

Institut für Physik
Karl-Franzens-Universität Graz

Bachelor-Thesis

**Analyzing the Dynamics of an
atomic Bose-Einstein-Condensate
within a Two-Mode Model**

Anna Aichmayr

Supervisor: Mag. Dr. Ulrich Hohenester

November 25, 2010

Contents

1	Theory	5
1.1	Bose-Einstein Condensates	5
1.2	Two-Mode Model	6
1.3	Bose-Einstein Condensates beyond Mean Field Theory	8
2	Implementation in MATLAB	10
2.1	Description	10
2.2	Expansions	14
3	Task	16
4	Results	18
5	Conclusion	33
6	Appendix	35
7	Literature	38

Abstract

In my bachelor thesis I studied Bose-Einstein Condensates described by a two-mode model using a MATLAB [4] program. I was especially interested in external influences on the system. In this case the trap for bosons was a double well potential. The investigated external influences corresponded to the shape of the double well potential. Three cases were simulated in particular:

First, the inequality of the depth of both wells was constant over time. Second the difference had a maximum on the first step and decreased exponentially. Third the difference oscillated with certain fixed frequencies. The basic program was from my supervisor Mag. Dr. Hohenester. My task was to extend it and to include the inequalities in the double well potential mentioned before. To solve the problem, the wave function was calculated using the Schrödinger equation. Another possibility to solve this is called "mean field theory", which only uses the wave function's mean and standard deviation for description of the result. This is the reason why I implemented the solution also via mean field theory to be able compare those obtained results.

Zusammenfassung

In meiner Bachelor-Arbeit untersuchte ich anhand eines Computerprogramms Bose-Einstein-Kondensate (BEK) und wie sie sich verhielten, wenn spezielle externe Einflüsse auf dieses System einwirkten. Bei einem BEK handelt es sich um einen Atomchip, auf dem sich stromdurchflossene Leiterdrähte befinden und somit eine Falle für Bosonen entsteht. Diese Falle sieht aus wie ein harmonischer Oszillator, in dessen Mitte eine Barriere aufgebaut wird, sodass zwei Potenzialtöpfe entstehen. Hauptsächlich wurde untersucht, welchen Einfluss es hat, wenn die beiden Potenzialtöpfe nicht die gleiche Tiefe besaen. Es wurde dabei distinguiert, ob die Unterschiede der Tiefen konstant blieben, wieder verschwanden oder sich veränderten bzw. oszillierten. Zur Umsetzung wurde MATLAB [4] gewählt. Als Basisprogramm diente ein Programm von Herrn Mag. Dr. Hohenester, welches von mir erweitert und untersucht wurde. Dieses Programm benutzt zur Lösung der Wellenfunktion die Schrödingergleichung. Ein anderer Ansatz zur Lösung nennt sich "mean field theory", worin nicht mit der Wellenfunktion gerechnet wird, sondern nur der Mittelwert und die Standardabweichung dieser genommen wird. Eine weitere Aufgabe für mich war anhand eines Beispiels zu vergleichen, ob diese beiden Lösungsansätze wirklich gleichwertig seien.

Theory

1.1 Bose-Einstein Condensates

A Bose-Einstein condensate, short BEC, consists of a gas of weakly interacting bosons which are confined in an external potential and cooled down to temperatures near to absolute zero. Due to the low temperature most of the bosons occupy the ground state (lowest state) of the external potential, whereby quantum effects on a macroscopic scale can be observed. Through freezing single degrees of freedom, low dimensional systems with novel physical characteristics are generated.

In former times it was necessary to build huge hardware to create such BECs, but nowadays when microelectronics meet quantum mechanics all one needs is the so called Atom chip to fabricate a BEC. Such a chip has the size of approximately 4 square centimeters. Its core consists of micro fabricated circuit paths that are arranged in a special order. By using different voltages and currents one can build traps near the surface of an Atom chip to hold the gas consisting of bosons. Furthermore it is possible to cool those bosons down and manipulate them.

For example BECs are used to analyze the structure of local magnetic fields or the transport of electrons in metal. Moreover it is possible to manipulate the trap by raising a potential barrier in the middle of the chip in such a way that the BEC is split into two halves. After turning off the magnetic trap the wave function of both halves overlap and interfere [1].

The aim of physicists is to understand the Josephson effect, which controls the tunneling current between two weakly coupled superconductors that are separated by a very thin isolating barrier - at least this was the way this effect was first proven. More generally spoken, this effect describes two weakly coupled macroscopic wave functions that are separated by a tunnel barrier [2].

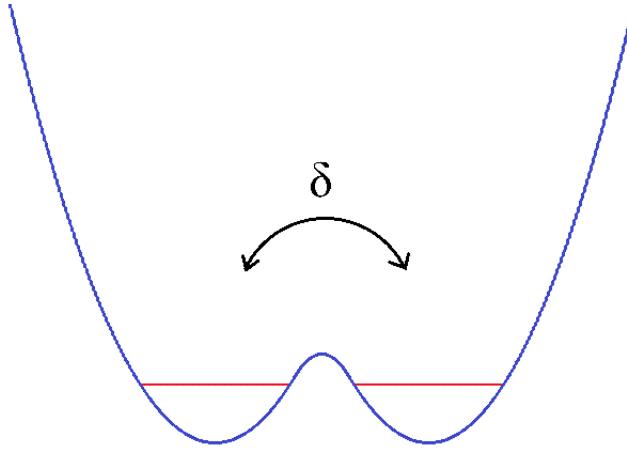


Figure 1: δ is called tunneling rate of a particle in the double well potential. This rate stands for the movement of the particle in between both wells.

1.2 Two-Mode Model

In my bachelor thesis I concentrated on the theoretical description of a double trap, where a BEC is split into two halves. To describe this the two-mode model was chosen.

In the two-mode model only two possible states of a particle are considered: the ground state and the excited state. This assumption is correct due to the fact that one can nearly cool down BECs to absolute zero, which leads to a drastic slowing down of all particles in the system. According to the temperature the higher states are artificially turned off, most of the particles tend to occupy the ground state.

A big advantage that comes along with those assumptions is the fact, that formulas and calculations get much easier although the results stay accurate enough.

The system that is described by the two-mode model is a so-called symmetric double well potential, which is a modified harmonic oscillator. The modification is a barrier in the middle, so that the well is split into two equal wells. In my bachelor thesis I consider cases where both wells are not equally high. They can have a constant difference but the difference can also change when time passes.

In such a system there are three key parameters to describe the behavior of the particles. The first is the so-called nonlinearity-parameter κ , it describes the effect of particle-particle repulsion.

The second is called single-particle tunneling rate δ and describes the movement of a particle in between both wells.

The last key parameter is λ , which stands for the difference in height of two wells. δ is depicted in figure 1, λ can be seen in figure 2 .

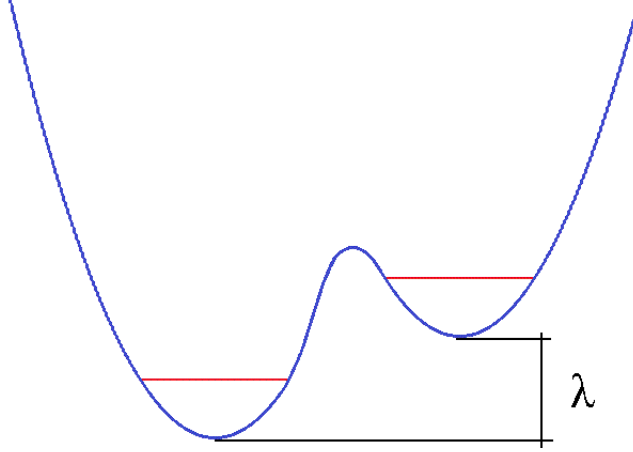


Figure 2: In this figure one can see one of the key parameters to describe the double well potential: λ . It stands for the inequality of height of both wells.

In a symmetric double well potential, the ground state of a particle is characterized by the even wave function ψ_g that belongs equally to both wells of the potential. Given that the tunneling rate depends on the height of the barrier in between the two wells of the potential, a barrier is provided which is high enough so that the tunneling rate between the potential wells is small. The excited odd state ψ_e also belongs to both halves of the double well.

