University of Trento

Department of Physics

Degree course in physics

Final Thesis

Time Evolution of Bose-Einstein Condensates in one dimension: a Numerical approach

Author:
Fabiano Lever

Supervisor:
Dr. Giovanni Garberoglio

September 2013

"Physics is like sex: sure, it may have some practical results, but that's not why we do it."

Richard P. Feynman

Acknowledgements

The acknowledgements and the people to thank go here, don't forget to include your project advisor...

Contents

Dedicated to my dad

Chapter 1

Introduction

The aim of this work is to study the dynamics of Bose-Einsten Condensates in one dimension using numerical methods, comparing the results with the evolution of a normal wave packet. The condensate dynamics is well described by the Gross-Pitaevskii equation, while the standard packet follows the ordinary Schrödinger equation.
The comparison is done by numerically integrating in time both the Gross-Pitaevskii and the linear Schrödinger equations, using an integration engine which has been developed in the form of a C++ library. The integration algorithm is based on a operator splitting tecnique, as shown in Chapter 2, and has been tested reproducing the temporal evolution of a bright soliton.
In Chapter 3 are presented the results of the soliton simulation and the comparison between the transmission coefficent through a rectangular barrier of a condensate and a normal wave packet in different energy regimes.

1.1 BEC and the Gross-Pitaevskii equation

At temperatures close to absolute zero, a dilute gas of bosons can condense in a state of matter called a Bose-Einstein Condensate. This phenomenon is observed when a large number of particles occupy the same quantum state, making quantum effects apparent in a macroscopic scale.
The existence of this phase of matter was theorized by Bose and Einstein in 1924 and is at the base of the superfluid properties shown by liquid ^4He below $2.17K^\circ$; Bose-Einstein condensation of a gaseous sample was first observed in 1995 with experiments on vapors of rubidium (Anderson et al.) and sodium (Davis et al.).

The assumptions made in this work (dilution of the system, as stated in the following paragraph) will restrict the applicability of the results to gaseous condensates.

A gas of N interacting boson is described by the following Hamiltonian:

$$\hat{H} = \sum_{i=0}^{N} \left[-\frac{\hbar^2}{2m}\nabla_i^2 + V(\mathbf{r}_i) \right] + \sum_{i=0}^{N} \sum_{j=0}^{i-1} V_{\text{int}}(|\mathbf{r}_i - \mathbf{r}_j|) \tag{1.1}$$

with the ulterior requirement that the total wavefunction $\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)$ is symmetric upon particle's swap.
We now assume that all the bosons are in the same single-particle state $\psi(\mathbf{r})$ (a safe assumption if the temperature of the system is very low, since all particles will be in the ground state), and that the gas is diluted enough to permit a mean-field approximation of the many-body potential (i.e. $\rho\sigma^3 \ll 1$, with ρ being the particle density and σ the characteristic length of the two-body interaction potential). Approximating the interaction potential with $V_{\text{int}}(|\mathbf{r}_i - \mathbf{r}_j|) = a_s\delta(|\mathbf{r}_i - \mathbf{r}_j|)$, and using the above assumptions it can be shown [1] that the single-particle wave function satisfies the following non-linear equation:

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[-\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r}) + \frac{4\pi\hbar^2 a_s N}{m}|\Psi(\mathbf{r},t)|^2 \right] \Psi(\mathbf{r},t), \tag{1.2}$$

which is known as the time dependent Gross-Pitaevskii equation, where a_s is the boson-boson scattering length, N is the total number of particles and $\int |\Psi(\mathbf{r},t)|^2 d\mathbf{r} = 1$*.
This equation describes the time evolution of a Bose-Einstein condensate in tree dimensions, and its one-dimensional approximation will be numerically integrated to study the behaviour of a quasi-1D BEC.

1.2 Dimension Reduction

In order to consider only one dimension of motion (say z), some assumptions are required on the form of the wave function in the other two dimensions (in this case x and y), since equation (1.2) is not easily separable in tree one-dimensional independent equation -as usually done with the normal Schrödinger's equation- because of the presence of the

*Some authors like to put $\int |\Psi(\mathbf{r},t)|^2 d\mathbf{r} = N$, using $-\frac{\hbar^2}{2m}\nabla^2 + V_{ext} + \frac{4\pi\hbar^2 a_s}{m}|\psi(\mathbf{r})|^2$ as Hamiltonian.

non linear term. Thus, it should be noted that the following results are to be considered as the 1D limit of a 3D mean-field theory, and not as a genuine 1D theory.

We assume that the condensate is under the action of two potentials: an anisotropic trapping potential $V_{\text{trap}}(x, y, z)$ and an external potential which depends only on the direction of motion $V_{\text{ext}}(z)$:

$$V(\mathbf{r}) = V_{\text{trap}}(x, y, z) + V_{\text{ext}}(z), \tag{1.3}$$

Following [2], we start by assuming that the trapping potential has the form:

$$V_{\text{trap}}(x, y, z) = \frac{1}{2} m(\omega_x^2, \omega_y^2, \omega_z^2) \cdot (x^2, y^2, z^2) \tag{1.4}$$

with $\omega_z \ll \omega_x = \omega_y =: \omega_r$; these are the conditions usually set to obtain the so-called cigar-shaped condensate.
We seek solutions to (1.2) in the form $\Psi(\mathbf{r}, t) = \varphi(z, t)\Phi(r, t)$ where $r^2 := x^2 + y^2$, $\Phi(r, t) = \phi(r)e^{-iEt}$ and $\phi(r)$ is a solution of:

$$-\frac{\hbar^2}{2m}\nabla_r^2\phi(r) + \frac{1}{2}m\omega_r^2 r^2\phi(r) = E\phi(r) \tag{1.5}$$

