

Project 3: Chaotic Behavior in BEC in a Double Well Potential

Quantum Optics Group 3
Hayden McGuinness, Anthony Clark, Anika Pflanzner
Department of physics
University of Oregon
97403, Eugene

14 June 2007

1 Introduction

Bose-Einstein Condensates (BEC) provide relatively clean, fundamentally quantum systems in which to test for chaotic behavior. Condensates can be produced in essentially ideal gases of atoms, and result in a large number of atoms falling into their quantum ground state. Since a large number of atoms occupy a very small volume, interactions are significant, and give rise to an effective nonlinear Schrödinger equation. Since chaotic behavior is known to arise classically in nonlinear systems, it may arise in nonlinear quantum systems. Qualitative examination of the numerically generated energy spectrum and correlation function of the initial and time-evolved states for a condensate in a quartic double-well suggests that chaotic behavior may be present. In order to quantitatively characterise the chaos in the system we choose to use the Lyapunov exponent, which measures the deviation of trajectories for small perturbations in initial conditions.

2 Background

2.1 Bose-Einstein Condensation

A BEC is a phase transition predicted to occur in large systems of identical bosons in which a significant fraction of the particles fall into the same quantum state at a finite temperature. This effect was first predicted from the statistics of identical bosons soon after these were developed by Bose and Einstein. Although superfluid Helium can be described with by modified BEC theory and has long been experimentally realizable, the phenomenon was not observed in an essentially noninteracting boson gas until 1995, due to the difficulty of carrying the necessary cooling. Cornell and Wieman, the first to realize this experimentally used a magneto-optical trapping and evaporative cooling of a system of Rubidium-87 to make the condensate.

The theoretical prediction of BEC formation is as follows. At any temperature T the average number of bosons in the i th energy eigenstate of a confining potential, $\langle n_i \rangle$, is given by

$$\langle n_i \rangle = \frac{1}{e^{\frac{\epsilon_i - \mu(T)}{kT}} - 1}, \quad (1)$$

where $\mu(T)$ is the chemical potential, the energy required to add or remove particles from the system. This distribution depends on the convergence of a geometric series, which requires that $\mu(T) < \epsilon_0$, where ϵ_0 is the lowest energy state of the system for all temperatures. Though this distribution assumes the grand canonical ensemble picture, where particle exchange with the surrounding environment is allowed, for large enough particle numbers these fluctuations are negligible, and so the grand canonical picture gives the same results as the microcanonical picture (which better describes this situation). The chemical potential in this case is thus a free parameter which must be determined by requiring the total particle number to be the actual (constant) particle number:

$$N = \sum_n \frac{1}{e^{\frac{\epsilon_n - \mu(T)}{kT}} - 1} \quad (2)$$

The sum is cannot be evaluated exactly even for simple known energy spectra, so in order to solve for the chemical potential, it must be approximated by an integral. Typically, the integral approximation will work well as long as there is not significant population in the lowest excited states. For systems capable of condensation, the integral will have an upper bound in the limit as $T \rightarrow 0$, and therefore will be unable to

satisfy the above constraint, leading to the conclusion that the lowest levels are significantly occupied and the integral approximation fails.

A simple example of this is a Bose gas in a cubic box of volume V , with periodic boundary conditions. This system has energy spectrum

$$\epsilon_{\vec{j}} = \frac{4\pi^2 \hbar^2 \vec{j}^2}{2mV^{\frac{2}{3}}}, \quad (3)$$

where \vec{j} is a vector whose cartesian coordinates are integers. The sum may be approximated well as an integral in regions where the condition that the distribution function varies slowly over a single j -step is met. As long as the contributions from energy states where the distribution function does not vary smoothly are small, the approximation remains reasonable. The distribution function clearly violates this for the very lowest energy levels.