The superpositions $\psi_{l,r} = \frac{1}{\sqrt{2}} \cdot (\psi_g \pm \psi_e)$ are states in which the particle is either on the left or on the right side of the barrier. In fact those states are not stationary - the particles can tunnel from one to the other half of the potential.

The basic assumption for the two mode model is that there are only those two one-particle states ψ_g and ψ_e available for N observed bosons. The two-particle contact interaction then is:

$$U(r_1, r_2) = \frac{4\pi\hbar^2 a}{m} \cdot \delta(r_1 - r_2),$$

where a is the s -wave scattering length and m the atomic mass.

The many-body Hamiltonian is given by:

$$H = -\frac{\delta}{2} \cdot (a_l^\dagger a_r + a_r^\dagger a_l) - 4\kappa a_l^\dagger a_l a_r^\dagger a_r.$$

We set $\hbar = 1$, a_l and a_r are the boson operators for annihilation and creation of the left and right-localized states of the particles. They are a combination of the boson operators for the ground and excited wave functions a_g and a_e :

$$a_l = \frac{1}{\sqrt{2}} \cdot (a_g + a_e),$$

$$a_r = \frac{1}{\sqrt{2}} \cdot (a_g - a_e).$$

κ and ϵ are constants and describe the one- and two-particle matrix elements:

$$\epsilon_e = \int d^3r \psi_e(r) \left[-\frac{1}{2m} \nabla^2 + V(r) \right] \psi_e(r),$$

$$\epsilon_g = \int d^3r \psi_g(r) \left[-\frac{1}{2m} \nabla^2 + V(r) \right] \psi_g(r),$$

$$\kappa_{ee} = \frac{2\pi a}{m} \int d^3r |\psi_e(r)|^2 |\psi_e(r)|^2,$$

$$\kappa_{gg} = \frac{2\pi a}{m} \int d^3r |\psi_g(r)|^2 |\psi_g(r)|^2,$$

$$\kappa_{eg} = \frac{2\pi a}{m} \int d^3r |\psi_e(r)|^2 |\psi_g(r)|^2,$$

where $V(r)$ is the symmetric double well binding potential. From ϵ_e and ϵ_g we receive one of the two key parameters of this many-body problem, the single-particle tunneling rate δ :

$$\delta = \epsilon_e - \epsilon_g$$

The second key parameter is called nonlinearity-parameter κ :

$$\kappa = \kappa_{ee} = \kappa_{eg} = \kappa_{gg}.$$

The particle number N can be calculated as follows:

$$N = a_e^\dagger a_e + a_g^\dagger a_g$$

N is conserved. [2]

1.3 Bose-Einstein Condensates beyond Mean Field Theory

Mean field theory is another possibility to describe a BEC. Again we consider a condensate in which particles can only effectively populate two second-quantized modes, which is realized in a double well trap.

The Hamiltonian is now given by:

$$H = -\frac{\delta}{2} \cdot (a_g^\dagger a_e + a_e^\dagger a_g) + \frac{\kappa}{2} \cdot (a_g^\dagger a_g - a_e^\dagger a_e)^2,$$

where δ is the coupling strength between two condensate modes and κ represents the two-body interaction strength. In the next step the angular momentum operators are defined:

$$L_x = \frac{1}{2} \cdot (a_g^\dagger a_e + a_e^\dagger a_g)$$

$$L_y = \frac{1}{2i} \cdot (a_g^\dagger a_e - a_e^\dagger a_g)$$

$$L_z = \frac{1}{2} \cdot (a_g^\dagger a_g - a_e^\dagger a_e)$$

Inserting these formulas into the Hamiltonian we obtain:

$$H = -\delta L_x + \frac{\kappa}{2} L_z^2$$

The Heisenberg equations of motion for the three angular momentum operators read:

$$\frac{d}{dt} L_x = -i[L_x, H] = -\frac{\kappa}{2}(L_y L_z + L_z L_y),$$

$$\frac{d}{dt} L_y = -i[L_y, H] = +\delta L_z + \frac{\kappa}{2}(L_x L_z + L_z L_x),$$

$$\frac{d}{dt} L_z = -i[L_z, H] = -\delta L_y.$$

The mean field equations in the two-mode model are obtained by approximating second order expectation values $\langle L_i L_j \rangle$ as products of the first order expectation values $\langle L_i \rangle$ and $\langle L_j \rangle$:

$$\langle L_i L_j \rangle \approx \langle L_i \rangle \langle L_j \rangle.$$

Finally we obtain the nonlinear Bloch equations defining the single-particle Bloch vector $\vec{s} = (S_x, S_y, S_z) = (\frac{2\langle L_x \rangle}{N}, \frac{2\langle L_y \rangle}{N}, \frac{2\langle L_z \rangle}{N})$, $\kappa' = \frac{\kappa N}{2}$:

$$\dot{S}_x = -\kappa' S_z S_y,$$

$$\dot{S}_y = \delta S_z + \kappa' S_z S_x,$$

$$\dot{S}_z = -\delta S_y,$$

The norm $|\vec{s}|$ is conserved. [3]

Implementation in MATLAB

For the implementation of the model MATLAB [4] was chosen, because it is optimized for vector and matrix operations and numerical calculations. Moreover it comes along with a variety of predefined functions and supports graphical output, which reduces the degree of difficulty of the realization by far.

The program is basically split into two sections - this is realized by two folders. The first is named "twomode" and contains all functions that are necessary for all calculations. The second folder is called "Demo" and consists of the main program plus the plot. In the following section each component of the program is described in detail in alphabetical order. Each title of the following sections corresponds to a homonymous .m-file.

2.1 Description

twomode

blochtransform

This function requires two input parameters ϕ and θ , which stand for an azimuthal and a polar angle, and returns a rotation matrix *rot*. Its task is to transform the two angles for the pseudo spin state into the matrix *rot*.

Assuming the distance between the origin of the system and the point is always constant there is only the need for two angles to be able to describe every point on the surface of the sphere.

densitymatrix

In "densitymatrix.m" two matrices ρ and c are calculated based on an input wave function ψ . The matrix ρ describes the one-particle density matrix, whereas c

characterizes a two-particle correlation function for a given wave function ψ .

figbloch

A wave function is plotted on the Bloch sphere. The following input values are accepted:

- 'save', 'filename'
- 'load', 'filename'
- ψ , where ψ is a wave function

The first input parameters lead to saving a certain plot to the file 'filename.mat', which is a binary MATLAB format for matrices. The second version leads to loading of a certain matrix file into the workspace - if there is no extension it is treated as a .mat-file, if there is an other extension, it is treated as an ASCII-file. The last possibility for the input plots the wave function ψ onto the Bloch sphere using the function "triplot".

It is important to notice that one has to load an empty Bloch matrix first before a direct plot of a wave function ψ onto the sphere is possible. Because of that the file bloch100.m in "Demo" represents such an empty matrix.

hamtwomode

The Hamiltonian is calculated and returned in this function. The following input parameters are required:

- δ represents the tunnel coupling
- λ describes the difference in height of the two wells

To compute the Hamiltonian the following formula is used:

$$H = -\frac{1}{2} \cdot \delta \cdot J_1 + 4 \cdot \kappa \cdot J_3^2 + \lambda \cdot J_3$$

J_1 and J_3 are pseudo spin operators that are included into the function by calling the method "pseudospin". κ corresponds to the nonlinearity parameter and is included via the method "paramtwomode".

numberfluctuations

This method calculates the number fluctuation for a given state ψ .

paramtwomode

The function "paramtwomode" manages the most important values of the whole program:

- the particle number n
- the nonlinearity parameter κ
- the tunnel function *tunnelfun*

It requires at least two input parameters, where the first is either 'set' or 'get'. The second input parameter can be either one, two or all of the values above. The output completely depends on the chosen input which is quite self explanatory.

pseudospin

The pseudo spin operators for the two mode Hamiltonian J_1, J_2 and J_3 are calculated in this function using the pseudo spin lowering and raising operators J_m and $J_p = J_m^\dagger$:

$$J_1 = \frac{1}{2} \cdot (J_p + J_m)$$

$$J_2 = -\frac{1}{2}i \cdot (J_p - J_m)$$

$$J_3 = \frac{1}{2} \cdot (n_1 + n_2)$$

where n_1 is the number of particles in the first well and n_2 is the number of particles in the second well.

