) = \sum_i c_i e^{-\frac{\tau E_i}{\hbar}}\phi_i.$$

Where E_i are the eigenvalues from each eigenfunction H_i. Assuming $0 < E_1 < E_2 < ...$, for enough long time $(\tau \to \infty)$ asymptotically the function becomes $\psi(x,\tau) \to c_0 e^{-\frac{\tau E_0}{\hbar}}\phi_0$.

From here, one can obtain the grounded state normalizing the asymptotic function, then the eigenvalue is computed by energy expected value $<\phi_0|H|\phi_0>$.

2.2.3   Numerical resolution for GPE

As result of Schrödinger equation and GPE similarities one would be tempted to add the nonlinear term $g|\psi|^2$ to the potential and apply directly the split-step Fourier method. Nonetheless, wave function evolve in time, for this reason a lineal combination of the different intermediate steps wave functions has to be taken into account. Specifically in the method developed here

$$\psi_0(x) = \psi(t_0, x),$$
$$\tilde{\psi}_1(p) = \mathcal{F}[e^{\frac{-idt}{2\hbar}(V(x)+g|\psi_0(x)|^2)}\psi_0(x)],$$
$$\psi_2(x) = \mathcal{F}^{-1}[e^{-\frac{idt}{\hbar}\frac{p^2}{2m}}\tilde{\psi}_1(p)], \tag{16}$$
$$\psi(t_0+dt, x) = e^{\frac{-idt}{2\hbar}[V(x)+g|c_0\psi_0(x)+c_1\psi_1(x)+c_2\psi_2(x)|^2]}\psi_2(x).$$

Accordingly to [3], the method gives an error $\mathcal{O}(h^3)$ if and only if $c_1 = -c_0$ and $c_2 = \pm 1$.

The final expression is obtained substituting the calculated coefficients in eq. (16)

$$\psi_0(x) = \psi(t_0, x),$$
$$\tilde{\psi}_1(p) = \mathcal{F}[e^{\frac{-idt}{2\hbar}(V(x)+g|\psi_0(x)|^2)}\psi_0(x)],$$
$$\psi_2(x) = \mathcal{F}^{-1}[e^{-\frac{idt}{\hbar}\frac{p^2}{2m}}\tilde{\psi}_1(p)], \tag{17}$$
$$\psi(t_0+dt, x) = e^{\frac{-idt}{2\hbar}[V(x)+g|\psi_0(x)-\psi_1(x)+\psi_2(x)|^2]}\psi_2(x).$$

2.3 Fourier transform

Split-step methods relies in the Fourier transform to pass from one space to other. On account of the importance that Fourier transform has in the numerical method it is essential to achieve a good understanding of Fourier transform fundamentals and how it is performed numerically.

2.3.1 Continuous Fourier transform (FT)

Continuous Fourier transform, or as it will be refereed here FT, is a mathematical tool developed by Jean Baptiste Joseph Fourier which decompose a function in a sum of cosines and sines function. Mathematically the FT of a function $h(x)$ or $H(u) = \mathcal{F}[h(x)]$ and his inverse or $h(x) = \mathcal{F}^{-1}[H(u)]$ are described as

$$H(u) = \int_{-\infty}^{\infty} h(x)e^{-2\pi i x u}dx, \tag{18}$$

$$h(x) = \int_{-\infty}^{\infty} H(u)e^{2\pi i x u}du. \tag{19}$$

Where the first equation corresponds to the properly FT, and the second one is the inverse Fourier transform or IFT.

For many purposes, as will be our case, it is useful to think of $h(x)$ and $H(u)$ as being two different representations of the same function. One goes back and forth between the two representations by means of the FT. Another interesting feature is the dimensional relation between x and u, the units of u are the inverse units of x. For our future interest it is better to check one example, if x are in meters, the units of u will be cycles per meter so the wave number k can be compute as $k = 2\pi u$.

Even thought the last definition of FT is correct, physicist are used to another possible definition

$$H(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h(x)e^{-ixk}dx, \tag{20}$$

$$h(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} H(k)e^{ixk}dk. \tag{21}$$

It is obvious the relationship between k and u comparing both representations, which is $k = 2\pi u$. Hopefully the last example now is more understandable. Due to less 2π factors and more directly relationship in dimensionality between variables the first one representation is used for the next chapter development.

2.3.2   Discrete Fourier transform (DFT)

Fourier transform is not only one of the most useful tools in mathematics but also in computing. However, in a great range of applications numerically calculations are performed, so continuous functions are replaced by sampled ones. This situation raised a problem, this is why a discrete version of FT is needed in these cases. The discrete Fourier transform is named DFT.

In fact the discretization of FT presents two problems that they will be analysed in this chapter. The first and more obvious one is how it is performed, the second one and more subtle comes with how can the information be recovered, which is related with theory information and sample theorem.

Sampling theorem and Aliasing

Function discretization is a delicate operation which should be done carefully. If the function is not well sampled, it could be indistinguishable from another.Then when the DFT is applied could give as result a different outcome from the expected one.

A really clarifying example and really easy to understand will be the following case, imagine you want to sample a function $f(x) = sin(2\pi)$.

In Fig. 1 it is observable the referred sampling problem, using not enough points the functions seems to be a sinusoidal one, but now the frequency of this sinusoidal is totally different than the real one. As more complex is the function sampled, for example more change in its curvature, more points are needed to encode in a proper way the information of the function. Now, the question is, how many N points needs the sample in order to be properly reconstructed? Sampling theorem gives us an answer to this particular problem. But instead of speaking of points N, it answers in terms of frequency f which according to our definitions of FT would correspond to u variable.

Suppose you are using a sample with N points, the sample range is the length (L) of our data. Denoting Δ the space between samples. Then the sample rate is the reciprocal of Δ, which corresponds to number of samples recorded per second.

The Nyquisit-Shannon sampling theorem describes how to sample a signal in such a way no information is lost. It states that to sample the function correctly, the sample frequency has to be equal or grater than the double of the maximum function frequency. Alternatively it is possible to define a critical frequency f_c given by
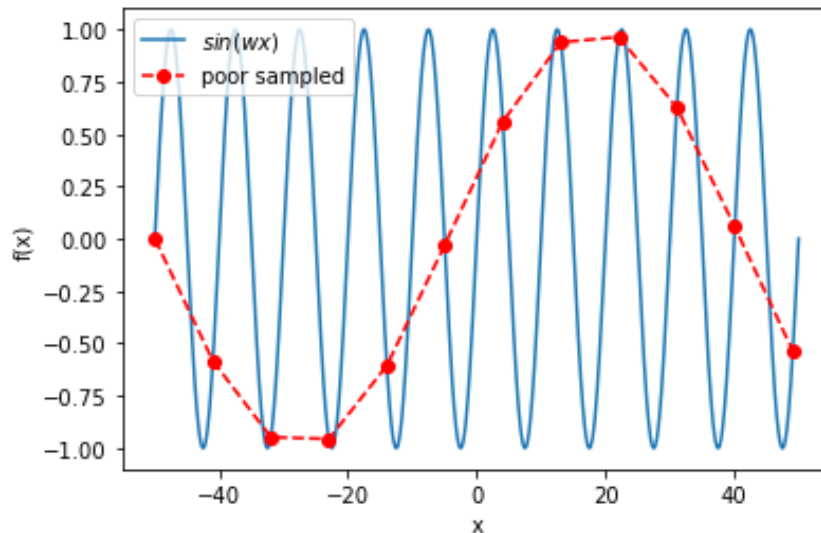
Figure 1: Sinus function plotted. It shows the effect of sampling with not enough points. The sinusoidal form is kept, however the frequency of the function is changed.

$$f_c = \frac{1}{2f_{sample}} = \frac{1}{2\Delta}. \tag{22}$$

In the case $f_c \geq f_{max}$ the function is well sampled, so the original signal can be completely recovered or reconstructed by these samples alone. On the other hand, if $f_c < f_{max}$ then the function is poor sampled. Consequently frequency information outside the $-f_c$ and f_c suddenly move into this range. This effect is called aliasing, an example of aliasing can be seen in Fig. 2

Sometimes the f_{max} of a function is unknown, so a priory the minium N needed to sample can not be estimated. Despite this, what it is possible to do is calculate the Fourier transform to tell whether the continuous function has been competently sampled and try to minimize the aliasing effect. This is done by looking to see whether the Fourier transform is already approaching zero as the frequency approaches fc from below or from above. If, on the contrary, the transform is going toward some finite value, then chances are that components outside of the range have been folded back over onto the critical range.
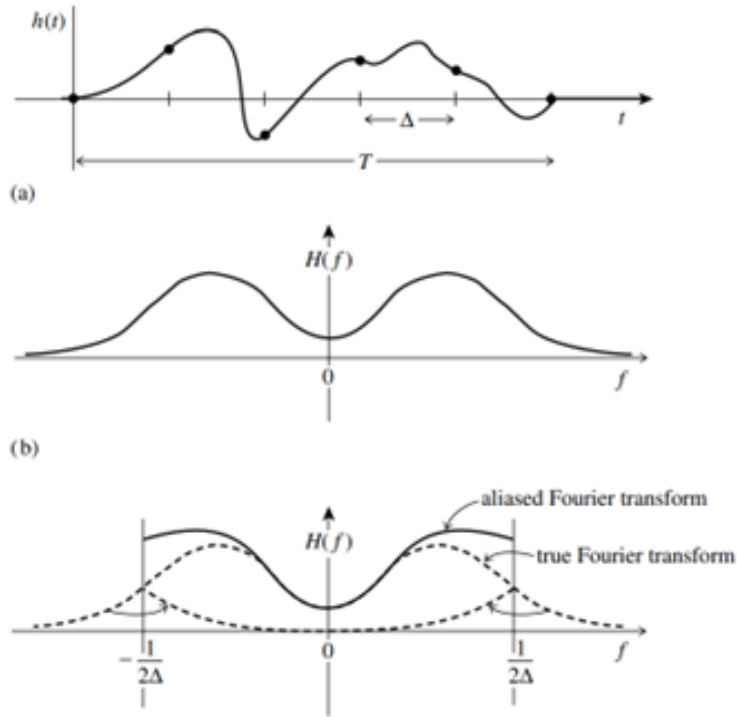
Figure 2: Example of aliasing effect extract from the book Numerical recipes.

DFT computation

The function wanted to transform is $h(x)$, which is sampled in N points. These are denoted as

$$h_k = h(x_k), \qquad x_k = k\Delta, \qquad k = 0, 1, 2, 3, ..., N - 1. \qquad (23)$$

With the aim of doing everything simpler and without losing efficiency in our objective, consider that N is even.

With N number of input, it is not possible achieve more than N number of output, so obviously the outputs $H(f_k)$ are associated with discrete frequencies in the range $-f_c$ to f_c. There is freedom to choose in which frequencies the outputs correspond so let us define this frequencies in the following way

$$f_n = \frac{n}{N\Delta}, \qquad n = -\frac{N}{2}, ..., \frac{N}{2}. \qquad (24)$$

When $n = \pm\frac{N}{2}$ then we recover the critical frequency. In addition, here are $N+1$ frequencies, but only N of this are independent.