Observing that the trapping frequencies in x and y direction are high (i.e. $\omega_z \ll \omega_r$), we can safely assume $\phi(r)$ to be the ground state of (1.5); we have therefore:

$$\Psi(\mathbf{r}, t) = \varphi(z, t)\frac{1}{a_r\sqrt{\pi}}\exp\left[-\frac{r^2}{2a_r^2}\right]e^{-iEt} \tag{1.6}$$

Where $a_r = \sqrt{\frac{\hbar}{m\omega_r}}$ is the typical lenght of the transverse oscillator.
Plugging this expression in equation (1.2), and averaging in the r direction (i.e. multiplying by Φ^* and integrating over r), we get:

$$i\hbar\frac{\partial}{\partial t}\varphi(z, t) = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial z^2} + \frac{1}{2}m\omega_z^2 z^2 + V_{\text{ext}}(z) + 2\hbar\omega_r a_s N|\varphi(z, t)|^2\right]\varphi(z, t) \tag{1.7}$$

Assuming $\omega_z \simeq 0$ (i.e. weak confinement in the z direction) we can neglect the term $\frac{1}{2}m\omega_z^2 z^2$, and rewrite (1.7) as:

$$i\hbar\frac{\partial}{\partial t}\varphi(z,t) = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial z^2} + V_{\text{ext}}(z) + 2\hbar\omega_r a_s N|\varphi(z,t)|^2\right]\varphi(z,t) \tag{1.8}$$

which is known as the quasi-1D time dependent Gross-Pitaevskii equation.

1.3 Note on Units

In this work, all plots and numerical results have been obtained using a set of units such that $\hbar = m = 1$; units are therefore not reported since not meaningful.
The only relevant parameter remaining is $g = 2\hbar\omega_r a_s N$, which indicates the overall nonlinearity of the system.

Chapter 2

Numerical Methods

In order to compute numerically the time evolution of a physical system, some approximation are to be made. First of all, the space domain has to be discretized and reduced to a finite number of points, limiting both the spatial extension of the simulation and the spatial resolution of the results. Likewise, the temporal domain is discretized, so that the state of the system is known only at separate points in time.
These inevitable approximations introduce errors in the computation that can only be diminished by reducing the spatial and temporal intervals between sampled points, at the cost of computation time and hardware resources.

2.1 Theoretical Background

To integrate (1.8) in time, we first write the time evolution of the system as (note that $\hat{H}$ depends on time implicitly since it depends on $\varphi(z,t)$):

$$\varphi(z,t+\Delta t) = \exp\left(-\frac{i}{\hbar}\int_t^{t+\Delta t} \hat{H}(z,\varphi(z,t))\mathrm{d}t\right)\varphi(z,t) \tag{2.1}$$

Where the hamiltonian is:

$$\hat{H}(z,\varphi(z,t)) = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial z^2} + V_{\text{eff}}(z,\varphi(z,t)) \tag{2.2}$$

having included both the external potential and the nonlinear term of (1.8) in the effective potential V_{eff}.
Approximating the integral with a finite sum, we can rewrite (2.1) as:

$$\varphi(z, t + \Delta t) = \exp\left(-\frac{i}{\hbar} \sum_{n=0}^{N} \hat{H}(z, \varphi(z, t_n))\delta t \right) \varphi(z, t) \tag{2.3}$$

where $\delta t = \Delta t/N$ and $t_n = \delta t\ n$. The time evolution operator becomes:

$$\exp\left(-\frac{i}{\hbar} \sum_{n=0}^{N} \hat{H}(z, \varphi(z, t_n))\delta t \right) = \prod_{n=0}^{N} \exp\left(-\frac{i}{\hbar} \hat{H}(z, \varphi(z, t_n))\delta t \right) \tag{2.4}$$

Our aim is now to split every term of the time evolution operator in two parts, and then compute separately each part. Recalling the Lie-Trotter product formula:

$$\exp\left(\hat{A} + \hat{B}\right) = \lim_{N \leftarrow \infty} \left[\exp\left(\frac{\hat{A}}{N}\right) \exp\left(\frac{\hat{B}}{N}\right) \right]^N, \tag{2.5}$$

and noting that δt is small we get:

$$\exp\left(-\frac{i}{\hbar} \hat{H}(z, \varphi(z, t_n))\delta t \right) = \exp\left(\frac{i\hbar}{2m} \frac{\partial^2}{\partial z^2} \frac{\delta t}{2} \right) \exp\left(-\frac{i}{\hbar} V_{\text{eff}}(z, \varphi(z, t_n)) \frac{\delta t}{2} \right) =: \hat{U}_n \tag{2.6}$$

Thus, it is possible to obtain the final state by computing separately the kinetic evolution and the potential term repeatedly:

$$\varphi(z, t + \Delta t) = \left[\prod_{n=0}^{N} \hat{U}_n \right] \varphi(z, t) \tag{2.7}$$

For the kinetic term, we can write:

$$e^{\frac{i\hbar}{2m}\frac{\partial^2}{\partial z^2}\delta t} \varphi(z, t) = e^{\frac{i\hbar}{2m}\frac{\partial^2}{\partial z^2}\delta t} \int_{-\infty}^{\infty} \hat{\varphi}(k, t) e^{ikz} dk \tag{2.8}$$

where $\hat{\varphi}(k, t)$ is the Fourier transform of the wave function.
Bringing the exponential in the LHS of (2.8) under the integral and recalling that:

$$\exp\left(\frac{i\hbar}{2m}\frac{\partial^2}{\partial z^2}\delta t\right) = 1 + \frac{i\hbar}{2m}\frac{\partial^2}{\partial z^2}\delta t + \frac{1}{2}\frac{-\hbar^2}{4m^2}\frac{\partial^4}{\partial z^4}\delta t^2 + \dots \tag{2.9}$$

we can rewrite 2.8 as:

$$e^{\frac{i\hbar}{2m}\frac{\partial^2}{\partial z^2}\delta t}\psi(z,t) = \int_{-\infty}^{\infty} e^{\frac{-i\hbar}{2m}k^2\delta t}\hat{\psi}(k,t)e^{ikz}\mathrm{d}k \tag{2.10}$$