There are two possible usages: On the one hand the initialization requires the input parameters 'init', the total particle number n and the atom difference vector m . On the other hand this function returns the pseudo spin operators J_1, J_2 and J_3 by passing the desired indices.

triplot

A figure showing the wave function on the Bloch sphere is created using this function. To use it one has to pass three arguments: fv contains the vertices and faces of triangulated surface. v represents the function to be plotted. The last input value stands for a color map but is not obligatory. If it is not specified, 'default' is used.

twomodecrank

This function computes the solution of the Schrödinger equation using the Crank-Nicholson scheme. Although this scheme is inaccurate to $O(\delta t^2)$ it is numerically stable, norm conserving and has a unitary time evolution. Its input parameters are the initial wave function ψ_0 , the time t_{out} and λ . It returns the wave function at the final time ψ , the number fluctuation dn and the output wave function ψ_{sav} for plotting.

On every time step of t_{out} the total Hamiltonian is calculated calling the function "hamtwomode". After that, the new wave function ψ_t is calculated as follows:

$$\psi_t = (1 + \frac{1}{2}i \cdot \delta t \cdot H) \cdot (1 - \frac{1}{2} \cdot \delta t \cdot H)^{-1} \cdot \psi_t$$

Last but not least the number fluctuation dn is computed using the function "numberfluctuation". ψ_{sav} is a matrix containing ψ_t for every time step t .

twomodecummulant

The cumulant expansion for the two mode model is solved in this function. It requires the initial wave function ψ_0 and the time t_{out} as input parameters and returns the mean value of the spin operators ρ , the quadratures of the spin operators c and the number fluctuation dn . To calculate the solutions the equation of motion for the mean value and quadrature of the spin operators is solved.

twomodegroundstate

The ground state ψ_0 for the two mode model is calculated here. At first the Hamiltonian is computed via "hamtwomode" and afterwards a few eigenvalues are calculated using the MATLAB-function "eigs". "eigs" returns a diagonal matrix of H's 6 largest magnitude eigenvalues and a matrix whose columns are the corresponding eigenvectors, which is ψ_0 in this case.

Demo

demotwomode

This is the main program which manages and calls all functions that are described in the sections above.

First the values for the following parameters can be fixed: particle number n , atom difference vector m , tunnel coupling δ , nonlinearity parameter κ , height difference of the wells λ and time scale t_{out} . After that, the calculation starts by calling the functions "pseudospin" to initialize the pseudo spin operators, "paramtwomode"

to set all important parameters and "twomodegroundstate" to receive the initial wave function. Next the Schrödinger equation is solved using "twomodecrank". Finally the function offers the possibility to plot the calculated wave functions.

bloch100

"bloch100" is a .mat file, which is required to create the Bloch plots. It basically contains an empty Bloch sphere. In detail, it consists of a 10201x101-rotation matrix *rot* and a 1x1-struct *fv* which in turn contains a 10000x4-matrix *faces* and a 10201x3-matrix *vertices*.

2.2 Expansions

Besides changes to existing functions like adding terms containing λ which are already included in the descriptions above, I added the following three functions to the existing program.

meanfield

The function "meanfield" is located in folder "twomode". It calculates the mean value using a density matrix formalism instead of a wave function approach. The required input parameters are the number of particles n , the coupling strength between the two condensate modes ω and the initial wave function ψ_0 . The output parameters are the coordinates S_x, S_y, S_z of the mean and the standard deviation Δ . The start values $S_{x,0}, S_{y,0}, S_{z,0}$ are obtained through the following formula:

$$S_{i,0} = \frac{2}{n} \cdot (\psi_0^\dagger \cdot \text{pseudospin}(i) \cdot \psi_0)$$

The values for S_x, S_y, S_z are computed using the MATLAB-function "ode45", where "Sfun" describes the differential equation to be solved.

plotbloch

After calling the main program it is useful to see the calculated results on the Bloch sphere. This function contains the corresponding calls.

At first the figure window is rescaled to full screen. Afterwards the wave function ψ is plotted onto the Bloch sphere every tenth time step.

Alternatively, if the mean has also been calculated using "meanfield", one can plot both the wave function and the mean value onto the Bloch sphere, where the mean is represented as a green star.

Sfun

This function is used by the "meanfield"-function, it contains the differential equation to calculate the values for mean field S and fluctuations Δ . The number of particles n , the nonlinearity parameter κ and the coupling strength ω characterize the input values. The following formula describes the used differential equation from [3]:

$$\dot{S}_x = -\kappa \cdot S_z \cdot S_y - \frac{\kappa}{2} \cdot \Delta_{yz}$$

$$\dot{S}_y = \omega \cdot S_z + \kappa \cdot S_z \cdot S_x + \frac{\kappa}{2} \cdot \Delta_{xz}$$

$$\dot{S}_z = -\omega \cdot S_y$$

$$\dot{\Delta}_{xz} = -\omega \cdot \Delta_{xy} - \kappa \cdot S_z \cdot \Delta_{yz} - \kappa \cdot S_y \cdot \Delta_{zz}$$

$$\dot{\Delta}_{yz} = \omega \cdot (\Delta_{zz} - \Delta_{yy}) + \kappa \cdot S_z \cdot \Delta_{xz} + \kappa \cdot S_x \cdot \Delta_{zz}$$

$$\dot{\Delta}_{xy} = (\omega + \kappa \cdot S_x) \cdot \Delta_{xz} - \kappa \cdot S_y \cdot \Delta_{yz} + \kappa \cdot S_z \cdot (\Delta_{xx} - \Delta_{yy})$$

$$\dot{\Delta}_{xx} = -2 \cdot \kappa \cdot S_y \cdot \Delta_{xz} - 2 \cdot \kappa \cdot S_z \cdot \Delta_{xy}$$

$$\dot{\Delta}_{yy} = 2 \cdot (\omega + \kappa \cdot S_x) \cdot \Delta_{yz} + 2 \cdot \kappa \cdot S_z \cdot \Delta_{xy}$$

$$\dot{\Delta}_{zz} = -2 \cdot \omega \cdot \Delta_{yz}$$

Chapter 3

Task

I investigated the behavior over time of the system on temporal modifications of the control parameters. Especially the influences of an inequality of the height of two wells λ were investigated.

So the Hamiltonian

$$H = -\frac{1}{2} \cdot \delta \cdot J_1 + 4 \cdot \kappa \cdot J_3^2$$

was modified as follows:

$$H = -\frac{1}{2} \cdot \delta \cdot J_1 + 4 \cdot \kappa \cdot J_3^2 + \lambda \cdot J_3 .$$

First a certain constant value was chosen for λ and the progress over time using diverse starting parameters was investigated.

Second λ was set to a certain value on the first time step and then decreased exponentially over time:

$$\lambda = \lambda_0 \cdot e^{-0.2 \cdot t}$$

Third, λ was changed such that the height of the two wells was oscillating. It was calculated as follows:

$$\lambda = \lambda_0 \cdot \cos(t)$$

Fourth, to investigate resonance the oscillation was set smaller assuming the resonance frequencies of the particles were small. λ developing in time was now given by:

$$\lambda = \lambda_0 + \eta \cdot \cos(t),$$

This led to λ oscillating between $\lambda_0 \pm \eta$, where η was a small number like 0.1. Finally the solution of the mean field approach was compared to the exact solution.

To realize those investigations the existing MATLAB program was modified - λ was inserted into the according parts of the program, the Hamiltonian was extended and functions to calculate the solution using the mean field approach were added which were described in detail in the chapter before.

The main parameters for all mentioned investigations were the inequality of the height of both wells λ , the tunnel coupling δ and the nonlinearity parameter κ , all other parameters like the number of particles N were constant during the process of research.

Chapter 4

Results

The first figures depict the normal system without external influences. The following numbers were chosen as parameters:

$$\lambda = 0,$$

$$n = 100,$$

$$\kappa = \frac{1}{n},$$

$$\delta = 10 \cdot e^{-t/3},$$

and time $t = 0$ to $t = 30$ using 1000 steps.