The next step is to approximate the integral by a discrete sum

$$H(f_n) = \int_{-\infty}^{\infty} h(x)e^{-2\pi i x f_n} dx \approx \sum_{k=0}^{N-1} h_k e^{-2\pi i x_k f_n} \Delta = \Delta \sum_{k=0}^{N-1} h_k e^{-2\pi i k \frac{n}{N}}. \quad (25)$$

In the last step, the eq. (23) and (24) have been used. The final summation in eq. (25) is the DFT.

$$H_n = \sum_{k=0}^{N-1} h_k e^{-2\pi i k \frac{n}{N}}. \quad (26)$$

Now is easy to check that eq. (26) is periodic in n index, with period N. Therefore, an standard method is to modify the index n, which now goes from 0 to $N-1$, this way k and n vary exactly over the same range. When this convention is followed, zero frequency is associated with $n = 0$, positive frequencies $0 < f < f_c$ correspond to index $1 \leq n \leq \frac{N}{2} - 1$, negative frequencies $-f_c < f < 0$ correspond to $\frac{N}{2} - 1 \leq n \leq N - 1$. The value $n = \frac{N}{2}$ is associated with both, $f = -f_c$ ans $f = f_c$.

The same process can be done with the IFT, mathematically the IDFT is described

$$h_k = \frac{1}{N} \sum_{k=0}^{N-1} H_n e^{2\pi i k \frac{n}{N}}. \quad (27)$$

2.3.3 Fast Fourier transform (FFT)

One really important question in computational algorithms is how the computational time needed escalates with N. For the DFT until the mid-1960s, the standard answer was:

$$W = e^{\frac{2\pi i}{N}}, \quad (28)$$

then the DFT can be written as

$$\sum_{k=0}^{N-1} W^{nk} h_k, \quad (29)$$

eq. (29) corresponds to a matrix multiplication, where the matrix (n, k)th element is the constant W up to the power $n \times k$ and h_k samples form a vector of N-dimensions. The result of the matrix multiplication is a vector of also N-dimensions whose components are H_n. If one wants to carry out the matrix product, an order of N^2 multiplications is needed,consequently the DFT appears that the computational time is $\mathcal{O}(N^2)$.

However, during the mid mid-1960s a new algorithm to calculate the DFT

became generally known. This algorithm was called fast Fourier transform or FFT due to its computation time, which is $\mathcal{O}(N \log_2(N))$.

The demonstration of the FFT is tedious and do not bring up anything new relevant for the purposes of this project. For those who want to learn more about the topic, there is a detailed explanation in [2] . Despite this fact, it is important to remark that FFT is more efficient when it is use a number of points $N = 2^k$. For our purpose, this condition can always be satisfied.

To sum up, the philosophy of the FFT is the same as DFT. The only remarkable change is the way it is compute, making it more efficient than DFT. For this reason in our algorithm FFT will be used instead the DFT Even thought the method applied is different the inputs an outputs obtained are exactly the same.

3 Numerical tests

Currently, all the theoretical background for a numerical method to resolve the GPE and Schrodinger equation has been explained. Now it is time to check that the numerical method code works correctly.

3.1 Before coding

The language chosen to develop the code is python and there are several reasons why it was chosen. One of the main reasons is due to the extended use of python in different professional so it is beneficial for our professional formation to get more experience using it. Another reason is the familiarity and experience our tutors Dr. Bruno Juliá and Dr. Carles Calero have with this particular program language, so if anyone of us get stuck with one problem it will be easier for them to help us.

3.2 Familiarazing with FFT

One of the most used functions in the algorithm is FFT, so before starting the real numerical method, it was necessary to get familiarize with FFT functions and try to understand how them worked.

The FFT is performed by an scipy package named Scipy.fft. Nevertheless, the output corresponding to frequency domain follows the standard convection where first the positive frequencies are returned and then the negative ones. In order to recover a better distribution of points, where the first ones to appear corresponds to negative frequencies and the last ones to positive, the function from numpy fft.fftshift is used.

The first test done was sample a sinusoidal function and then the FFT and the shift function were applied. The result is observed in Fig. 3, the only two frequencies that appear in transformed domain are the negative and positive frequency of the sinus. In addition this two frequencies appear at the same distance from the zero frequency, that means the shift function has correctly sort all them.
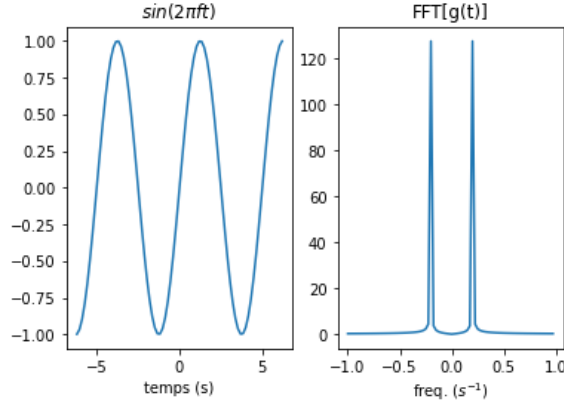

Figure 3: In the left image a sinusoidal function is sampled with $N = 2^8$ points,in the right image FFT is implemented on sinusoidal function.

,

To recover the original function IFFT function was applied. The results are shown in Fig.4, initial function and recovered one are identical.

Another important feature to check was the similarity between the FFT and the FT. For the numerical method what really is necessary is the FT. It is important that the values used in the FFT not differ from the FT. In eq. (25), it is calculated the relationship between FT and FFT ($FT = \Delta FFT$).

For the test, the Gaussian function was used because the FT of a Gaussian it is also a Gaussian, then the analytical solution was compared with the FFT solution. The Gaussian function used was

$$f(x) = e^{-5x^2}. \tag{30}$$

Analytically the FT of the Gaussian is

$$F(k) = \int_{-\infty}^{\infty} e^{-5x^2} e^{-2\pi i x k} dx = \sqrt{\frac{\pi}{5}} e^{\frac{-\pi^2 k^2}{5}}. \tag{31}$$
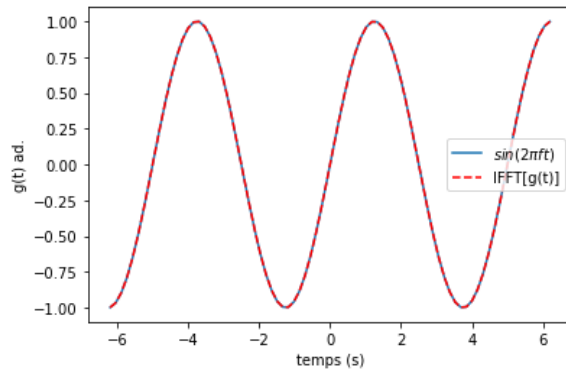
14

Figure 4: The IFFT applied to FFT[f(x)].

,

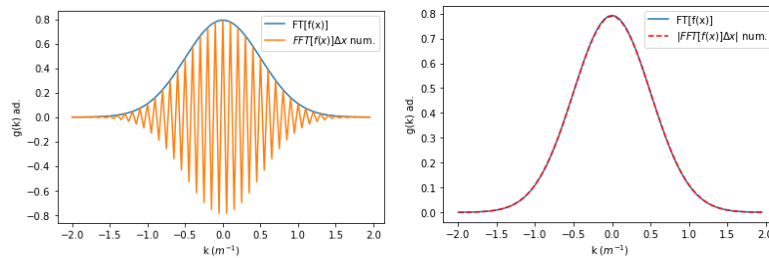The results obtained applying the FFT to $f(x)$ are shown in Fig. 5.

Figure 5: In the left image FFT of a Gaussian function multiplied by Δx and the analytical FT.In the right image the absolute value FFT multiplied by Δx of a Gaussian and the analytical FT.

,

When the FFt is applied the outcome was a Gaussian with values oscillating between negative and positive instead of a Gaussian with only positive real values.

To discard that the result was not caused by the way scipy FFT is programmed, a own-DFT was programmed and both results were compared. It is shown in Fig. 6 that the results were exactly the same in the two cases, so it was not a problem of how works the scipy function but something more deep related to how DFT works.

The next step was to check the definition of the DFT and how was calculated. If one search the most uses of DFT, it can be seen that generally is operated with time and frequency domain. Time in general starts at $t = 0$, so it seemed
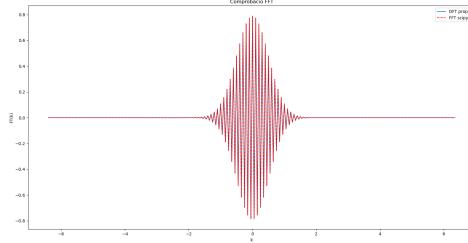
15

Figure 6: My own programmed DFT and FFT from scipy applied to Gaussian function.

,

a good idea to calculate the analytical FT of a Gaussian not centered in $x = 0$, but in $x > 0$. The new initial function was

$$f(x) = e^{-5(x-x_0)^2}. \tag{32}$$

If a change of variable $u = x - x_0$ is done, then the exponential in the FT change to $e^{-2\pi i x k} \rightarrow e^{-2\pi i k(u+x_0)}$. In fact, as x_0 is a constant, a new exponential term can go outside of the integral and the exact same result as eq. (31) is obtaned but with an additional complex phase

$$F(k) = \sqrt{\frac{\pi}{5}} e^{\frac{-\pi^2 k^2}{5}} e^{-2\pi i k x_0}. \tag{33}$$

The apparition of the complex phase gives a hint of what is really happening. Another test that helped to clarify the problem was defining two Gaussian, one centered in $x_0 = 0$ and in the interval x $[-10, 10]$ as $f(x)$. The second Gaussian centered in $x_0 = 10$ and in the interval x $[0, 20]$ as $f'(x - x_0)$.

From the results shown in Fig. 7 , one can conclude that somehow FFT interprets that our function is sampled in the interval [0,20] and the Gaussian is centered in the middle of the interval.

In eq. (23), it was forced that x starts at $x = 0$. In other words, DFT does not know which points corresponds to each x, it assumes that the first point is equivalent to $x = 0$ and the other points are equivalent to $x = \Delta x i$ where $i = 1, 2, ..., N - 1$.

Even thought this seems that we are restricting the DFT formula, there is not any lost of generality, but it is necessary to be more carefully with our sampled function and how they are sorted. Particularly, what was happening is that FFT see the Gaussian centered in $\frac{N}{2} + 1$. Consequently, it interprets that the sampled function corresponds to a Gaussian with the center not in $x = 0$,
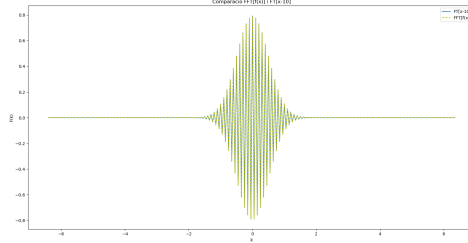
Figure 7: In blue the analytical FT of a Gaussian centered in $x_0 = 10$ and sampled in x [0,20]. In yellow the FFT of a Gaussian centered in $x_0 = 0$ and sampled in x [-10,10].

,

which would be the maximum in the first point, but a Gaussian with center in $x = (\frac{N}{2} + 1)\Delta x$, which corresponds to a Gaussian centered in $x_0 = 10$.