So, to evaluate the kinetic evolution of the system we can simply multiply the Fourier trasform of the wave function by the phase factor:

$$\exp\left(\frac{-i\hbar}{2m}k^2\delta t\right) \tag{2.11}$$

and then antitrasform the result.
To complete one step of the time evolution, it is now sufficient to multiply the wave function by the potential phase factor ($g := 2\hbar\omega_r a_s N$):

$$\exp\left(\frac{-iV_{\text{eff}}\delta t}{\hbar}\right) = \exp\left(\frac{-i(V_{\text{ext}} + g|\varphi|^2)\delta t}{\hbar}\right) \tag{2.12}$$

which can be easily obtained form the external potential and the square norm of the wave function at every spatial point.

2.2 The Algorithm

The complete algorithm is summarized in the following scheme[1] [2]:

```
1. while t_elapsed < total_time :
2.     PHI_K = FT(PHI_Z)
3.     PHI_K[k] = PHI_K[k]*exp(-i k^2 dt)
4.     PHI_Z = INVERSE_FT(PHI_K)
5.     PHI_Z[z] = PHI_Z[z]*exp(-i (V_ext + |PHI_Z[z]|^2) dt)
6.     t_elapsed = t_elapsed + dt
```

[1] Physical constants have been omitted for clarity; see section 2.1 (equations (2.11) and (2.12)) for the correct values.

[2] In lines 3 (5) the operation is implicitly performed for all values k (z)

This method ensures that the evolution is unitary, therefore the computation is unconditionally stable (i.e. no requirements are needed to ensure that the solution does not blow up or vanish) as well as being *time transverse invariant* , i.e. the probability density remains the same if a constant value is added to the potential.
Besides, it should be noted that the momentum space phase shifts due to the kinetic evolution and the phase shift due to the external potential are constant (as long as the potential does not change) and can thus be precalculated at the beginning of the computation, improving performances.
As mentioned in the introduction, a convenient implementation of this algorithm is provided in the form of a C++ library, as described in appendix A.

Chapter 3

Results

In the following pages we report the results of various simulations that have been carried out to test the integration engine.
The results are in good accord with the theoretical predictions (where such predictions could me made), and enable us to study situations are not amenable to analytical solution.

3.1 Linear Schroedinger equation

In the first place, some standard situations have been considered. The usual Schrödinger equation has been integrated, simulating one-dimensional wave packets evolving in time; this is obtained by putting $g = 2\hbar\omega_r a_s N = 0$, effectively discarding the non-linear therm of equation (1.8).

3.1.1 Gaussian packet

The evolution of a stationary gaussian wave packet in free space has been obtained numerically and then confronted with the theoretical results. In figure 3.2 is shown the shape of the packet at different times, while in figure 3.1 is shown the comparison of the numerically obtained width with the theoretic prediction $\sigma(t) = \sqrt{\sigma^2(0) + \frac{\hbar^2 t^2}{\sigma^2(0) m^2}}$. It can be noted that the packet maintains a gaussian shape while disperding as time advances.

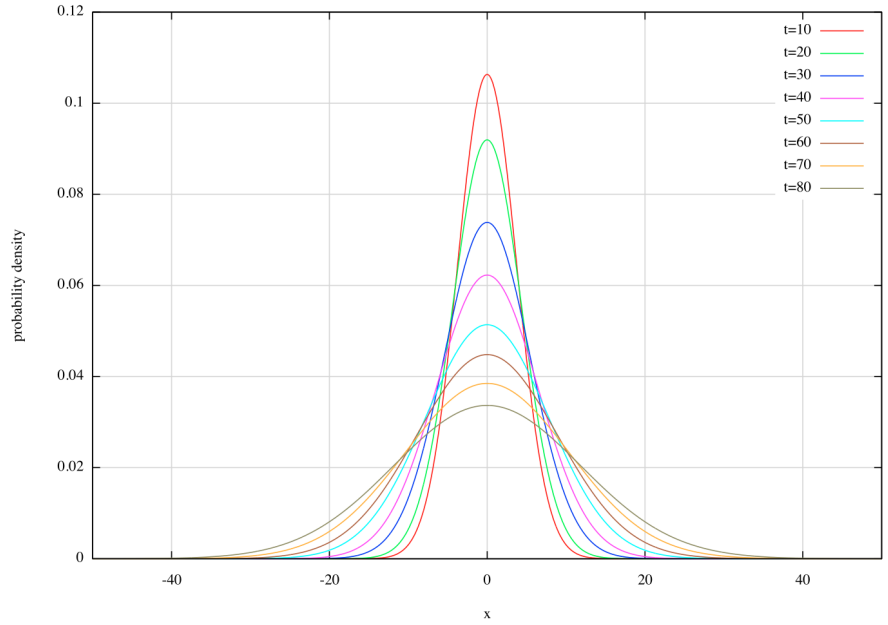


FIGURE 3.1: Time evolution of the probability density for a gaussian packet; $\sigma(0) = 5$, $\delta z = 0.25$, $\delta t = 0.04$