Making the integral approximation, performing a variable transformations and writing it in spherical coordinates:

$$\frac{N}{V} = \int \frac{d\vec{k}}{2\pi^3} \frac{1}{\exp\left(\frac{\hbar^2 \vec{k}^2}{2m} - \mu(T)\right)\beta} = \left(\frac{2mkT}{\hbar^2}\right)^{\frac{3}{2}} \int_0^\infty \frac{1}{2\pi^2} \frac{r^2}{\exp r^2 - \mu(T)\beta - 1}. \quad (4)$$

The integral can be seen to be convergent for all T , and have an upper bound where μ reaches its maximum value of the ground state energy, which here is 0. Thus, the integral approximation must fail when the system is brought to a temperature for which this upper bound on the number density becomes lower than the actual density of particles in the system, and it is at this critical temperature that the condensation occurs. [3]

Similar analysis can be done for harmonic oscillators and other potentials. In the cases we will be interested in here, we will take the potentials to be asymmetric, so that they are much weaker in one direction than in the other two, so that the system can be effectively treated as one-dimensional.[1] The state of the condensate then refers to the ground state of the 1-dimensional potential in the weak direction. For our purposes, we will consider creating the condensate in such an effectively 1-dimensional harmonic potential, then changing the potential to a quartic double well in this weak direction, so that the condensate wavefunction in the double well shall always begin as the gaussian ground-state of a harmonic potential.

2.2 Particle interactions

Though the atoms may be well approximated to be noninteracting for very dilute vapors above the critical temperature, when condensation occurs and all have the same wavefunction this approximation breaks down, and interaction between atoms becomes significant. A model for the dynamics of the BEC with 2-body interactions is given by the Gross-Pitaevskii equation

$$i\hbar \frac{\partial}{\partial t} \psi = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} \psi + V(x, t)\psi + u |\psi|^2 \psi, \quad (5)$$

where $\psi(x, t)$ is the condensate wave function. The term $u |\psi|^2 \psi$ accounts for the interaction between the particles and “ u ” accounts for the strength.[1] This model is intuitively plausible, as the interactions between particles should depend on the spacing between particles, and there are likely to be more particles found

where the wavefunction squared is highest, and therefore the interactions should be highest since more atoms are likely to be interacting together there. More generally, the Gross-Pitaevski model can be derived as the lowest order term arising from bosonic many-body field theory with a single complex field.[2]

3 Chaos: Classical and Quantum

3.1 Introduction and Definition

In antiquity the word chaos was used to describe events or processes which displayed various forms of randomness and/or disorder. Early scientific studies in the field also pointed to randomness as the defining characteristic of chaos. In 1860 James Clerk Maxwell studied how the collisions between molecules idealized as hard spheres, through progressive amplification of small changes, could lead to microscopic randomness in gases.[4] ¹ In popular terminology chaos can still have that meaning, for example, the first definition of chaos on dictionary.com is “a state of utter confusion or disorder; a total lack of organization or order.” But the scientific study of chaos has over time has come to imply a particular type of order, and one that at least classically, is completely deterministic.

There seems to no canonical definition of chaos but it is generally agreed [5] [6] that systems which exhibit chaos generally have three properties in common:

- Topological transitivity, which essentially means that neighborhoods of points, on some type of trajectory space, eventually disperse so they don't necessarily stick together in one localized clump over their evolution.
- Periodic points of the system must form a dense subset of the space in which the system is defined.
- Sensitivity to initial conditions, which some authors believe may in fact be a consequence of the previous two conditions[7]

In this report we will focus on the latter condition, sensitivity on initial conditions, as the measure of chaos in the BEC systems we study. This condition was chosen because sensitivity to initial conditions is often lauded as the hallmark of a chaotic system and because it was the most accessible measure for our systems.

Sensitivity to initial conditions means that each point in such a system is arbitrarily closely approximated by other points with “significantly” different future trajectories. Thus an arbitrarily small perturbation of the current trajectory may lead to significantly different future behavior. But not all systems which exhibit this condition are thought of as chaotic. Take the the system which evolves along the real line by doubling its current value (the so called $x \rightarrow 2x$ system). A small difference in initial conditions eventually leads to a large difference in trajectory, but because its behavior is extremely simple (for example all points between zero and infinity go to infinity), it is not considered chaotic. It is usually the case that systems must be bounded in their trajectory space to be considered chaotic. [6] Even bound systems can not always be considered chaotic. Take the system of a torus described by two angles (x, y) , each ranging between zero and two. Consider a map which takes (x, y) to $(2x, y + a)$ where a is an irrational number. This map exhibits sensitivity to initial conditions, but because of the irrational rotation in the second coordinate there are no periodic orbits, and hence the map is not chaotic from the definitions formerly given.

¹Interestingly, Maxwell later suggested that mechanical instability and amplification of small changes in the human body could explain free will.