To try avoid this effect, the Gaussian function $f(x)$ was split in two parts, one that samples in the interval x $[-10, 0]$ with N points and another with N points which samples the interval left. This way the maximum of $f(x)$ would correspond to the initial and final points where due to periodicity of DFT is the same that saying our $f(x)$ is centered in $x_0 = 0$. Results can be appreciated in Fig. 8, where now the FFT do not oscillate between negative ans positive real values. However, the sum it is not exactly the same so this was not an optimal solution, but it leads to confirm the hypothesis that there was a problem with how FFT interprets the sampled $f(x)$.

Finally a way to avoid this effect was find. One must sort the sampled $f(x)$ in a way that the first point contains the maximum value $x = 0$, then all points with $x > 0$ and finally all points with $x < 0$ from maximum to minimum $|x|$. This order is exactly the same used in the DFT outputs, performed for numpy.fft.fftshift. In Fig. 9 it is shown the result of applying the DFT into the sampled $f(x)$ sorted by fft.fftshift.

Despite having developed a method to avoid the shift created in the FFT, it was not used. Even thought a shift is created in the k domain, this only adds a relative phase to our wave function. At the end, for calculations the only relevant physics magnitudes are probabilities, which are calculated with the modulus of our complex function. As it can be seen in Fig. 5, the module and it is not affected by the shift. Consequently the final results extract from Split-step method should not depend on shifting or not the sample of $f(x)$.

Once the FFT of the Gaussian is calculated, one has to be able of recovering the initial Gauss function applying the IFFT. Two ways of recovering the
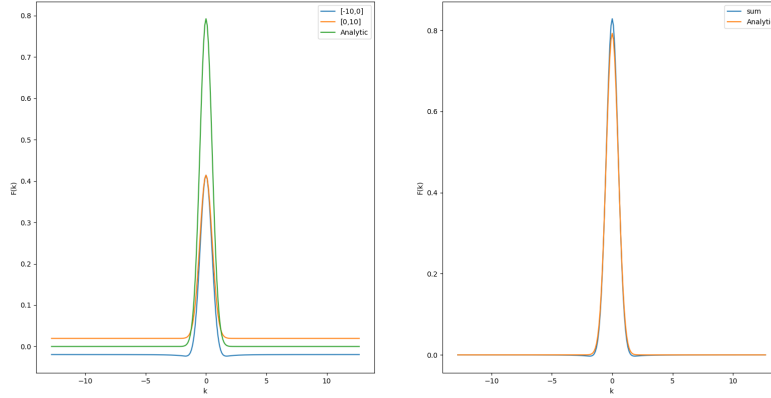
17

Figure 8: Left image shows the analytical FT and FFT of Gaussian centered in $x_0 = 0$ ($f(x)$). FFT split in two samples, the blue one in interval x $[-10, 0]$ and the orange one in interval x $[0, 10]$. Right image shows the sum of the two FTT and the analytical solution of $\mathcal{F}[f(x)]$.
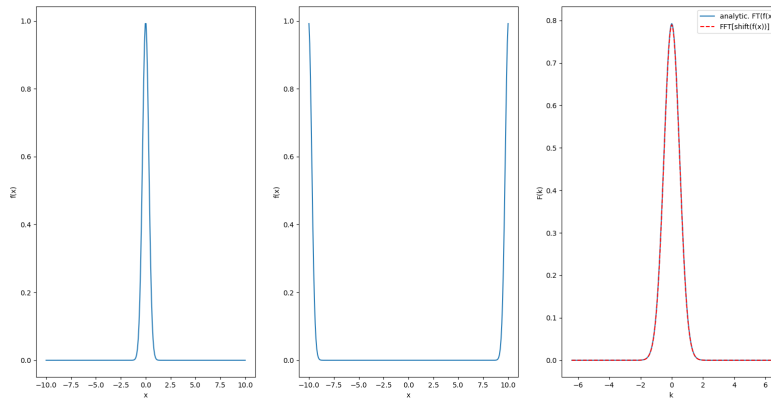
,

Figure 9: First image shows Gaussian centered in $x_0 = 0$ ($f(x)$). Second image shows the result of apply shift function into $f(x)$. Third image shows the FFT of $shift(f(x)$ and the analytical FT of $f(x)$.

,

initial wave function were tried. One was multiplying the IFFT by Δk, the other one was multiplying by $\frac{1}{\Delta x}$.

18

In Fig. 10 the results obtained in both cases are shown. From the results it is directly to see that the correct way to recover the original function is multiplying by $\frac{1}{\Delta x}$. The factor $\frac{1}{\Delta x}$ is needed instead of Δk because in eq. 27 the IFFT is defined with the fraction $\frac{1}{N}$, which multiplied by $\frac{1}{\Delta x}$ returns Δk .
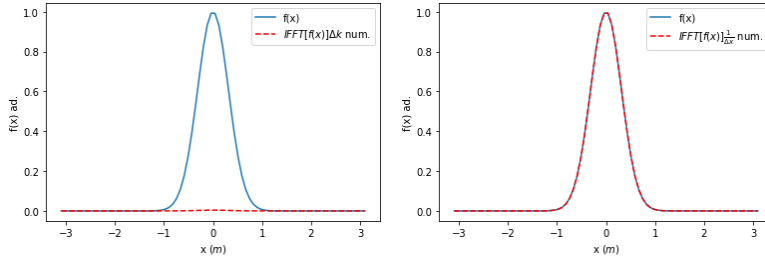

Figure 10: In the left image IFFT multiplied by Δk of a Gaussian function and the initial Gaussian function.In the right image IFFT multiplied by $\frac{1}{\Delta x}$ of a Gaussian function and the initial Gaussian function.

3.3 Split-step method

In this section will be explained how the Split-step program was schemed and all the tests made to prove its reliability.

3.3.1 Program scheme

The Split-step method is applied by a function named split_step. The inputs must be variables that did not change during the program. The inputs chosen for the split-step function were:

- Number of points N used in position and momentum space.
- Space vector x, where our wave function and external potential is defined.
- Distance between position points Δx
- Vector with the initial values of the wave function $\psi(t_0, x)$
- Distance between time points Δt
- Vector with the values of the external potential $V(x)$
- Coupling constant g

19

- Momentum vector k.

- Distance between momentum points Δk

The split-step function take the initial wave function given in the argument, and calculates the final wave function following eq. (17). Each step described in eq. (17) is done separately by internal functions.

The functions named psi_onex and psi_twop calculate the intermediate wave functions, while psi_fx calculate the final wave function in $t + dt$. The functions fourier_trans and inv_fourier_ trans converts FFT and IFFT to FT and IFT.

To calculate the exponential with the momentum p term one has to be carefully because the relation between k domain and p is subtle. From quantum mechanics the relationship between momentum and wave number K_w is

$$p = \hbar K_w. \tag{34}$$

Intuitively, one would think that K_w is the transformed domain, which indeed it is when you use the eq. (20). However the equation used is eq. (18), so appears a 2π factor and K_w is related with k by the relationship

$$K_w = 2\pi k. \tag{35}$$

At the end the relation between k domain and momentum p is

$$p = 2\pi\hbar k. \tag{36}$$

3.3.2   Split-step method study

To test the reliability of the Split-step method programmed, it was applied to solve two well-known problems in quantum mechanics:

- Temporally evolution of free Gaussian packet wave

- Temporally evolution of Harmonic oscillator fundamental estate

In each case the analytical solution is known, so the numerical solutions obtained numerically with the Split-step method can be compared with the theoretical ones.

Free Gaussian packet wave

From quantum mechanics theory one can obtain the expression for a Gaussian wave packet, which saturate the position-moment uncertainty relationship $(\sigma_x \sigma_p = \frac{\hbar}{2})$ at $t_0 = 0$ and centred on $x_0 = 0$. This particular function corresponds to

$$\phi(x,0) = \frac{1}{(2\pi\sigma_x^2)^{\frac{1}{4}}} e^{-\frac{1}{4}\frac{x^2}{\sigma_x^2} + \frac{i}{\hbar}p_0 x}, \tag{37}$$

$$\tilde{\phi}(p,0) = \frac{1}{(2\pi\sigma_p^2)^{\frac{1}{4}}} e^{-\frac{1}{4}\frac{(p-p_0)^2}{\sigma_p^2}}, \tag{38}$$

where $\sigma_p = \frac{\hbar}{2\sigma_x}$.

Even thought this result is correct, it was obtained using eq. (20). In order to be consequent with the IFT definitions used for the program, the eq. (38) has to be multiplied by a factor $\sqrt{2\pi}$. Then the result become

$$\tilde{\phi}(p,0) = \left(\frac{2\pi}{\sigma_p^2}\right)^{\frac{1}{4}} e^{-\frac{1}{4}\frac{(p-p_0)^2}{\sigma_p^2}}. \tag{39}$$

For free force systems, the hamiltonian H only has the kinetic energy term

$$H = \frac{P2}{2m}. \tag{40}$$

Because the hamiltonian only have P operator terms, the momentum wave function is an eigenfunction. Therefore the evolution operator is only a phase and the probability density do not change.

$$\tilde{\phi}(p,t) = e^{-\frac{i}{\hbar}\frac{p^2 t}{2m}} \tilde{\phi}(p,0) \rightarrow |\tilde{\phi}(p,t)|^2 = |\tilde{\phi}(p,0)|^2. \tag{41}$$

On the other hand, in the position space $\phi(x,0)$ is not an eigenfunction. Therefore, the probability density change over time. Even so, one can obtain analytically $\phi(x,t)$ applying the IFT to $\tilde{\phi}(p,t)$

$$\phi(x,t) = \frac{1}{\sqrt{2\pi\hbar}} \int dp\, e^{\frac{i}{\hbar}px} \tilde{\phi}(p,t) = \frac{1}{\sqrt{2\pi\hbar}} \int dp\, e^{\frac{i}{\hbar}(px - \frac{p^2 t}{2m})} \tilde{\phi}(p,0). \tag{42}$$

If one does the proper calculations in eq. (42) , the final expression for $\phi(x,t)$ correspond to

$$\phi(x,t) = \pi^{-\frac{1}{4}} \left(\frac{\sqrt{2}\frac{\sigma_p}{\hbar}}{1 + \frac{i2\sigma_p^2 t}{m\hbar}} \right)^{\frac{1}{2}} e^{\frac{ip_0\frac{x}{\hbar} - (\frac{\sigma_p}{\hbar})^2 x^2 - \frac{ip_0^2 t}{2m\hbar}}{1 + \frac{i2\sigma_p^2 t}{m\hbar}}}, \tag{43}$$

once $\phi(x,t)$ is obtained, the probability density $|\phi(x,t)|^2$ is

$$|\phi(x,t)|^2 = \pi^{-\frac{1}{2}} \frac{\sqrt{2}\frac{\sigma_p}{\hbar}}{(1 + \frac{4\sigma_p^4 t^2}{m^2\hbar^2})^{\frac{1}{2}}} e^{-\frac{2(\frac{\sigma_p}{\hbar})^2 (x - v_g t)^2}{1 + \frac{4\sigma_p^4 t^2}{m^2\hbar^2}}}, \tag{44}$$

where v_g is named velocity group. It is the velocity with which the mean position moves during the time evolution, it can be calculated by

$$v_g = \frac{p_0}{m}. \tag{45}$$

From eq. (44) it is possible to get an expression that relates σ_x and σ_p

$$\sigma_x^2(t) = \frac{1}{4}\frac{\hbar^2}{\sigma_p^2} \left(1 + \frac{4\sigma_p^4 t^2}{m^2\hbar^2} \right) = \sigma_x^2(0) + \frac{\sigma_p^2}{m^2} t^2. \tag{46}$$

One can notice that $\sigma_x^2(t)$ is a parabolic function. Moreover, if $\sigma_x^2(t)$ is plotted in front of $\sigma_p^2 t^2$, the result is a straight line with an slope value of $\frac{1}{m}$.