I obtained a plot of all particles being on the equatorial line of the Bloch sphere, which means that they were equally distributed in the two wells. Over time, the distribution spread around the equator according to the chosen values for κ and δ .

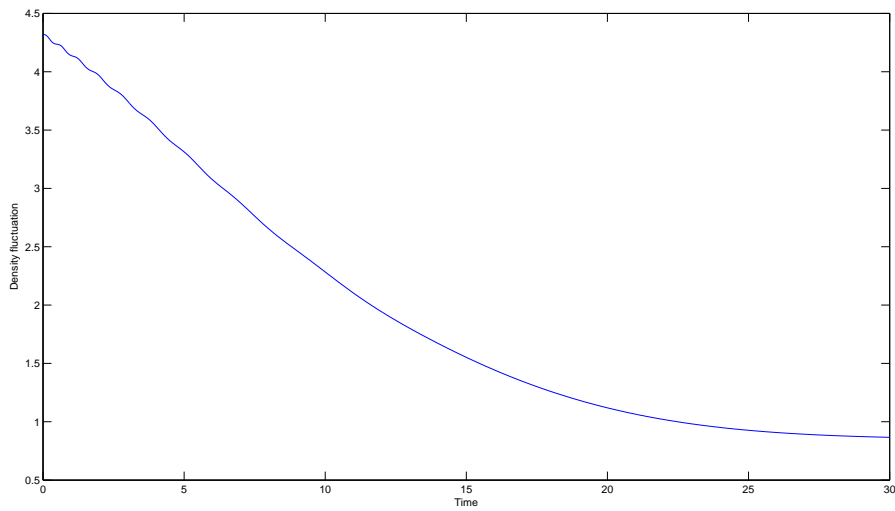


Figure 1: Density fluctuation for the first case $\lambda = 0$

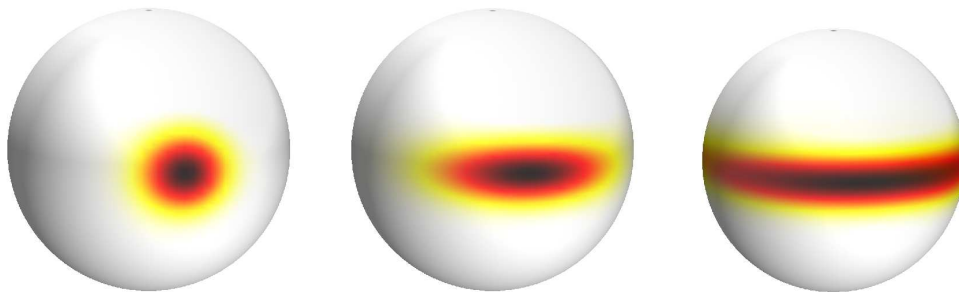


Figure 2: The initial wave function ψ_0 is situated on equatorial line of the Bloch sphere. After 500 time steps it has already broadened a bit, as one can see in the middle figure. ψ at the final time step is depicted on the right. It has nearly equally distributed around the equator of the Bloch sphere.

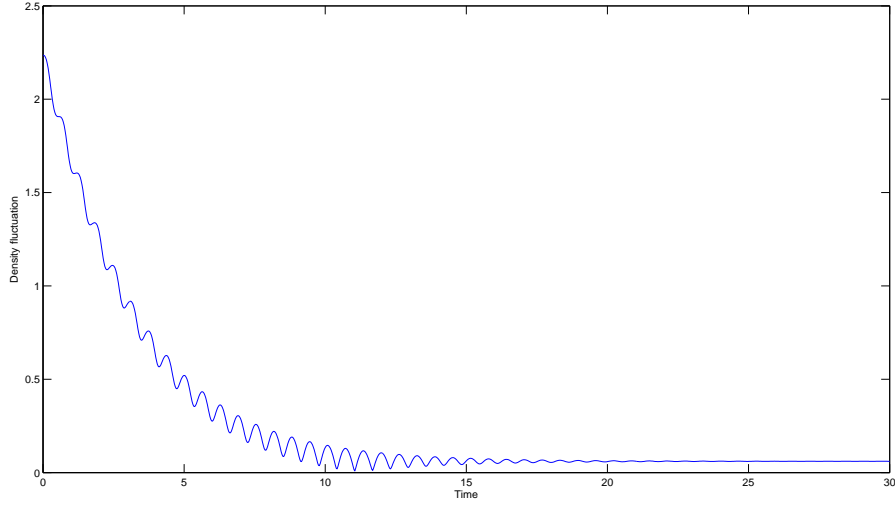


Figure 3: Density fluctuation using a constant $\lambda \neq 0$

To observe the consequences of an imbalance of the wells λ we first investigated the influence of a constant λ to the system.

$$\lambda = 10,$$

$$n = 100,$$

$$\kappa = \frac{1}{n},$$

$$\delta = 10 \cdot e^{-t/3},$$

and time $t = 0$ to $t = 30$ using 1000 steps.

Depending on the value for λ , the initial wave function was positioned along the same meridian, on the upper half sphere for positive values and on the lower half sphere for negative values. Due to λ being a relatively large positive integer in this case the wave function was positioned near the north pole. During evolution of time the particles moved towards the lower well - in case of positive λ towards the northern pole of the Bloch sphere.

The obtained results are shown in figures 3 and 4.

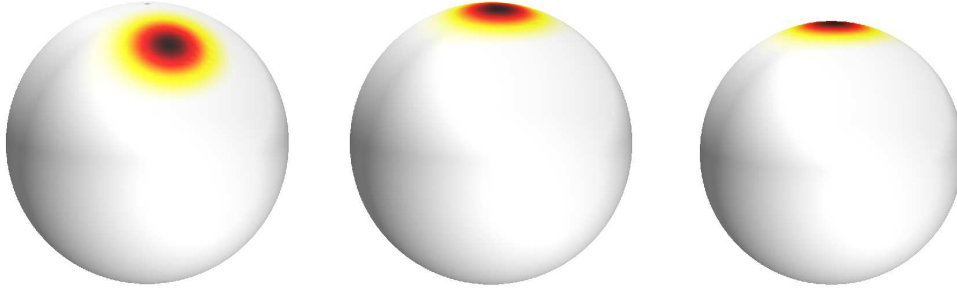


Figure 4: The wave function ψ at the initial time step for a system containing a constant λ is shown left. 150 time steps later, the wave function already nearly reached the north pole. After time step 150 it only circles around the north pole describing smaller and smaller circles until it reaches the position depicted on the right after 1000 time steps.

Next λ was modified, all other parameters were as before.

$$\begin{aligned} \lambda_0 &= 10, \\ \lambda &= \lambda_0 \cdot e^{-0.2 \cdot t}, \\ n &= 100, \\ \kappa &= \frac{1}{n}, \\ \delta &= 10 \cdot e^{-t/3}, \end{aligned}$$

and time $t = 0$ to $t = 30$ using 1000 steps.

Again the initial wave function was positioned on the northern hemisphere of the Bloch sphere near the north pole. Because of the exponential decay of λ and tunnel coupling δ the particles tended to distribute equally during the simulation. Finally the particles distributed around the equatorial line of the Bloch sphere, as one can see on the figures 5 and 6.