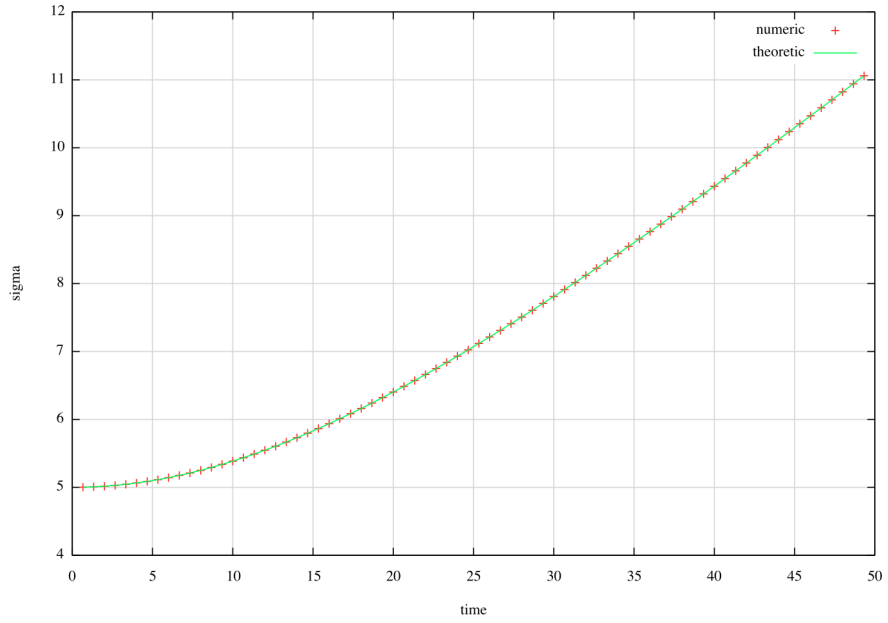


FIGURE 3.2: Gaussian packet width as a function of time; $\sigma(0) = 5$, $\delta z = 0.25$, $\delta t = 0.04$

3.1.2 Tunnelling

Tunnelling of a gaussian-shaped packet through rectangular and gaussian potential barriers has been simulated, using the Wentzel-Kramers-Brillouin (WKB) approximation

to obtain analytical predictions in the second case.

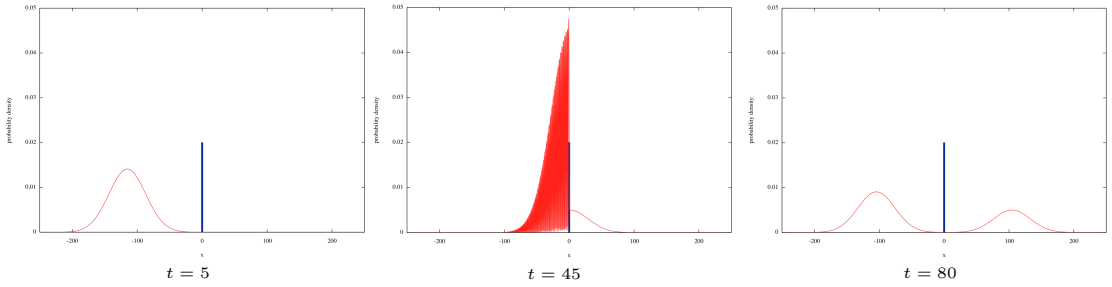


FIGURE 3.3: Tunnelling trough a rectangular barrier; probability density at various times. $k = 5$, $L = 3.5$, $V_0 = 10$, $\delta t = 0.01$, $\delta z = 0.05$

Figure 3.4 show the comparison between numerical and theoretical transmission coefficients for a rectangular barrier with height V_0 and width L as a function of $\kappa = E/V_0$. Theoretical values are calculated for a plane wave with the same energy as the incident gaussian packet. In figure 3.5 is reported a more accurate calculation of the first peak shown in figure 3.4; this is obtained by using a broader packet, which corresponds to a more peaked spectrum, approximating more the plane wave behaviour.

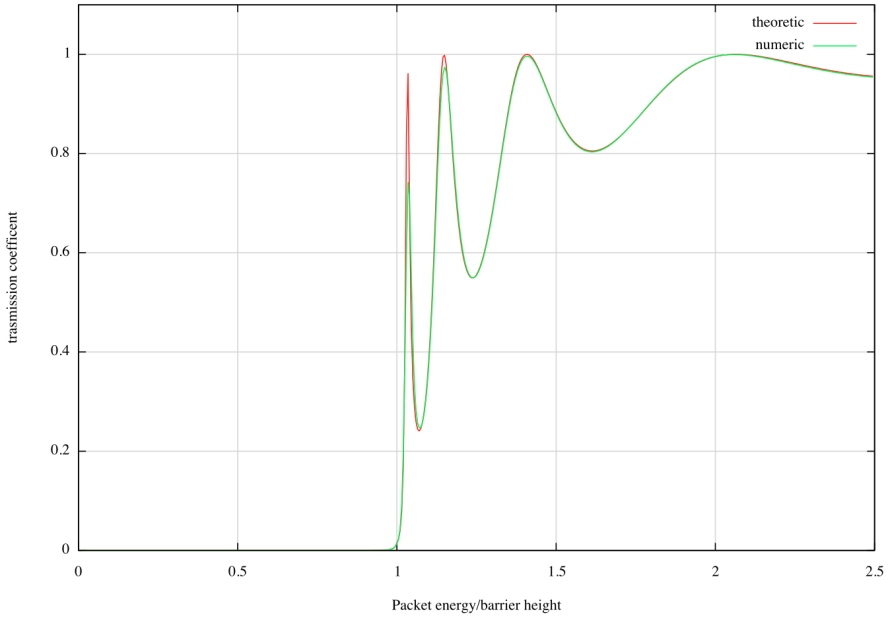


FIGURE 3.4: Trasmission coefficent as a function of κ for a rectangular barrier; packet width $= 50$, $k = 5$, $L = 3.5$, $\delta t = 0.01$, $\delta z = 0.04$

Tunnelling through a gaussian shaped barrier is confronted in figure 3.6 with the results of WKB approximation. The table shows that the WKB approximation is accurate only