Another option to obtain this exact same result, could be with the Ehrenfest theorem. Ehrenfest theorem enunciates that expected values of position and momentum operators, respectively X and P, evolves with the classical physics laws, but with mean forces.

In a free force system, where our initial expected values are $< X > (t_0) = 0$ and $< P > (t_0) = p_0$ it is easy to apply the Ehrenfest theorem.

$$\frac{d < P >}{dt}(t) = < F > = 0 \rightarrow < P > (t) = p_0,$$
$$\frac{d < X >}{dt}(t) = \frac{< P >}{m} = \frac{p_0}{m}. \tag{47}$$

In a reference system where $< X > (t_0) = < P > (t_0) = 0$, it is possible to calculate in a straight forward manner the momentum and position dispersion of our system.

The momentum dispersion evolution $\sigma_p^2(t)$ would be

$$\sigma_p^2 =< P^2 > - < P >^2=< P^2 >,$$
$$\frac{d\sigma_p^2}{dt} = \frac{d < P^2 >}{dt} = \frac{1}{i\hbar} < [P^2, H] >= 0. \tag{48}$$

Position dispersion evolution $\sigma_x^2(t)$ is a bit more tedious. First it is calculated the commutator of X^2 and P^2

$$[X^2, P^2] = [X^2, P^2] = P[X^2, P] + [X^2, P]P = 2i\hbar\,(PX + XP)\,, \tag{49}$$

then it is calculated the commutator between the solution obtained in eq. (49) and P^2

$$[(PX + XP)\,, P^2] = XP^3 - P^2XP + PXP^2 - P^3X =$$
$$= [X, P^3] + P[XP - PX]P = [X, P^3] + P[X, P]P. \tag{50}$$

Other important expression is the commutator property

$$[A[A, B]] = 0 \rightarrow [A^n, B] = [A, B]nA^{n-1}. \tag{51}$$

Now using eq. (51) in eq. (50)

$$[X, P^3] + P[X, P]P =$$
$$= -[P^3, X] + P[X, P]P =$$
$$= -[P, X]3P^2 + P[X, P]P = \tag{52}$$
$$i\hbar(3P^2 + P^2) = 4i\hbar P^2.$$

Finally Position dispersion evolution $\sigma_x^2(t)$ can be calculated

$$\sigma_x^2 =< X^2 > - < X >^2=< X^2 >,$$
$$\frac{d^2\sigma_x^2}{dt^2} = \frac{d^2}{dt^2} < X^2 >= \frac{d}{dt}\left(\frac{1}{i\hbar} < [X^2, H] >\right) = \tag{53}$$
$$= \frac{d}{dt}\left(\frac{1}{2i\hbar m} < [X^2, P^2] >\right).$$

Applying eq. (49)

$$\frac{d}{dt}\left(\frac{1}{2i\hbar m} < [X^2, P^2] >\right) = \frac{1}{m}\frac{d}{dt} < PX + XP >=$$
$$= \frac{1}{mi\hbar} < [PX + XP, H] >= \frac{1}{2i\hbar m^2} < [PX + XP, P^2] > . \tag{54}$$

Using eq. (50) and eq. (52)

$$\frac{d^2\sigma_x^2}{dt^2} = \frac{1}{2i\hbar m^2} < [PX + XP, P^2] >=$$
$$= \frac{2}{m^2} < P^2 >= \frac{2}{m^2}\sigma_p^2, \tag{55}$$

as it can be seen in eq. (48), σ_p^2 is constant. Differential equation in eq. (55) can be solved directly, and give as a result a parabolic function

$$\sigma_x^2(t) = \sigma_x^2(0) + \frac{2}{m^2}\sigma_p^2 t^2. \tag{56}$$

In addition, taking into account that at t_0 position-moment uncertainty is saturate, eq. (56) can be rewritten as

$$\sigma_x^2(t) = \sigma_x^2(0) + \frac{\hbar}{4m^2\sigma_x^2(0)}t^2. \tag{57}$$

The units used in the simulation are atomic units. Where $\hbar = 1$, mass is measured by electron mass (m_e), position in Bohr radius (a_0), energy by Hartree energy (E_h) and time in ($\frac{\hbar}{E_h}$).

The parameters used in the simulation for the Gaussian functions are

$$p_0 = 0\,\frac{m_e a_0 E_h}{\hbar},$$
$$\sigma_x^2(0) = 1\,\frac{1}{a_0},$$
$$\sigma_p^2(0) = \frac{1}{2}\,\frac{\hbar}{m_e E_h}, \tag{58}$$
$$m = 1\,m_e.$$

Using the specified parameters, all the relevant expressions calculated became

$$\phi(x,0) = \frac{1}{(2\pi)^{\frac{1}{4}}} e^{-\frac{1}{4}x^2},$$

$$\tilde{\phi}(p,0) = \frac{1}{(\frac{\pi}{2})^{\frac{1}{4}}} e^{-p^2},$$

$$<P> = 0 \ \frac{m_e E_h}{\hbar},$$

$$<X> = 0 \ a_0,$$

$$\sigma_x^2(t) = \left(1 + \frac{1}{4}t^2\right) \frac{1}{a_0^2}.$$

(59)

For the numerical simulation $N = 2^{15}$ points were used to sample $\phi(x,0)$ in the x interval $[-50, 50]\ a_0$. For time evolution, $N_t = 31$ points were used with $dt = 1\ \frac{\hbar}{E_h}$.

As result of none force is exerted to the particle, probability density should became shorter and wide over time. In the numerical simulation this effect is appreciated in Fig. 11.
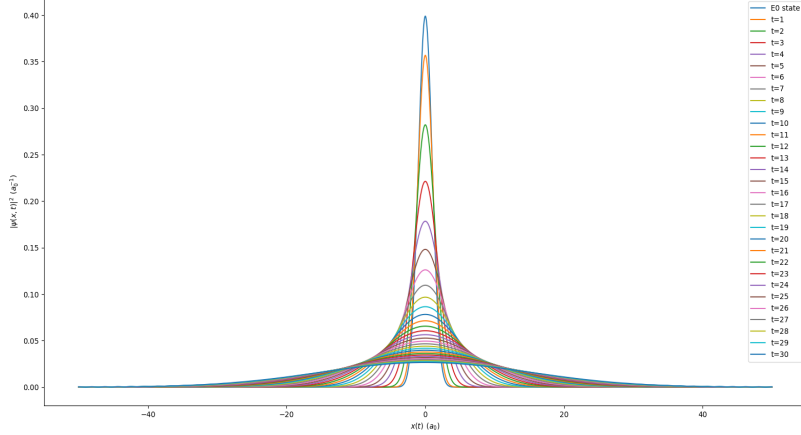

Figure 11: Time evolution numerical simulation of probability density $|\phi(x,t)|^2$.

The comparison between analytical mean values and the obtained in the numerical simulation are shown in Fig. 12. As it can be seen, not only momentum and position values are constant in time as were predict, but also the deviation between numerical and analytical results are smaller than approximately 10^{-12}.
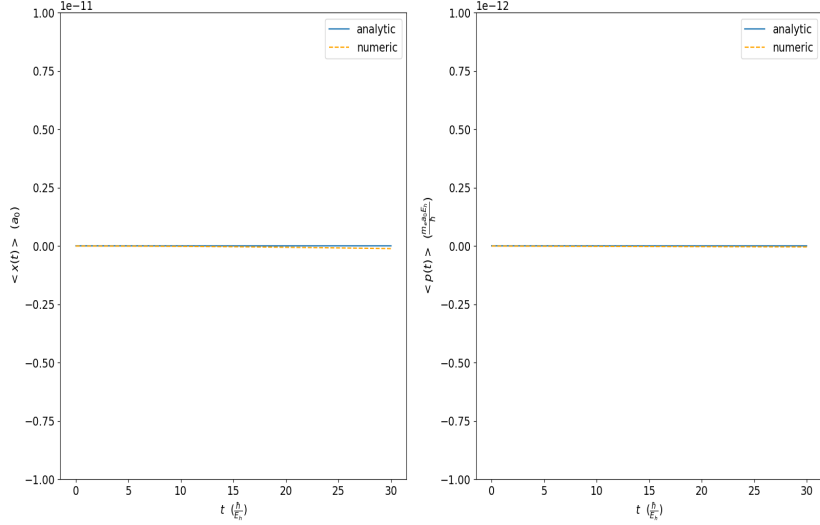
Figure 12: Left image is shown position mean value time evolution calculated analytically and numerical in scale of $10^{-11}a_0$. Right image is shown momentum mean value time evolution calculated analytically and numerical in scale of $10^{-12}\frac{m_e a_0 E_h}{\hbar}$.

Additionally, comparison between analytical and numerical σ_p^2 and σ_x^2 was made. The results are shown in Fig. 13 and Fig. 14. In the three cases, the numerical simulation fits perfectly the analytical solutions from eq. (59).

The first time the comparison was made, $N = 2^8$ were used for the simulation. In this case the momentum dispersion numerically give an error of approximately 0.4%. With the objective to discern if the error was caused due to discretization or due to a bad performance of my Split_step program, it was done a study of error convergence with the number of points used.

For error convergence with spatial number of points N, it was used σ_p^2. The σ_p^2 was chosen because showed the biggest relative error. Only the initial wave function $(t = 0)$ was used to avoid error caused by time discretization. Results of parameter N study are in Table 1. From Fig 15 can be seen that the error fall exponentially with N.