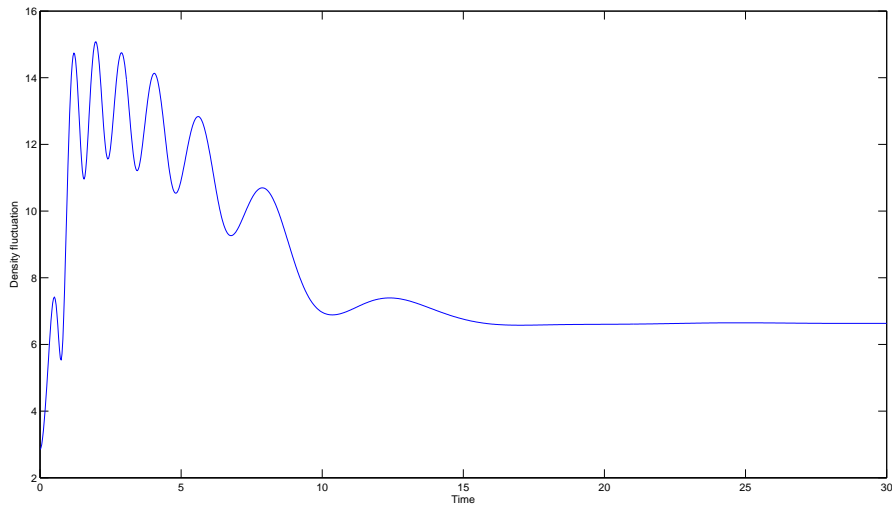


Figure 5: Density fluctuation for the case containing a decreasing λ

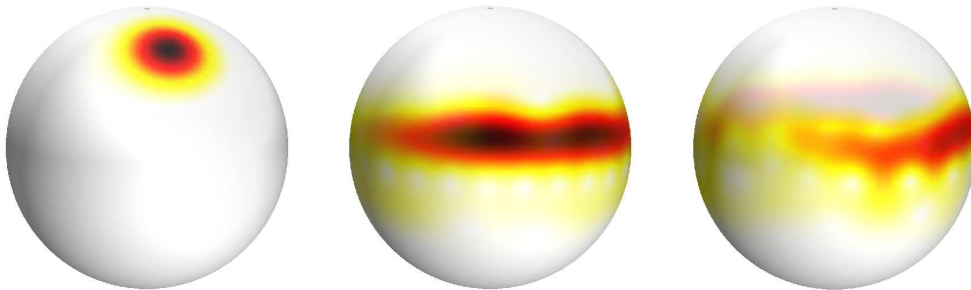


Figure 6: On the left one can see the initial wave function situated on the upper half of the Bloch sphere. The same wave function ψ after 450 time steps is depicted in the middle. It has broadened and lowered towards the equatorial line. ψ at the final time step is shown on the right.

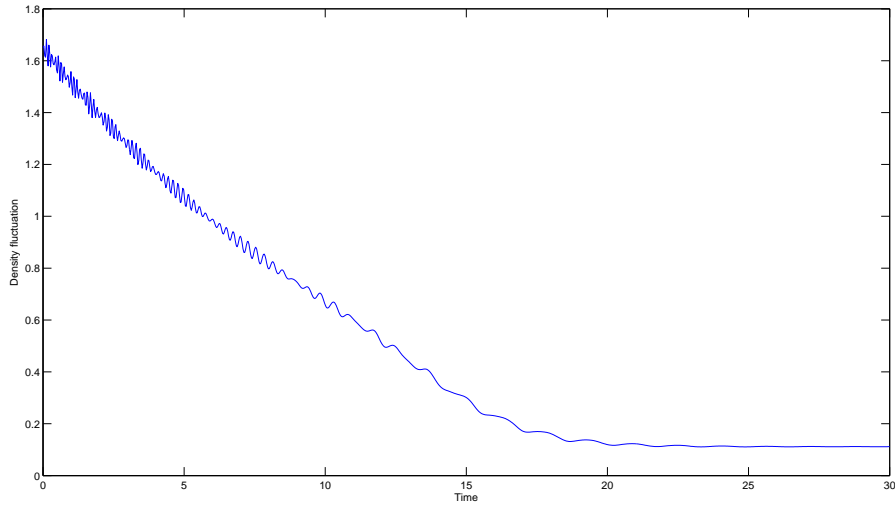


Figure 7: Density fluctuation for $\kappa = 1$

To investigate the influence of the nonlinearity parameter κ it was now modified, all the other parameters stayed the same.

$\kappa = 1, 0.1, 0.001$

One could see that the lower the value for κ the more influence the inequality of the height of both wells had. For $\kappa = 1$ the system nearly behaved as if there would not have been any λ , although it was initialized with 10. During the simulation the wave function spread around the equator and after 1000 time steps it had nearly equally distributed around it.

Regarding the second case, where κ was a tenth, the initial wave function was situated a bit more northern and with decreasing λ it moved towards the equatorial line. Contrary to the case before, now the wave function did not spread equally around the equator after 1000 time steps.

In the third case κ was set to 0.001. The initial wave function ψ_0 was nearly situated on the north pole. Obviously λ had a great influence now. Moreover the wave function never spread around the Bloch sphere in any time step, it only got a little broader. With decreasing λ it even happened to be situated on the southern hemisphere.

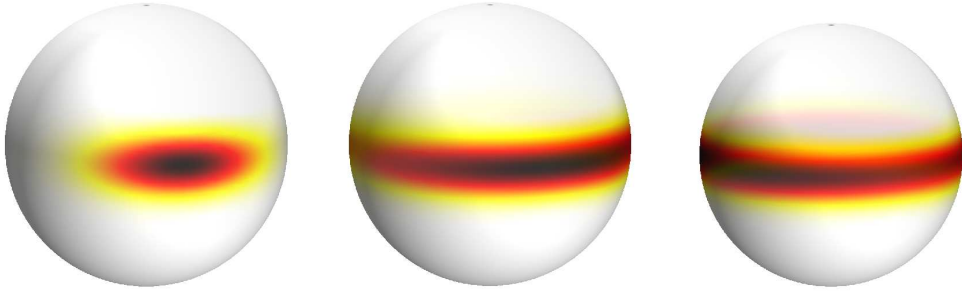


Figure 8: The wave function at initial time for $\kappa = 1$ is depicted on the left side. Compared to the plots before one can see that it already starts near the equator and it is broader already in the initial time step. The same wave function ψ after 450 time steps is shown in the middle. It nearly has equally distributed around the equator after less than half the time. Finally one can see ψ at the final time step on the right figure.

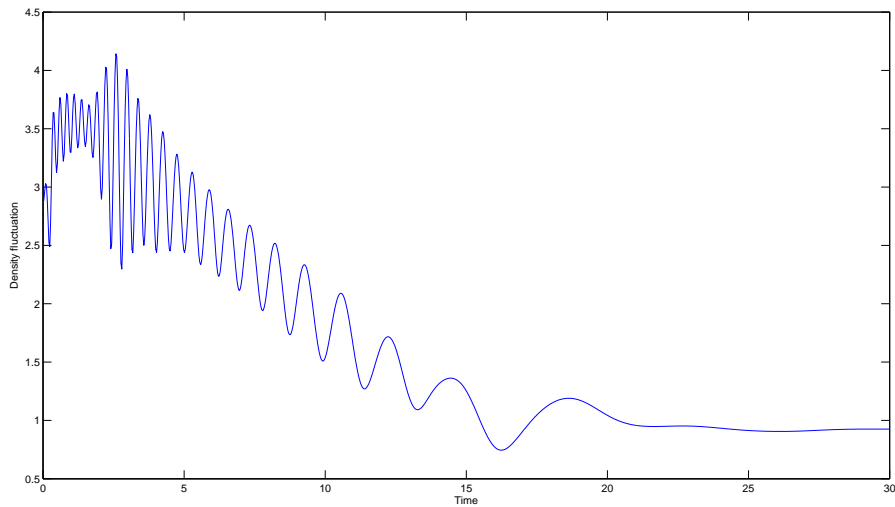


Figure 9: Density fluctuation for $\kappa = 0.1$

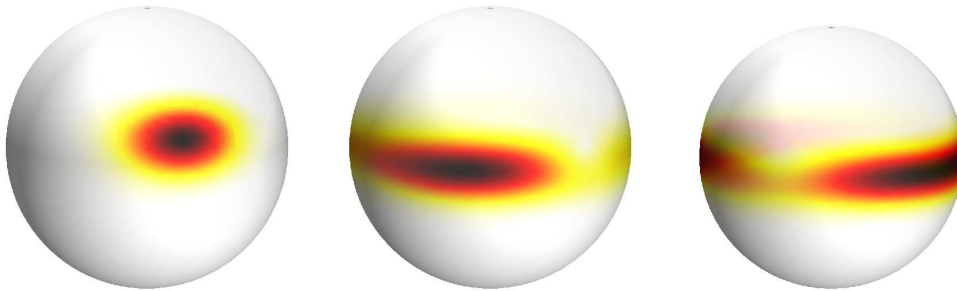


Figure 10: On the first time step ψ now is placed more northern and not as broad as before. After 450 time steps it lowers down to the equator already but still needs longer to distribute around it. Reaching the final time step it still does not distribute completely around the equatorial line as shown in the figure on the right.