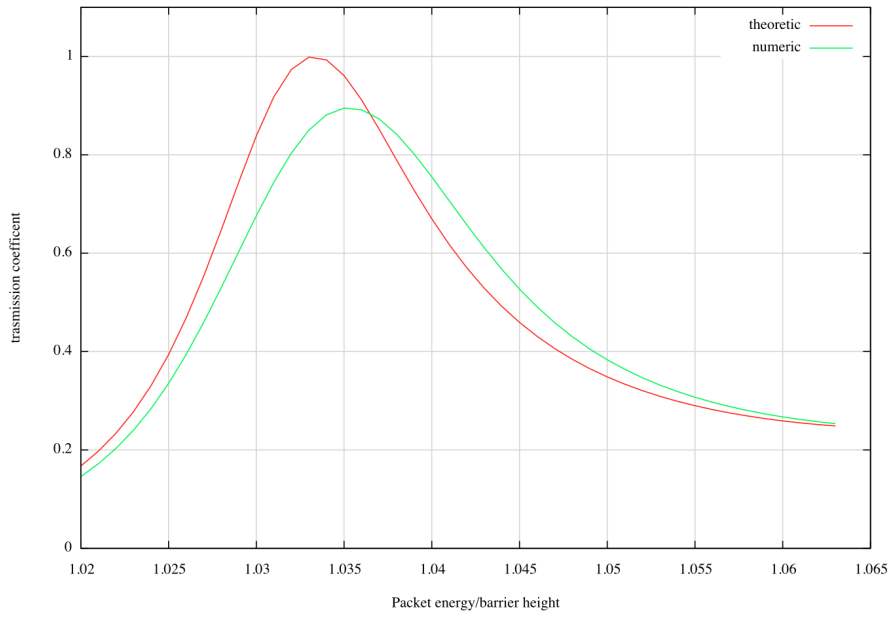
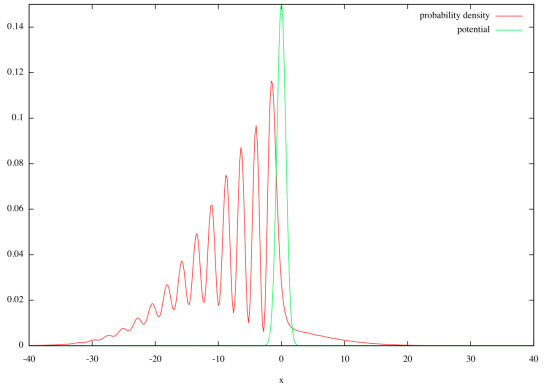


FIGURE 3.5: Trasmission coefficent as a function of κ (detail); packet width = 100, $k = 5$, $L = 3.5$, $\delta t = 0.01$, $\delta z = 0.04$



$\mathbf{E/V_0}$	**Numerical**	**WKB**
1.00	0.567	1
0.80	0.272	0.592
0.50	0.015	0.030
0.43	0.005	0.006
0.40	0.003	0.003
0.37	0.001	0.001
0.30	0.000	0.000

FIGURE 3.6: Left: snapshot of the collision of a gaussian packet on a gaussian potential barrier; Right: Transmission coefficient as a function of κ. Packet width = 10, barrier $\sigma = 10$, $k = 1.7$, $\delta t = 0.02$, $\delta z = 0.14$

in low transmission regime, and becomes progressively less precise when the packet's energy approaches the height of the barrier.

3.2 Gross-Pitaevskii equation

The results of numerical simulations using the one dimensional GPE will now be presented.
One of the most interesting properties of the non linear Schrödinger equation is the fact that it's possible to cancel the dispersive behaviour using appropriately shaped wave packets, enabling the creation of bright and dark solitons.
Attractive interactions (i.e. $a_s < 0$) are needed to create bright solitons, while repulsive forces are required to produce dark ones.

3.2.1 Bright soliton simulation

It can be proven [3] that a moving bright soliton is described by the following wave function:

$$\varphi(z) = \sqrt{\frac{|\eta|}{2}} sech(z|\eta|)e^{ikz} \tag{3.1}$$

where $\eta = \frac{m\omega_r N a_s}{\hbar}$.

Its propagation has been studied, verifying that the shape and width of the packet remains constant in time.

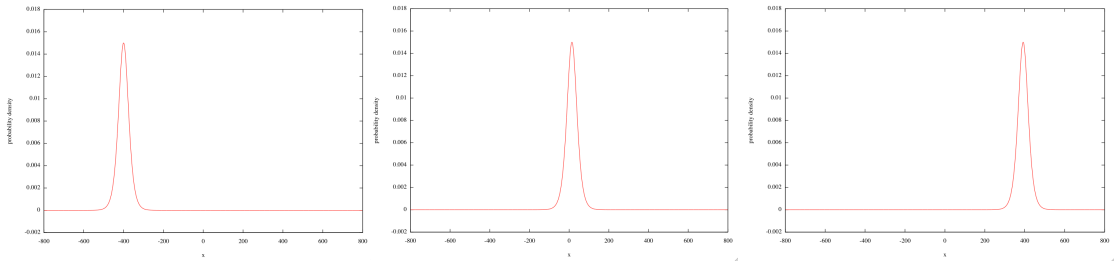


FIGURE 3.7: Soliton evolution, probability density at various times; $\eta = -0.03$, $k = 3$, $\delta t = 0.02$, $\delta z = 0.39$

As shown in figure (3.8) is constant within one part in ten million after 12500 steps, confirming that the considered packet is a soliton.