To study parameter N_t, $\sigma_x^2(t_{final})$ was used. The reason why $\sigma_x^2(t_{final})$ was chosen is because it showed the biggest change in time, so it should have the biggest error due to time evolution at the end of the simulation. Results are found in Fig. 16, it shows that error decay exponentially also with N_t.
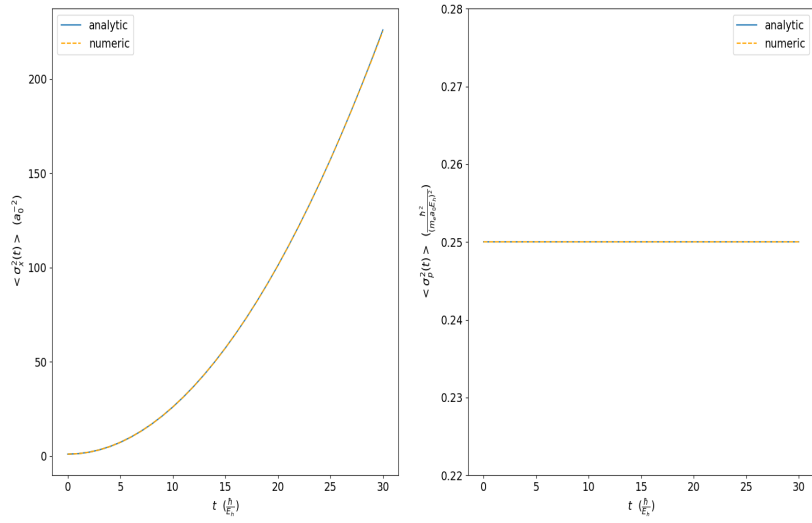
Figure 13: Left image is shown position dispersion value over time calculated analytically and numerical. Right image is shown momentum dispersion value over time calculated analytically and numerical.
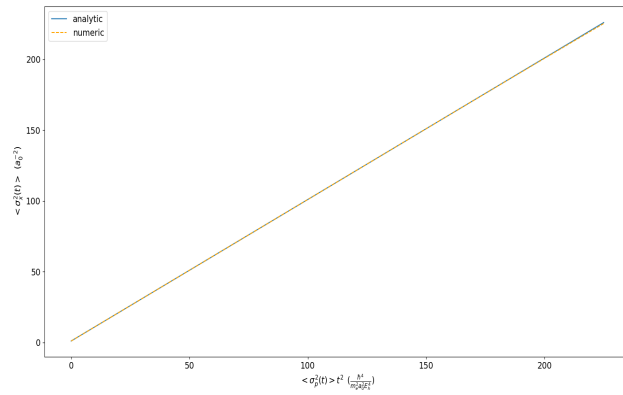

Figure 14: Relation between position dispersion, momentum dispersion and time analytically and numerically.

Study N parameter			
Number of points (N)	Time needed to run the program (t)	Difference $\|\sigma^2_{p\,analy.} - \sigma^2_{p\,num.}\|$	Error σ^2_p in %
2^8	$t \approx 3s$	10^{-3}	0.4%
2^{12}	$t \approx 3s$	$10^{-4.18}$	0.025%
2^{15}	$t \approx 3s$	10^{-5}	0.004%
2^{17}	$t \approx 10s$	$10^{-5.64}$	0.0009%
2^{19}	$t \approx 20s$	$10^{-6.3}$	0.0002%

Table 1: Study of the parameter number of sampled points N. The study focus on the error in momentum dispersion σ^2_p.
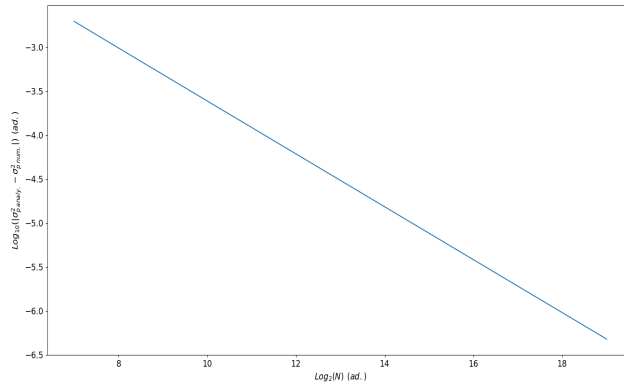

Figure 15: Logarithmic representation of difference between analytical momentum dispersion $\sigma^2_{p\,anally}$ and numerical momentum dispersion $\sigma^2_{p\,num}$ in front of the number of sampled points N.

Harmonic oscillator

The harmonic oscillator is a system that, when displaced from its equilibrium position, experiences a restoring force F proportional to the displacement x. It is one of the most studied systems in physics and have analytical solutions.

The hamiltonian of the particle H is

$$H = \frac{P^2}{2m} + \frac{1}{2}m\omega^2 X^2. \tag{60}$$

The time independent Schrödinger equation for the harmonic oscillator is

$$-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}\psi(x) + \frac{1}{2}m\omega^2 x^2\psi(x) = E\psi(x). \tag{61}$$
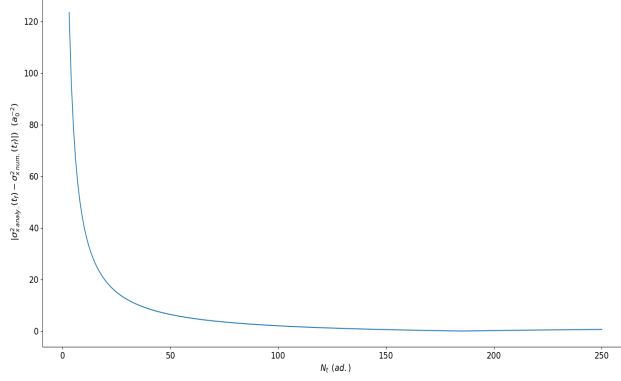
Figure 16: Representation of difference between analytical momentum dispersion $\sigma^2_{x\,anally}(t_f)$ and numerical momentum dispersion $\sigma^2_{x\,num}(t_f)$ in front of the number of time points N_t used in time evolution in a fixed time interval t $[0,30]\frac{\hbar}{E_h}$.

For the numerical simulation, it is useful to reduce the numerical constants and work with a special units named oscillator units. To obtain the equation in the oscillator units, one divide by $\hbar\omega$

$$-\frac{\hbar}{2m\omega}\frac{d^2}{dx^2}\psi(x) + \frac{1}{2}\frac{m\omega}{\hbar}x^2\psi(x) = \frac{E}{\hbar\omega}\psi(x).$$ (62)

Now if the variable are renamed as $\bar{x} = \sqrt{\frac{m\omega}{\hbar}}x$ and $\bar{E} = \frac{E}{\hbar\omega}$, then eq. (62) became dimensionless

$$-\frac{1}{2}\frac{d^2\bar{\psi}}{d\bar{x}^2} + \frac{1}{2}\bar{x}^2\bar{\psi} = \bar{E}\bar{\psi},$$ (63)

the stationary dimensionless solutions of harmonic oscillator are

$$\bar{\psi}_n(\bar{x}) = C_n H_n(\bar{x})e^{-\frac{1}{2}\bar{x}^2},$$ (64)

where H_n are the Hermite polynomials.

For the numerical study, the ground state $n = 0$ will be used. In this case $H_0 = 1$. Substituting in eq. (64), the ground state is described as

$$\bar{\psi}_0(\bar{x}) = \frac{1}{\pi^{\frac{1}{4}}} e^{-\frac{1}{2}\bar{x}^2}. \tag{65}$$

Time evolution is determinate by time dependent Schrödinger equation, if now is divided by $\hbar\omega$

$$\frac{i}{\omega}\frac{\partial}{\partial t}\bar{\psi} = \left[\frac{P^2}{2m\hbar\omega} + \frac{1}{2}\bar{X}\right]\bar{\psi}. \tag{66}$$

From eq. (66) it is possible to define new adimensional variables as $\bar{t} = \omega t$ and $\bar{P} = \frac{P}{\sqrt{m\hbar\omega}} = \frac{d\bar{x}}{d\bar{t}}$. Finally, time evolution equation, energy kinetic operator and potential energy operator in oscillator units are

$$i\frac{\partial}{\partial\bar{t}}\bar{\psi} = \bar{H}\bar{\psi} = [T+V]\bar{\psi},$$
$$\bar{T} = \frac{1}{2}\bar{P}^2, \tag{67}$$
$$\bar{V} = \frac{1}{2}\bar{X}^2.$$

From Ehrenfest theorem, one is able to calculate analytically the expected position and momentum value over time

$$< F > = - < \bar{X} >,$$
$$\frac{d < \bar{P} >}{d\bar{t}} = - < \bar{X} >, \tag{68}$$
$$\frac{d < \bar{X} >}{d\bar{t}} = < \bar{P} > .$$

If one derive the $\bar{X}$ term again, the differential equation can be decoupled

$$\frac{d^2 < \bar{X} >}{d\bar{t}^2} = \frac{d < \bar{P} >}{d\bar{t}} = - < \bar{X} > . \tag{69}$$

The solutions of this differential equation are sinusoidal functions. With a lineal combination of sin and cos functions.Taking in count the initial conditions $< \bar{X} > = x_0$ and $< \bar{P} > (0) = p_0$ the mean values evolution over time are

$$< \bar{X} > = x_0 cos(\bar{t}) + p_0 sin(\bar{t}),$$
$$< \bar{P} > = p_0 cos(\bar{t}) - x_0 sin(\bar{t}). \tag{70}$$

For the numerical simulation $N = 2^{15}$ spatial points were used to sample the initial wave function $\bar{\psi}_0(\bar{x})$ in $\bar{x}\,ad.$ interval $[-10, 10]$ with $d\bar{x} = 0.006\,ad..$ The initial wave function was initially centered in $\bar{x}_0 = 5$. For time evolution $N_t = 100$

points were used in $\bar{t}\,ad.$ interval $[0,6]$ with $d\bar{t} = 0.06\,ad.$

The fundamental state of the harmonic oscillator is a coherent state, so it behave as a classical particle. When it is not centered it starts to oscillate, keeping intact his form during the oscillation, such as a ball attach to a spring. This phenomenon appeared in the numerical simulation, it is shown in Fig. 17.
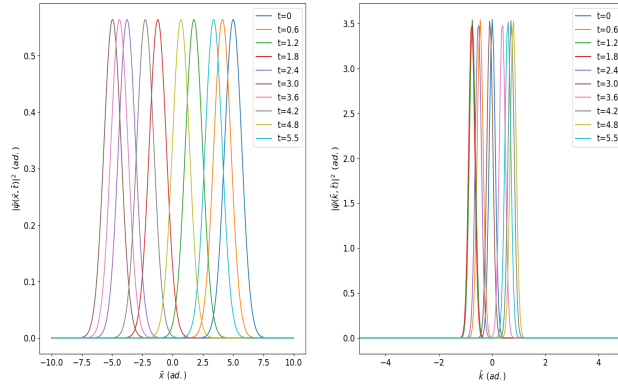

Figure 17: In the left image, numerical simulation of density probability $|\bar{\psi}(\bar{x},\bar{t})|^2$ evolution over time $\bar{t}$. In the rigth image, numerical simulation of density probability $|\bar{\psi}(\bar{k},\bar{t})|^2$ evolution over time $\bar{t}$.

The comparison between the analytical and numerical mean values is shown in Fig. 18, where it can be seen that the numerical results fits with the theoretical ones.