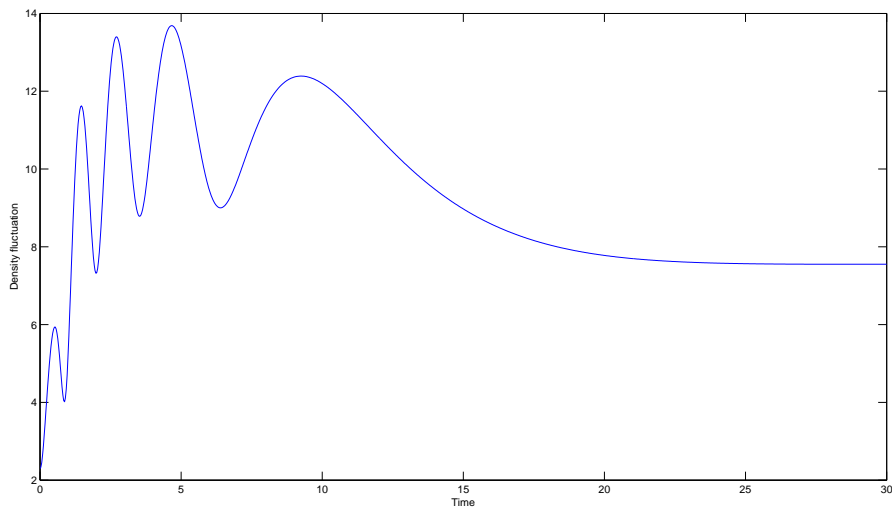


Figure 11: Density fluctuation for the last case $\kappa = 0.001$

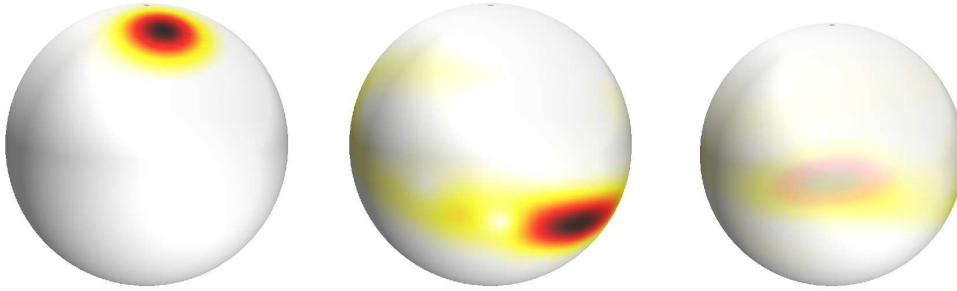


Figure 12: The initial wave function depicted on the left is not broadened now and situated far more northern than the others before. After 500 time steps it lowers even under the equatorial line. But finally the wave function rises back to the equatorial line and concentrates at the back side of the Bloch sphere as can be seen on the right figure.

To simplify the system the nonlinearity parameter κ was turned off for the next investigations. λ oscillated as follows:

$$\lambda_0 = 1, 0.1,$$

$$\lambda = \lambda_0 \cdot \cos(t),$$

$$\kappa = 0,$$

$$n = 100,$$

$$\delta = 10 \cdot e^{-t/3},$$

time $t = 0$ to $t = 30$ using 1000 steps.

Due to the fact that κ was set 0, the wave function never broadened in any time step. According to the chosen value for λ , the initial wave function was situated on the northern hemisphere of the Bloch sphere. The smaller the value for λ , the smaller the distance to the equatorial line was. When time passed, λ oscillated and the wave function moved according to the actual value for λ to the lower of both wells, describing circles around the x-axis of the sphere.

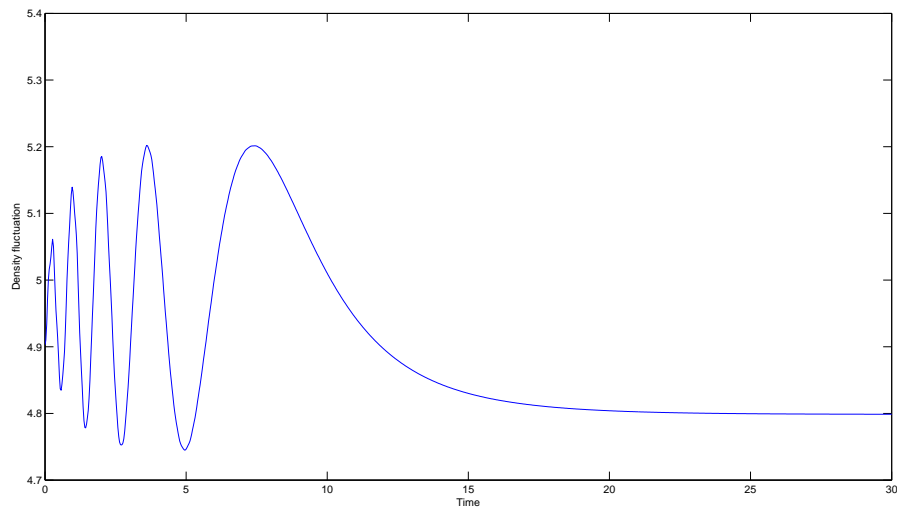


Figure 13: Density fluctuation for the case $\lambda = 1$

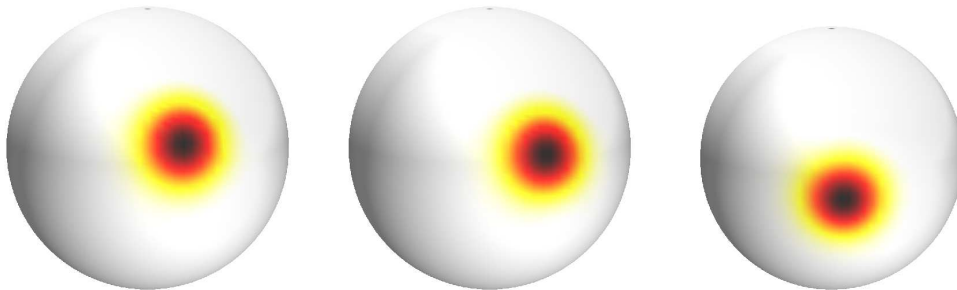


Figure 14: The initial wave function ψ for $\lambda = 1$ is depicted on the left. When the time passes the wave function describes circles around the x-axis, here to be seen at time step 150 in the middle. It continues to circle around the x-axis until it is stopped after 1000 time steps.