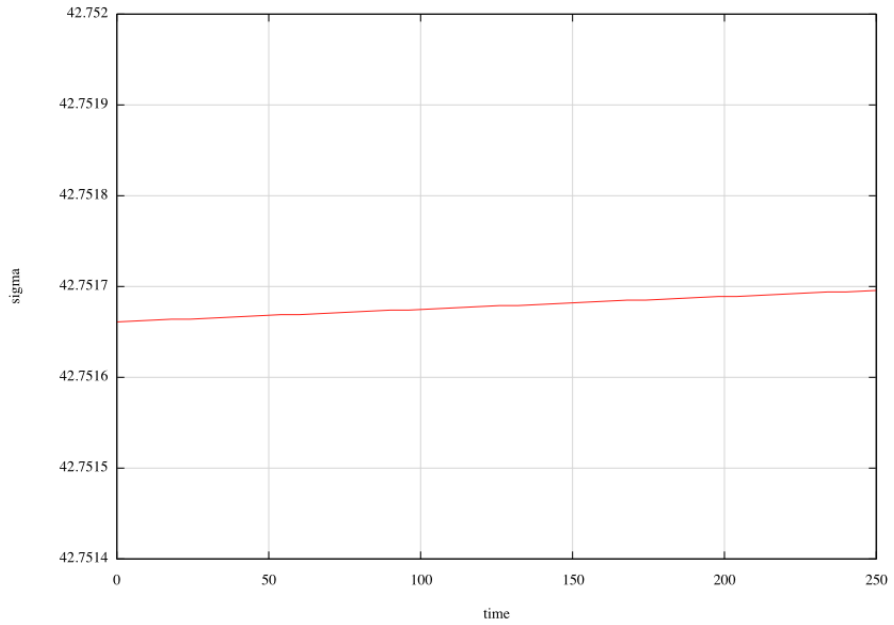


FIGURE 3.8: Soliton width as a function of time in numerical simulation; $\eta = -0.03$, $k = 3$, $\delta t = 0.02$, $\delta z = 0.39$

3.2.2 Linear-Non linear comparison

In this section we will present a comparison between the tunnelling through a rectangular potential barrier of a condensate soliton and a similar shaped standard wave packet. All the results presented have therefore been obtained using as starting state the soliton wave function (equation 3.1) and confronting its evolution under the GPE and the SE.

One of the main differences between a normal wave packet and the condensate is that the latter possesses an amount of internal energy that has to be added to the packet kinetic energy. It's then clear that the ratio between the internal energy of the soliton and it's kinetic energy plays an important role in the dynamics of the system.
Since the two packets have the same initial wave function, their starting kinetic energy is the same:

$$\langle\varphi|\,\hat{T}\,|\varphi\rangle = \frac{-\hbar^2}{2m}\int_{-\infty}^{\infty}\varphi^*\nabla^2\varphi\;dx = \frac{\hbar^2}{2m}\left(k^2+\frac{1}{3}\eta^2\right) \tag{3.2}$$

As said before, the condensate has also the chemical term (remember that $g = 2\hbar\omega_r aN$):

$$\langle\varphi|\,\hat{U}\,|\varphi\rangle = \int_{-\infty}^{\infty}\frac{1}{2}g|\varphi|^4dx = \frac{g|\eta|}{6} \propto \eta^2 \tag{3.3}$$

Thus:

$$\frac{\langle \hat{T} \rangle}{\langle \hat{U} \rangle} = \frac{3}{2}\frac{k^2}{\eta^2} + \frac{1}{2} \tag{3.4}$$

Equation (3.4) shows that two different energy regimes are possible: $\langle \hat{T} \rangle \simeq \langle \hat{U} \rangle$ and $\langle \hat{T} \rangle \gg \langle \hat{U} \rangle$. For the first case we must require:

$$3k^2 + \eta^2 \simeq \frac{m}{\hbar^2} g|\eta| = 2\eta^2 \tag{3.5}$$

That is,

$$3k^2 \simeq \eta^2 \Rightarrow k \simeq \eta \tag{3.6}$$

In figures 3.9 and 3.10 is reported the comparison between transmission coefficients as a function of κ in the two energy conditions.

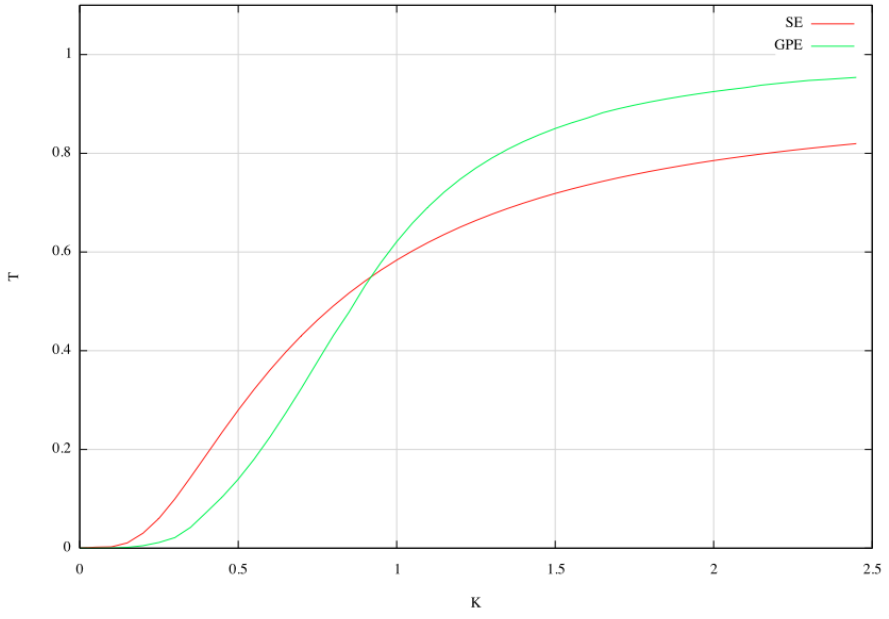


FIGURE 3.9: Trasmission coefficient as a function of κ, with $\langle \hat{T} \rangle \simeq \langle \hat{U} \rangle$; $k = 0.03$, $\eta = -0.03$, $L = 35$, $\delta t = 2$, $\delta z = 0.75$