Another test done in the simulation was study the shape change of the wave function during time evolution. It was done calculating width at half maximum or FWHM, which is related with the dispersion by

$$FWHM = 2\sqrt{2Ln(2)}\sigma. \tag{71}$$

To do the study, position and momentum dispersion was calculated over time. If the wave function keeps the shape in time evolution, the dispersion should be constant. The results are shown in Fig. 19. As it was expected the dispersion do not change significantly over time.
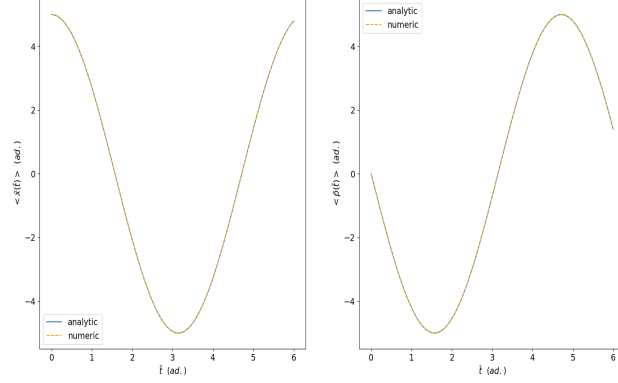
31

Figure 18: Left image is shown position mean over time calculated analytically and numerical. Right image is shown momentum mean over time calculated analytically and numerical.
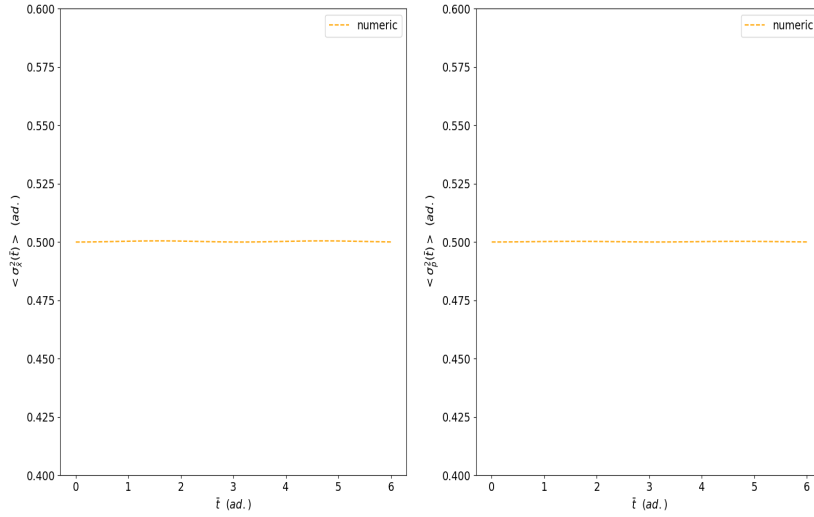

Figure 19: Left image is shown position dispersion value over time calculated analytically and numerical. Right image is shown momentum dispersion value over time calculated analytically and numerical.

3.3.3 Imaginary time evolution study

Besides calculating the time evolution, Split-step method also can find the eigenfunctions of a given Hamiltonian H by replacing time for imaginary time.

Because it will be necessary to solve the GPE equation, it is important to test if the program is able to find the eigenfunctions of different systems.The systems chosen for the test were:

- Infinite well potential

- Harmonic oscillator

Infinite well potential

The time independent Schrödinger equation can be analytically solve, the corresponding wave function to different energy states E_n are

$$
\begin{aligned}
\phi_n(x) &= \frac{1}{\sqrt{a}} cos\left(\frac{n\pi x}{L}\right) && n = 1, 3, 5, ... \\
\phi_n(x) &= \frac{1}{\sqrt{a}} sin\left(\frac{n\pi x}{L}\right) && n = 2, 4, 6, ...
\end{aligned}
\tag{72}
$$

From last equation, $x = \pm a$ are the points where the potential barrier starts and $L = 2a$. Also the discrete energy level E_n analytically are calculated by

$$
E_n = \frac{\hbar^2 \pi^2 n^2}{2mL^2}.
\tag{73}
$$

For the numerical study I used again atomic units and $m = 1\,m_e$.

The aim of this study was to test if the program was able to recover the ground state. Because the ground state have even parity, an initial function with the same parity was chosen. In addition,it has to be a function that tended to zero as it approximates to the barrier position.

One function that fits these characterizes was $\psi_0 = x^2 e^{-2x^2}$. Another one which fulfill the same requirements were $\psi_1 = [x^4 - 5x^2 + 1]cos(x)e^{-4x^2}$. This two functions were sampled with $N = 2^{12}$ points in the x interval $[-10, 10]$ a_0.

To simulate the infinite potential barrier was used

$$
\begin{aligned}
V(x) &= 10000 && |x| > 9.2 \\
V(x) &= 0 && |x| < 9.2
\end{aligned}
\tag{74}
$$

With respect to time parameters, $N_t = 5000$ points were used in the time interval τ $[0, 100]$ $\frac{\hbar}{E_h}$ with $d\tau = i0.01\frac{\hbar}{E_h}$.

Furthermore a study of the energy convergence in each time step was done in parallel. Energy of every step E_n was calculated by

$$E_n = \int_{-\infty}^{\infty} \frac{p^2}{2}|\bar{\psi}(k,t)|^2 dk, \tag{75}$$

then eq. 73 was used to obtain n.

From Fig. 20 and Fig. 21, one can conclude that the method seems to be stable and that in the two cases the ground state ϕ_1 was indeed recovered.
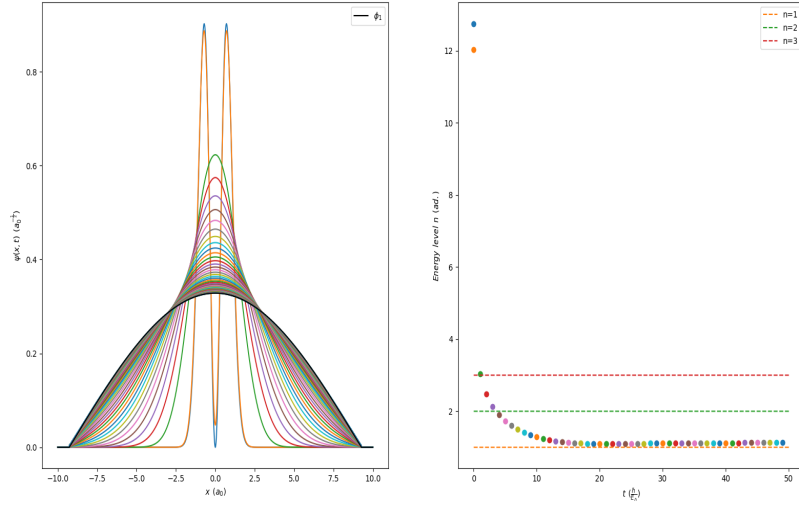

Figure 20: In the left image, imaginary time evolution of ψ_0 until reach the ground state ϕ_0. In the right image energy level n convergence in time.

The next step was to study the relationship between error and time pass $d\tau$. Two different error studies were done. The first one, was the difference between energy state obtained numerically at the end of the simulation $n_{num.}$ and the analytical one $n = 1$. The second one was $< \psi(x, t_f)|\phi_1 >$. The results are shown in Fig. 22.
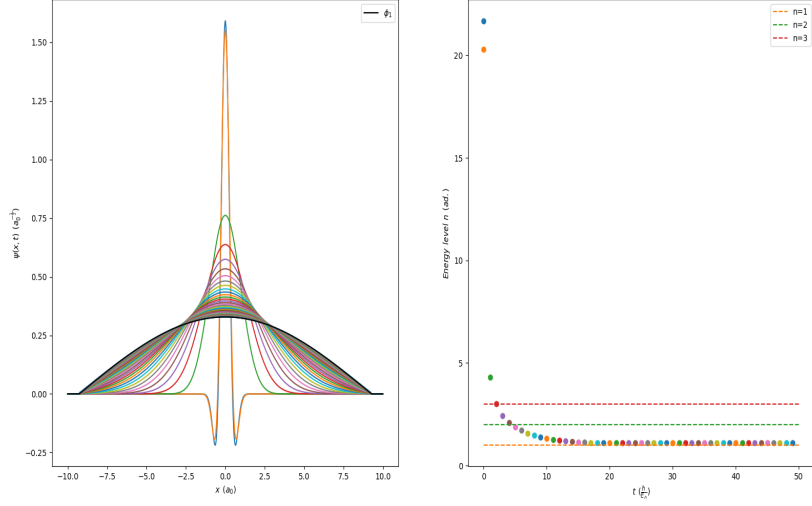
Figure 21: In the left image, imaginary time evolution of ψ_0 until reach the ground state ϕ_1. In the right image energy level n convergence in time.

Harmonic oscillator

In oscillator units, the analytical stationary states are described in eq. (64). When n is even, the Hermite polynomials are even, reversely when n is odd.

To obtain the ground state, the initial function must be even. The function chosen was $\bar{\psi}(\bar{x})_0 = [\bar{x}^4 - 5\bar{x}^2 + 1]cos(\bar{x})e^{-4\bar{x}^2}$. It was sampled with $N = 2^{12}$ points in the x interval $[-10, 10]$ $ad..$

With respect to time parameters, $N_t = 1000$ points were used in the time interval τ $i[0, 10]$ $ad.$ with $d\tau = i0.01$ $ad..$

Equally as infinite well potential, a study of energy convergence in each time step was done in parallel. Energy of every step E_n was calculated by

$$\bar{E}_n = \int_{-\infty}^{\infty} \frac{\bar{p}^2}{2} |\bar{\bar{\psi}}(\bar{k}, \bar{t})|^2 d\bar{k} + \int_{-\infty}^{\infty} \frac{\bar{x}^2}{2} |\bar{\psi}(\bar{x}, \bar{t})|^2 d\bar{x}. \qquad (76)$$

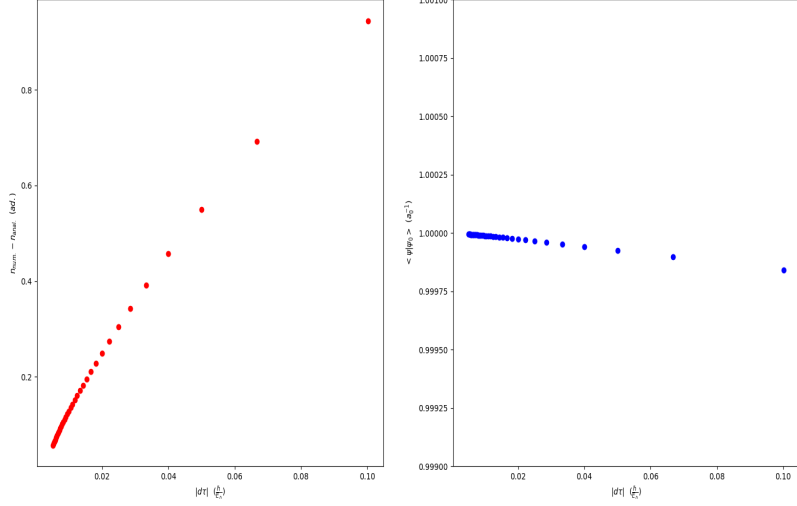For an harmonic oscillator stationary states, the corresponding analytical en-

Figure 22: In the left image, convergence to energy level n with time pass $d\tau$. In the right, convergence of scalar product between simulated ground state and analytical ground state $<\psi(x, t_f)|\phi_1>$ with time pass $d\tau$.

ergy in oscillator units are

$$\bar{E}_n = n + \frac{1}{2}. \tag{77}$$

Combining eq. (76) and eq. (77), it is possible to calculate the state n.

From the results obtained in Fig. 23, the method seems to be stable and recover without problems the ground state ϕ_0.