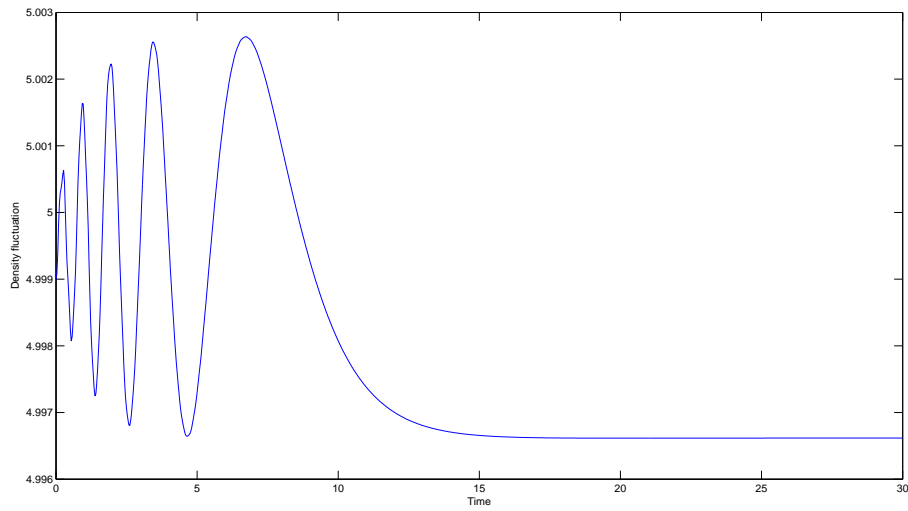


Figure 15: Density fluctuation for the case $\lambda = 0.1$

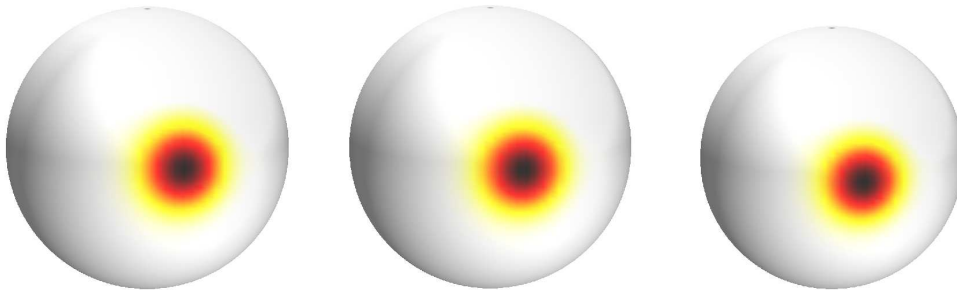


Figure 16: $\lambda = 0.1$. This means that the two wells don't oscillate as much as before. The wave function again describes circles around the x-axis, but the circles are far smaller. Here ψ at time step 250 is depicted in the middle. The circles around the x-axis get smaller and smaller, until it is stopped after 1000 time steps.

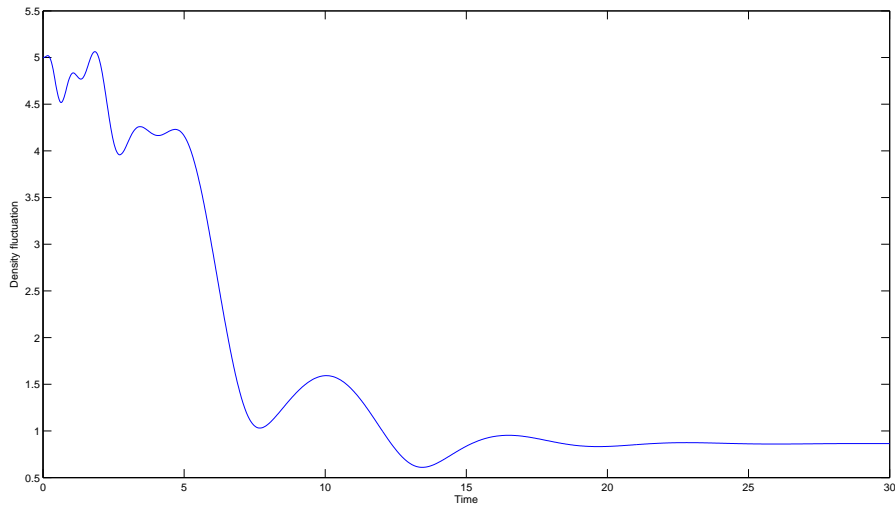


Figure 17: Density fluctuation for the case $\eta = 0.1$

Now λ was given by:

$$\lambda_0 = 1,$$

$$\lambda = \lambda_0 + \eta \cdot \cos(t),$$

$$\kappa = 0, n = 100,$$

$$\delta = 10 \cdot e^{-t/3},$$

time $t = 0$ to $t = 30$ using 1000 steps.

and $\eta = 0.1, 0.01$, which meant that λ oscillated between $\lambda_0 \pm \eta$.

The initial wave function was situated on the equatorial line in both cases. When time passed, the wave function moved towards the north pole describing circles. In the first case it reached the pole earlier and continued to circle around it, in the second case the wave function needed more time to reach the pole.

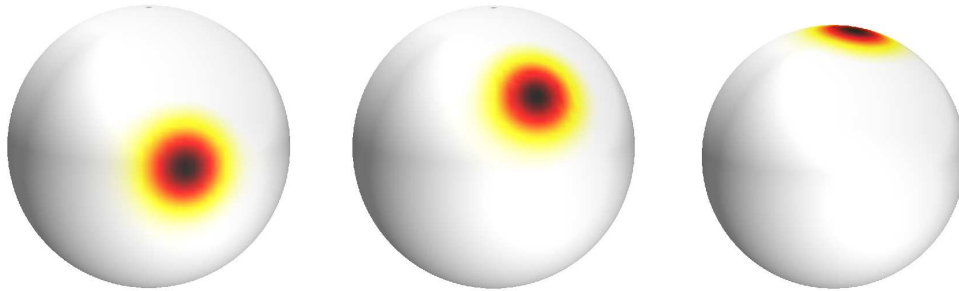


Figure 18: $\eta = 0.1$. The initial wave function shown on the left is positioned at the equator. In the first few time steps the wave function wanders in circles to the northern pole of the Bloch sphere. In the middle the wave function after 200 time steps is depicted. After reaching the north pole, it describes small circles around the pole.

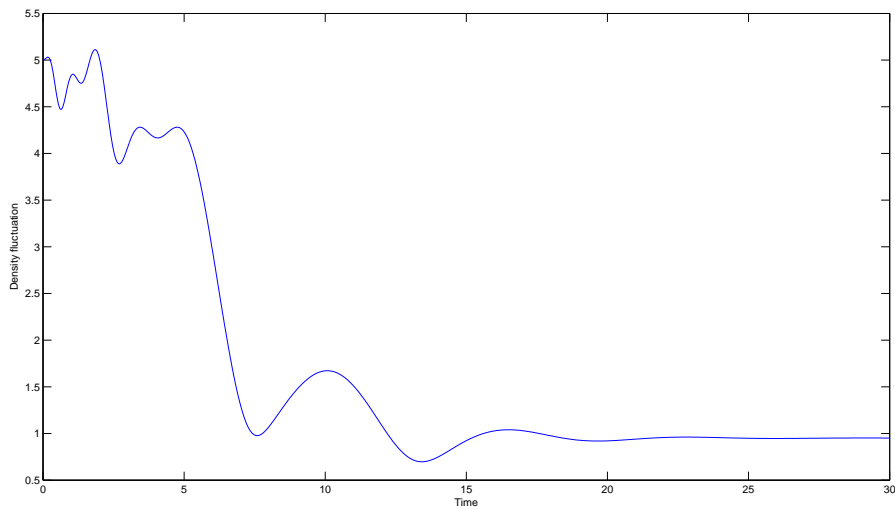


Figure 19: Density fluctuation for the case $\eta = 0.01$

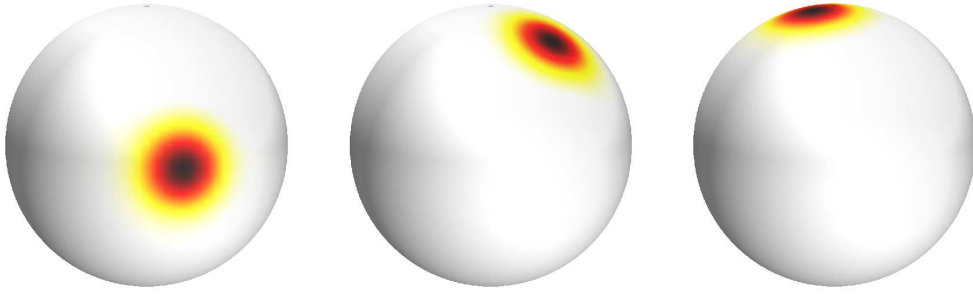


Figure 20: Plots for $\eta = 0.01$. On the left the initial wave function is positioned at the equator again. The wave function again wanders towards the pole in circles, but it takes longer to reach it. After reaching it, the wave function describes very small circles around the north pole.

Finally the solutions of the mean field approach and the two mode approach solving the Schrödinger equation were compared. To simplify the system the tunnel coupling δ was set constant and the nonlinearity parameter κ was set 0. The following parameters were chosen:

$$n = 100,$$

$$\lambda = 1,$$

$$\kappa = 0,$$

$$\delta = 1,$$

$t = 0$ to $t = 30$ using 1000 steps.

According to the nonlinearity parameter being zero the wave function did not broaden. This made a comparison to the calculated mean value easier: if the mean value was situated in the middle of the wave function, both solutions were equal. Due to the chosen values the wave function described a big circle around the x-axis.