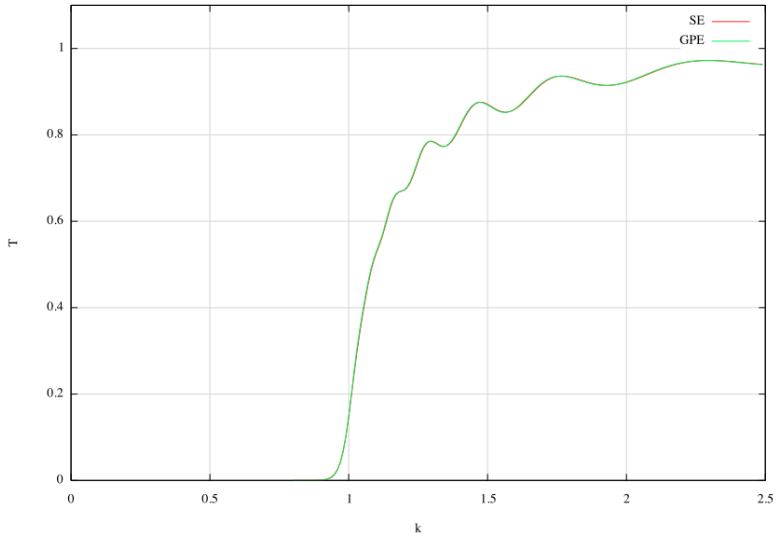


FIGURE 3.10: Trasmission coefficient as a function of κ, with $\langle\hat{T}\rangle \gg \langle\hat{U}\rangle$; $k = 1$, $\eta = -0.03$, $L = 35$, $\delta t = 0.06$, $\delta z = 0.75$

As expected, significant differences between the condensate and the normal wave packet arise when the internal energy of the condensate is comparable with the kinetic term, allowing a redistribution of the total energy during the evolution.
In figure 3.11 is shown the transmission coefficient as a function of the energy ratio $\frac{\langle\hat{T}\rangle}{\langle\hat{U}\rangle}$ for $\kappa = 1.5$, showing how the difference between two behaviours goes to zero as the internal energy becomes irrelevant.

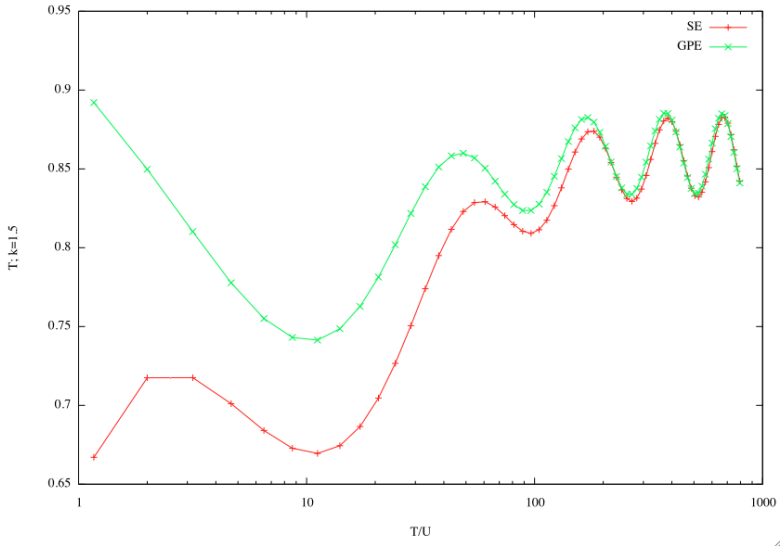


FIGURE 3.11: Transmission coefficient vs $\frac{\langle\hat{T}\rangle}{\langle\hat{U}\rangle}$; $\kappa = 1.5$, $\eta = -0.03$, $L = 35$, $\delta t = \frac{0.06}{k}$, $\delta z = 0.75$

In the following figures is shown the change of the energy distribution in time for various values of κ.

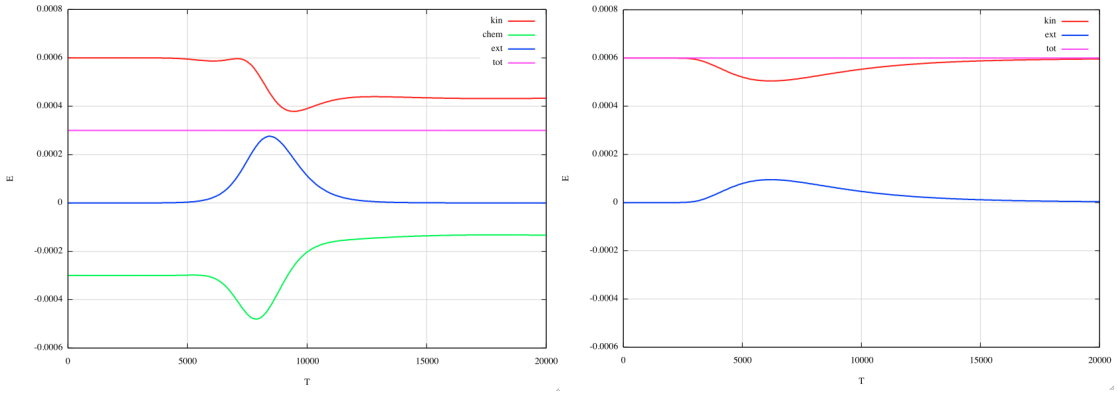


FIGURE 3.12: Left: GPE, right: SE, $\kappa = 0.6$; $k = 0.03$, $\eta = -0.03$, $L = 35$, $\delta t = 0.5$, $\delta z = 0.75$