The next step was to try to get the following energy state possible, which due to parity of our initial function had to be $n = 2$. To obtain the first excited state, the wave function $\bar{\psi}$ was orthogonalized with ϕ_0

$$\bar{\psi}' = \sum_i C_i\phi_i - C_0\phi_0,$$

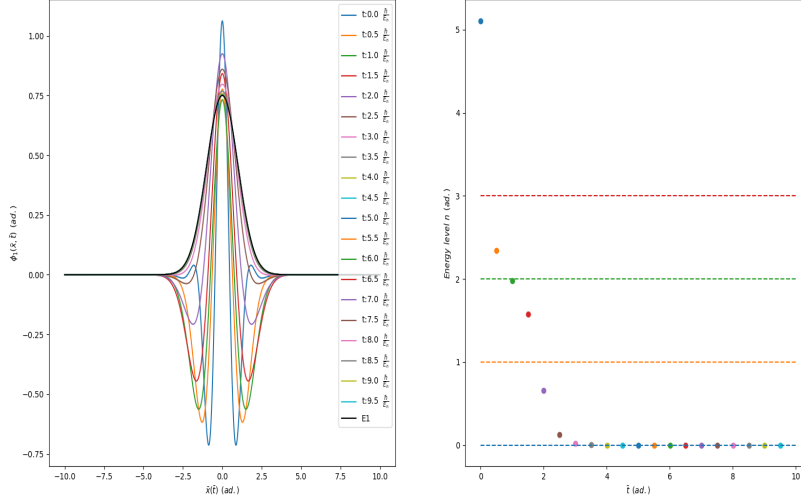$$C_0 = \int_{-\infty}^{\infty} \phi_0^*(\bar{x})\bar{\psi}(\bar{x})d\bar{x}. \tag{78}$$

36

Figure 23: In the left image, imaginary time evolution of $\bar{\psi}$ until reach the ground state ϕ_1. In the right image energy level n convergence in time.

However, this method was not good enough because in every calculation a numerical error is introduced. As result the function which at the beginning was orthogonal to ϕ_0, after some iterations ceases to be orthogonal and eventually evolve to the ground state. This particular effect can be appreciated in Fig. 24.

There are two possible options to avoid this problem. The quicker and less computational expensive is use a small time interval, although sometimes a small time interval do not produce a result satisfactory enough. If it is the case, the other solution is to repeat eq. (78) every time step. This option ensure that in every time step the function do not have any component of our ground state so always evolve to the next low energy level.

The next study was how converge the error with time pass. Time $\bar{t}$ interval used was $[0, 10]$, then the total number of points N_t was being changed. The results are shown in Fig. 25. They seemed to point that split-step converge to the ground state really fast, however in the energy value more error is accumulate and small $d\tau$ is needed to obtain satisfactory results.

To end the study, it was checked that the solutions obtained are truly stationary states, which consequently do not evolve in time. The results are shown in Fig. 26, which conclude that the numerical solutions indeed are stationary.
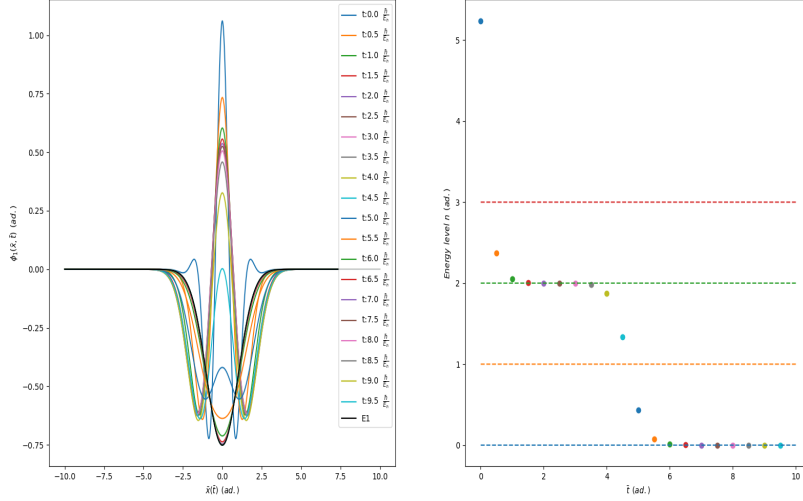
Figure 24: In the left image, imaginary time evolution of $\bar{\psi}$ until reach the ground state ϕ_1. In the right image energy level n convergence in time.

3.4 Ground state of GPE

To obtain the ground state of the GPE with an harmonic potential one have to rely on the imaginary time evolution method.

Even though one can not compare with analytical solution, the Thomas Fermi approximation can be used to find analytical results to study the reliability of the solution obtained.

1D Thomas Fermi approximation

The Thomas Fermi approximation is used when the interaction term is repulsive and large $g >> 0$. In this case, the kinetic term can be negligible in front of the harmonic and interaction terms.

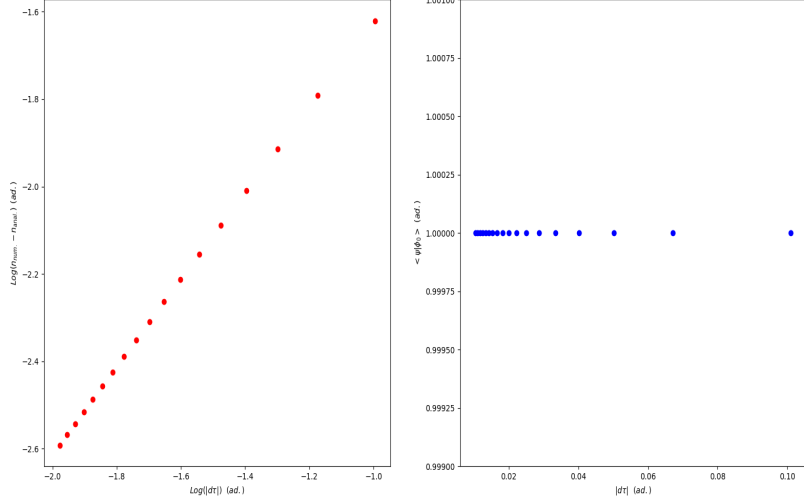The first equation from eq. (5) using this approximation is rewritten as

Figure 25: In the left image, convergence to energy level n with time pass $d\tau$ in logarithmic scale. In the right, convergence of scalar product between simulated ground state and analytical ground state $<\psi(x, t_f)|\phi_1>$ with time pass $d\tau$.

$$\frac{1}{2}\bar{x}^2\bar{\varphi}(\bar{x}) + \bar{g}|\bar{\varphi}(\bar{x})|^2\bar{\varphi}(\bar{x}) = \bar{\mu}\bar{\varphi}(\bar{x}), \tag{79}$$

from the last equation one can obtain directly

$$|\bar{\varphi}(\bar{x})|^2 = \frac{1}{\bar{g}}\left(\bar{\mu} - \frac{1}{2}\bar{x}^2\right). \tag{80}$$

The Thomas−Fermi radius is defined as $\bar{x}$ where the density of the wave function is zero, which it means that $|\bar{\varphi}(\bar{x})|^2 = 0$. In 1D case, the Thomas−Fermi radius expression is

$$R_{TF} = \sqrt{2\bar{\mu}}, \tag{81}$$

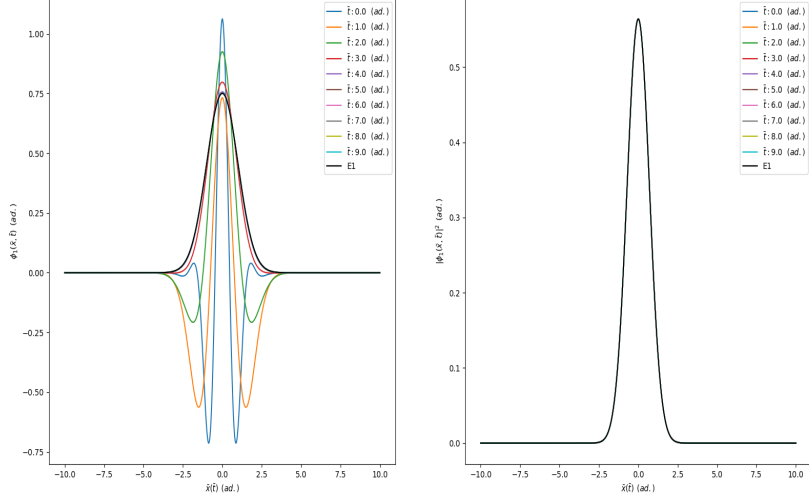in addition, one can have an extra condition imposing the normalization of the

39

Figure 26: imaginary time evolution of $\bar{\psi}$ until reach the ground state ϕ_1. In the right image density probability $|\bar{\psi}|^2$ evolution in time.

probability density

$$\int_{-R_{TF}}^{R_{TF}} |\bar{\varphi}(\bar{x})|^2 d\bar{x} = \int_{-R_{TF}}^{R_{TF}} \frac{1}{\bar{g}} \left(\bar{\mu} - \frac{1}{2}\bar{x}^2 \right) d\bar{x} = 1. \tag{82}$$

Applying eq. (81) into eq. (82) and integrating, the relationship between Thomas−Fermi radius and the interaction term is obtained

$$\begin{aligned} R_{TF}^3 &= \frac{3}{2}\bar{g}, \\ R_{TF} &\, \alpha \, \bar{g}^{\frac{1}{3}}. \end{aligned} \tag{83}$$

In Fig. 27 it is observable that numerical results fits perfectly with analytical results when $g >> 0$. Consequently, the ground state seems to be computed correctly.
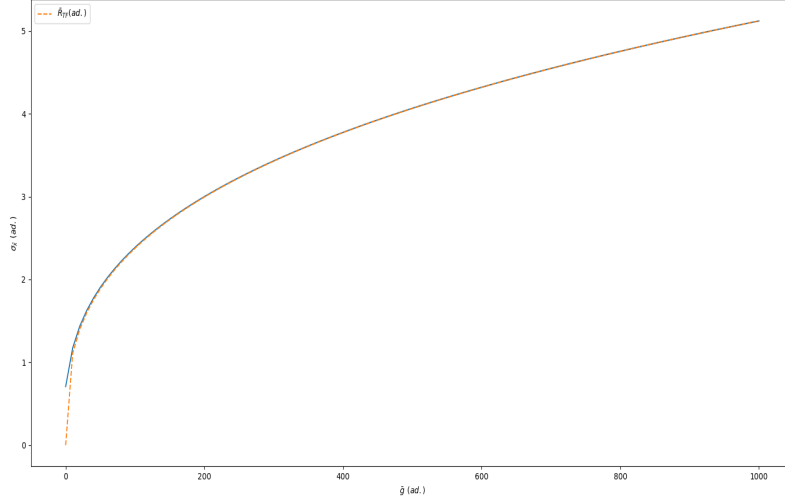
Figure 27: Amplitude of ground state $\sigma_{\bar{x}}$ in front of the interaction term $\bar{g}$ calculated numerically and analytically.

3.5 Dark soliton dynamics

Finally the program is ready to simulate the dynamics of the dark soliton.

To create the dark soliton it is necessary to create a node (lack of density) in the ground state, one possibility to achieve it is multiplying the function by $x - x_0$ or $tangh(x - x_0)$, where x_0 is the node position.

Then, the resulting function is evolved with imaginary time. Even though the imaginary time should return our function back to the ground state, due to orthogonality between our present function and the ground state, the imaginary time evolution pass trough a meta stable state which correspond to the dark soliton solution. Taking advantage of this, if the time evolution is short enough the evolution stops in the meta stable state.

In the program the $tangh(x - x_0)$ is used because it showed the superior efficiency to find the meta stable state.