During the whole simulation the mean value always stayed in the center of ψ . This led to the conclusion that both approaches for describing such a system were equal.

The results are depicted in figure 21.

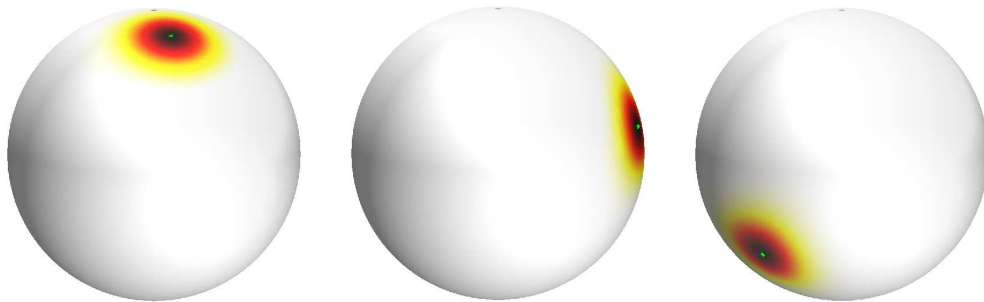


Figure 21: On the left side one can see the initial wave function ψ_0 and the initial mean value plotted as a green star. Both the wave function and the mean value describe a big circle counterclockwise around the x-axis. In the figure in the middle they are shown at time step 750. At the final time step the wave function and the mean calculated by the mean field approach are still equal.

Conclusion

My bachelor thesis is about traps for bosons - the so called Bose-Einstein condensates (BEC) - and the behavior of this system when the trap is changed.

The so-called two-mode model was chosen to describe the system. In this model all bosons could only be in one of two states: either the ground state or the first excited state. All other (higher) states were neglected due to the assumption that the temperature of the system was so low that all other states were rather unlikely. As trap a double well potential was chosen. This potential had two equal wells in which the bosons could freely move around, tunnel from one well to the other, and interfere.

First the normal behavior of the system was investigated, afterwards the potential was modified as follows:

λ was introduced. This value represented the inequality in height of both wells of the potential.

The two mode model was implemented and extended in MATLAB. Basically, the results were obtained by solving the Schrödinger equation using the Crank-Nicholson scheme and plotted onto a Bloch sphere afterwards. If there was an equal amount of particles in both wells, the wave function was situated on the equator of the Bloch sphere. If there were more particles in one of the wells, the wave function was moved towards one of the poles of the Bloch sphere.

The following solutions were obtained:

Introducing a constant λ to the system led to a movement of the wave function according to the chosen value. If λ was turned off during the progress of time, for example by setting to exponential decay, the wave function tended back to the equatorial line of the Bloch sphere.

Using oscillating values for λ like changing it via $\lambda = \lambda_0 \cdot \cos(t)$ led to movements

of the wave function according to the values. It always tended to the lower of both wells. At approximately half-time both wells were nearly filled with the same amount of particles, only small changes could still be determined.

Reducing the oscillations to $\lambda = \lambda_0 + \eta \cdot \cos(t)$ using small values for η , resonance was expected, but did not occur. The wave function moved towards the lower well in circles. Depending on the values for η it reached the pole earlier or later.

Finally the two mode model was solved by the mean field approach and the results were compared. Both solutions were equal - the mean which was calculated by the mean field approach was situated in the center of the wave function in every time step.

Chapter 6

Appendix

The most important code sequences of the program are presented here. The core of the main program: Here all the functions are called and the solutions are plotted afterwards:

```
1 pseudospin('init',n,m);
2 paramtwomode('set','n',n,'kappa',kappa,'tunnelfun',tunnelfun);
3
4 psi0=twomodegroundstate(tunnelfun(tout(1)),lambda);
5 [psit,dn,psisav]=twomodecrank(psi0,tout,lambda);
6 [S,Delta]=meanfield(n,tunnelfun(tout(end)),psi0);
7
8 plotbloch
```

Every 10th time step the wave function ψ is plotted onto the Bloch sphere

```
9 fullscreen=get(0,'ScreenSize');
10 set(gcf,'Position',[0,-50,fullscreen(3),fullscreen(4)])
11 for i=1:10:length(tout)
12     clf
13     figbloch('load','bloch100.mat');
14     figbloch(psisav(:,i));
15     pause(0.01)
16 end
```

The function to calculate the Hamiltonian is as follows:

```
17 function ham=hamtwomode(delta,lambda)
18
19 [j1,j3]=pseudospin(1,3);
20 kappa=paramtwomode('get','kappa');
21 ham=-0.5*delta*j1+4*kappa*j3^2+lambda*j3;
```

The ground state ψ_0 of the wave function is obtained by calling "twomode-groundstate"

```

22 function psi0=twomodegroundstate(delta ,lambda)
23
24 ham=hamtwomode(delta ,lambda);
25 [psi0 ,ene0]=eigs(ham,1 , 'sa' ,struct('disp' ,0));
26 psi0=psi0*sign(sum(psi0));

```

The evolution over time is computed in "twomodecrank":

```

27 function [psi ,dn ,psisav]=twomodecrank(psi0 ,tout ,lambda)
28
29 [kappa ,tunnelfun]=paramtwomode('get' , 'kappa' , 'tunnelfun');
30 one=speye(length(psi0));
31 psit=psi0;
32 dn=zeros(size(tout));
33 dn(1)=numberfluctuation(psi0);
34 psisav=zeros(length(one) ,length(tout));
35 psisav(:,1)=psit;
36
37 for it=2:length(tout)
38     dt=tout(it)-tout(it-1);
39     delta=feval(tunnelfun ,tout(it));
40
41     lambda2=lambda*exp(-tout(it)/5);
42     %lambda2=lambda*cos(tout(it));
43
44     ham=hamtwomode(delta ,lambda2);
45     ham=ham-one*(psit'*ham*psit);
46     psit=(one+0.5i*dt*ham)\((one-0.5i*dt*ham)*psit);
47     dn(it)=numberfluctuation(psit);
48     if(nargout==3)
49         psisav(:,it)=psit;
50     end
51 end
52 psi=psit;

```

The solution for the mean field approach is calculated as follows:

```

54 function [S ,Delta]=meanfield(N ,omega ,psi0)
55
56 kappa=paramtwomode('get' , 'kappa');
57 S0=[0;0;0;0;0;0;0;2/N;2/N;0];
58 for k=1:3
59     S0(k)=real(psi0'*pseudospin(k)*psi0)/(N/2);
60 end
61 tspan=linspace(0 ,30 ,1000);
62 ode=@(t ,S) Sfun(t ,S ,omega ,kappa);
63 [t ,S]=ode45(ode ,tspan ,S0);
64 Delta=S(:,4:end);
65 S=S(:,1:3);

```

with:

```
66 function S=Sfun(t,y,omega,kappa)
67
68 S=[-kappa*y(3)*y(2)-0.5*kappa*y(5); ...
69     omega*y(3)+kappa*y(3)*y(1)+0.5*kappa*y(4); ...
70     -omega*y(2); ...
71     -omega*y(6)-kappa*y(3)*y(5)-kappa*y(2)*y(9); ...
72     omega*(y(9)-y(8))+kappa*y(3)*y(4)+kappa*y(1)*y(9); ...
73     (omega+kappa*y(1))*y(4)-kappa*y(2)*y(5)+kappa*y(3)*(y(7)-y(8)); ...
74     -2*kappa*y(2)*y(4)-2*kappa*y(3)*y(6); ...
75     2*(omega+kappa*y(1))*y(5)+2*kappa*y(3)+y(6); ...
76     -2*omega*y(5)];
```

Chapter 7

Literature

- [1] T. Schumm, J. Schmiedmayer: Materiewellen auf dem Chip. 2009. Physik Journal 8, Nr. 1
- [2] J. Javanainen, M. Y. Ivanov: Splitting a trap containing a Bose-Einstein condensate. 09.1999. Physical Review A Vol. 60, Nr. 3
- [3] A. Vardi, J. R. Anglin: Bose-Einstein Condensates beyond Mean Field Theory. 22.01.2001. Physical Review Letters Vol. 86, Nr. 4
- [4] MATLAB is a registered trademark of The MathWorks, Inc.