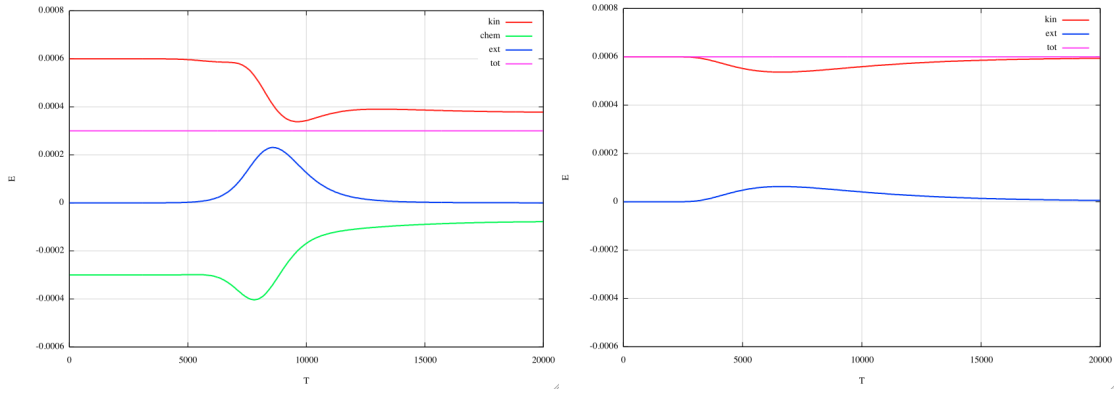


FIGURE 3.13: Left: GPE, right: SE, $\kappa = 0.95$; $k = 0.03$, $\eta = -0.03$, $L = 35$, $\delta t = 0.5$, $\delta z = 0.75$

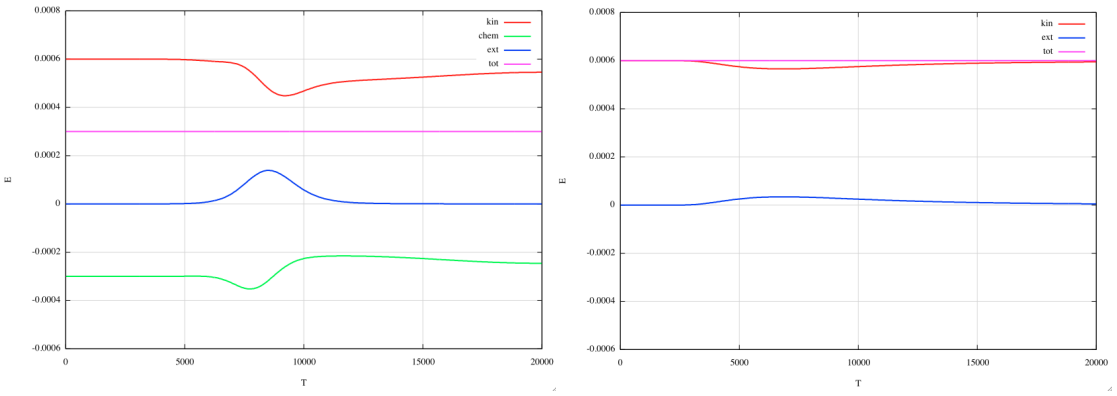


FIGURE 3.14: Left: GPE, right: SE, $\kappa = 2$; $k = 0.03$, $\eta = -0.03$, $L = 35$, $\delta t = 0.5$, $\delta z = 0.75$

Appendix A

Library Overview

The algorithm described in section 2.2 has been implemented as a part of a convenient C++ library which can be used to perform simulation of physical systems.
The complete code is available at `http://www.github.com/pnjun/gpsys` and is released under a Creative Commons Attribution licence.

The library takes care of handling the computation data and parameters, simplifying the implementation of the physical simulations. In addition, it is completely thread safe, allowing multiple simulations to be run simultaneously by different threads.
Two main classes are provided, one is used to store the information about the state of the system, while the other manages the actual simulation.

The State Class

The State Class contains the methods needed to control the state of the system:

```
class State
{
public:
    State(double start, double end, int N, double mass,
          double gas_parameter);
    ~State();

    double *wf;

    void setState(void (*state)(double, double*, double*));
```

```
    void setState(void (*state)(double, double*, double*,
                                                void*), void*);
    void normalize();
    [...]
};
```

The constructor takes care of memory allocation, and stores internally the spatial starting and ending point (as well as the spatial step) allowing the user to specify the starting state via a function of a continuous variable, which get automatically sampled by the `setState(...)` functions. The state is stored as an array of $2N$ values (N is the space point number), with the even (odd) indexed values corresponding to the real (imaginary) part of the WF. The class also provide a method to normalize the state, as well as others convenient functions (not shown in the code above).

The GPSys Class

This class handles the actual time evolution:

```
class GPSys
{
public:
    GPSys(double start, double end,
          int slices, double dt_in, double mass,
          double hbar, double eff_gas_parameter);
    ~GPSys();

    State* state;

    void next();
    void setPotential(double (*pot)(double));
    void setPotential(double (*pot)(double,void*),
                                    void*);
    [...]
};
```

The class constructor creates a state and prepares the data to begin a new simulation; it also precalculates all the momentum space phase shift caused by the kinetic evolution. The external potential can be set using `setPotential(...)` functions, which sample the given potential and store the induced phase shift. The `next()` function moves the

state one step ahead in time. A function that gives the total time elapsed since the beginning of the simulation is also present.

Bibliography

[1] Christopher Pethick and Henrik Smith. *Bose-Einstein Condensation in Dilute Gases*. Cambridge University Press, 2001.

[2] R Carretero-González, DJ Frantzeskakis, and PG Kevrekidis. Nonlinear waves in bose–einstein condensates: physical relevance and mathematical techniques. *Nonlinearity*, 21(7), 2008.

[3] Victor M. Perez-Garcia, Humberto Michinel, and Henar Herrero. Bose-einstein solitons in highly asymmetric traps. *Phys. Rev. A*, 57:3837–3842, 1998.