To check the veracity of the simulation is useful to compare the oscillation frequency of the dark soliton obtained numerically with the theoretical one, which is calculated analytically in [4] for a soliton trapped in an harmonic potential. The analytical value is

41

$$w_{sol} = \frac{w_{har}}{\sqrt{2}},$$
$$\bar{w}_{sol} = \frac{1}{\sqrt{2}}. \tag{84}$$

In first equation w_{sol} and w_{har} are the frequency oscillation of the soliton and the harmonic trap respectively. In the second expression $\bar{w}_{sol}$ is the frequency oscillation of the soliton in oscillator units.

The method used to calculate the numerical soliton frequencies is plotting the time evolution of the condensate density in a 2-D image, where the y-axis is the time, the x-axis is the position and the density is showed with a color bar. In Fig. 28 an example of the plot used is shown.

Once the image is obtained the period of oscillation T can be obtained directly. In Table 2. the numerical w_{sol} calculate with different parameters of the simulation is shown. As it was expected from eq. (84), all the values are around 0.7 approximately, the difference between values can be attributed to the imprecision of the method used which relies in a graphical study instead of a numerical one.
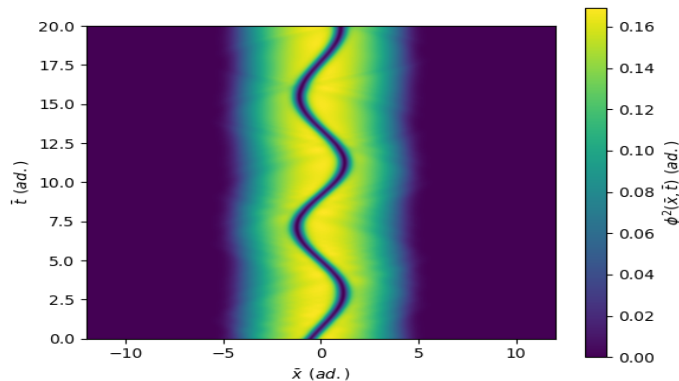

Figure 28: Graphical study of the condensate density $(\phi^2(\bar{x}, \bar{t}))$ with respect time $(\bar{t})$ and position $(\bar{x})$. The parameters used in this simulation were $\bar{(g)} = 70$ and $x_0 = -1$

Study soliton frequency w_{sol}		
Coupling constant $(\bar{g})$	Node initial position (x_0)	soliton frequency (w_{sol})
400	0.5	0.70
400	5	0.70
300	5	0.74
300	1	0.71
50	1	0.73

Table 2: Study of the dark soliton frequency oscillation w_{sol} with respect the change of the numerical parameters used in the simulation.

4   Application

The final objective of this project was to create an application which let the user explore the dynamics and behavior of the dark soliton.

At this point not only the numerical method which the application relies to do all the calculations needed has been deeply tested, but also the program has proved to simulate correct not only one dark soliton but more than one in the same condensate.

However, all this tests and studies has occupied the most part of my time in the internship. Consequently I decided to focus on creating a simple functional application which could be used as a base to someone who will be willing to continue the project in the future.

4.1   Application objective

The application has two main purpose. The first one is to show graphically how the GPE is solved, the second one is to let the user explore the special behaviour of the dark soliton when it interacts between them.

To fulfil the first one, the application displays how the GPE ground state is found, how the dark soliton is introduced and finally the time evolution.

To let the user interact and explore the interaction dynamics, the application allows the user to decide not only the number of solitons that will be introduced, but also where they want to put them in the condensate.

4.2 Application structure

The application is programmed with python, using Kivy. Kivy is a multi-platform application development framework for Python.

The first screen corresponds to solve the ground state of GPE. Here the user can see how starting with an initial Gaussian function and applying the imaginary time evolution, the wave function starts to change until reach a stationary state, which corresponds to the ground state.
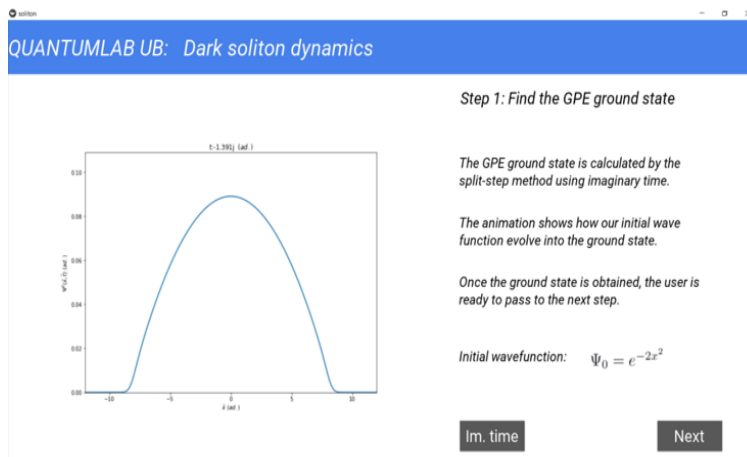

Figure 29: Application screenshot. It is shown the ground state calculated after doing the imaginary time evolution.

The next screen is only to decide how many solitons the user want to introduce in the condensate. When a number is selected, the user is moved forward to the next screen.

After selecting the number of solitons the user has arrived to the second step. In this screen the user first decide where he want to put the solitons, then the imaginary time evolution is displayed to arrive at a stable state. At this point the user can change again the solitons position or can observe the time evolution of the condensate.

Once the user want to change the number of solitons, he can go back to the last screen and repeat all the process.

Further acknowledgment of how the application is and works can be found in the annex.
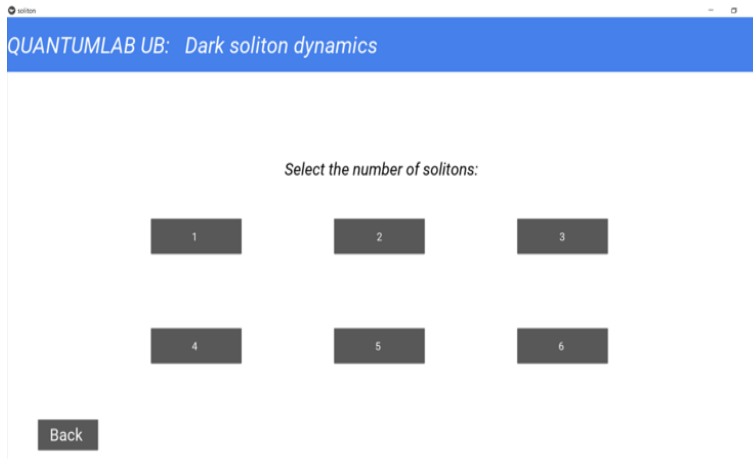
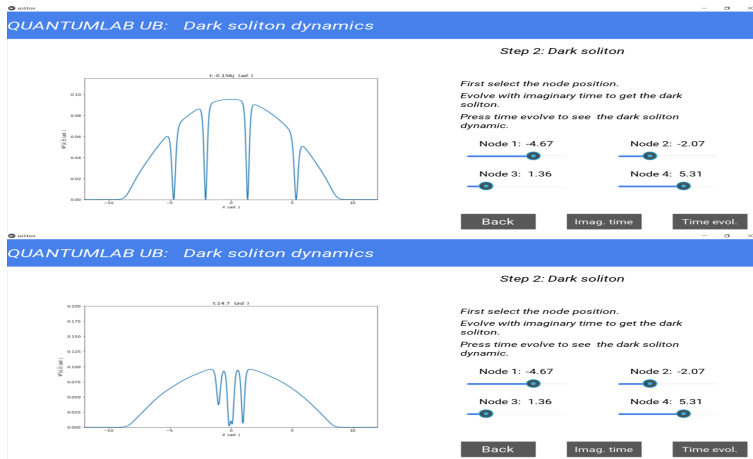Figure 30: Application screenshot. The screen let the user decide how many solitons wants to introduce.

Figure 31: Application screenshot. The first screen shows the wave function with the dark solitons introduced after doing imaginary time. The second screen shows one frame of the time evolution animation.

4.3 Further implementations

Even thought the application is functional, it is far from complete. In this section will be pointed some improvements and features, which could not be done due to lack of time, hopping it will be helpful to the next one who continue the project in the future.

45

Improvements

The most simple and direct improvement which can be implemented is code optimization using the parent-child features of python classes. This could avoid repeating some functions common in different screens and sharing variables which are calculated in different screens.

Other optimization could be trying to avoid "for in range" in the kivy application. When the "for in range" is called, kivy stops and wait until all the loop is done. Consequently the flow application is really slowed down.For this reason it will be better to find a way of computing time evolution loop with only kivy functions.

New features

Finally, there are a few features which will be interesting to add.

One is the comparison between the harmonic and the soliton frequency, for example an animation where both evolve in time could be shown. Following the same line but being a more challenging feature, an animation comparing the interaction between classical particle oscillator and dark solitons could be done.

Another one to make more visually understandable what is happening, could be add the 2-D image of condensate density with respect time and space could be added. An example of this image is Fig. 28, where the user could see better that the dark soliton is a lack of density, which oscillates with time as a sinusoidal function.

5   General conclusions

At the end of the project, all the goals set at the beginning were accomplished. Not only all the numerical tests fitted perfectly with the analytical solutions, as it is detailed in section 3, but also the application was completed and help the user to understand how the GPE equation is solved and how the dark solitons behave.

However, the application has not reached his fully potential. As it is pointed in section 4, the code can be modified to fit in better with kivy structure and more features could be implemented. Nonetheless, it is still functional and what it is more important, it will serve as a good base for the next one to finish the project.

6   Experience and personal conclusions

This project was a thrilling and fascinating experience. It was the first contact I had with a physics project that was not related with the academic world.

At the beginning I was frightened of not being able of prosper with my tasks because I did not know anything about the topic of the project. Nevertheless, I succeed in all the tasks and adversities faced, and to be honest there were not few. All this make me more confident for my future academic and professional career and I am really grateful for it.

To end, I had the opportunity to learn from two great professionals, my tutor Dr. Bruno Juliá and Dr. Carles Calero, who helped me with every reunion to set my next objective and to advise me in my studies. I really thank both of them for this experience, to all the help and mentoring they did I am very grateful. Also to mention that I really enjoyed to share this experience with my fellow student Eric Vidal, who also has supported me in the whole project.

References

[1]   S. Burger et al. "Dark Solitons in Bose-Einstein Condensates". In: *Phys. Rev. Lett.* 83 (25 Dec. 1999), pp. 5198–5201. DOI: 10.1103/PhysRevLett.83.5198. URL: https://link.aps.org/doi/10.1103/PhysRevLett.83.5198.

[2]   In Fortran et al. "Numerical Recipes". In: *Cambridge, UK, Cambridge University Press* (Jan. 1992).

[3]   Juha Javanainen and Janne Ruostekoski. "Symbolic calculation in development of algorithms: split-step methods for the Gross–Pitaevskii equation". In: *Journal of Physics A: Mathematical and General* 39.12 (Mar. 2006), p. L179. DOI: 10.1088/0305-4470/39/12/L02. URL: https://dx.doi.org/10.1088/0305-4470/39/12/L02.

[4]   Carlos Bravo Prieto. "Dark solitary waves in 1D Bose-Einstein condensates". In: 2016.

[5]   Laura Malo Roset. "Bright and dark solitons in a Bose-Einstein condensate". In: 2018.