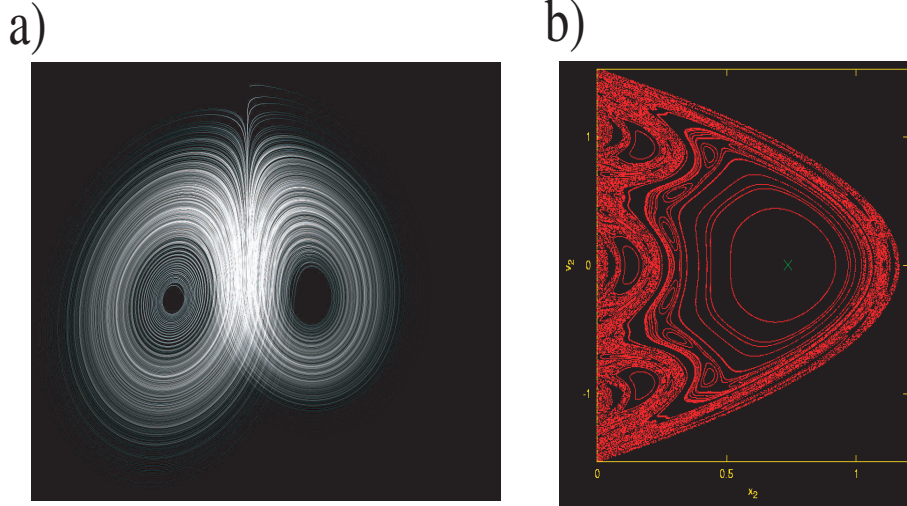


Figure 1: **a)** The Lorenz Attractor system, which exhibits high sensitivity to initial conditions. **b)** A Poincare map of part of the phase space of the Bouncing ball map. As can be imagined from viewing, this map exhibits dense periodic orbits.

3.2 The Lyapunov Exponent

Generally the most used measure of sensitivity to initial conditions is a system's characterization by the Lyapunov exponent, which quantifies the rate of separation of infinitesimally close trajectories. For example, consider a one dimensional system with two trajectories $x_1(t)$ and $x_2(t)$ which at some point t_0 are arbitrarily close together, and their difference in time tracked by the function $\delta x(t) = |x_1(t) - x_2(t)|$. For a chaotic system which can be characterized by a Lyapunov exponent $\delta x(t)$ would go as

$$\delta x(t) \approx e^{\lambda t} |\delta x(t_0)| \quad (6)$$

where λ is the Lyapunov exponent. The Lyapunov exponent is formally defined for such a system as

$$\lambda = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \left(\frac{\delta x(t)}{\delta x(t_0)} \right). \quad (7)$$

The sign, or lack thereof, of the Lyapunov exponent characterizes whether or not the system is exhibiting chaotic behavior. If the exponent is negative the system, at least in that set of initial conditions, is said to be stable (like trajectories go to like trajectories). A Lyapunov exponent of zero implies an unstable system, which is essentially on the edge of stable and chaotic. And of course a positive exponent implies the system is chaotic, where trajectories exhibit exponential divergence.

For multidimensional system there can be a Lyapunov exponent for each degree of freedom. Such a set is called the Lyapunov spectrum. From this there is also what is called the Maximal Lyapunov exponent, which has the formal definition

$$\lambda = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \left(\frac{|\overrightarrow{\delta x(t)}|}{|\overrightarrow{\delta x(t_0)}|} \right). \quad (8)$$

The Maximal Lyapunov exponent is particularly useful because it determines the predictability of the system, the degree to which a systems state can be correctly forecasted for its future evolution.

3.3 Quantum Chaos

Chaos in classical dynamics, if not uniquely defined, is at least well understood on fundamental theoretical grounds. The same cannot be said to be true for fundamentally quantum systems. Even the term “Quantum Chaos” is somewhat controversial, with opponents claiming the idea that quantum systems can exhibit chaos similar to classical systems is simply incorrect.

The controversy arises from the notion that quantum mechanics is a purely linear theory in the state of the system, the state equation of course being

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = H|\psi\rangle \quad (9)$$

which implies the evolution of the state is unitary. Because of this linearity and unitarity of the evolution, it is argued, small perturbations in initial conditions can only lead to small perturbations in latter states. Without the characteristic exponential growth in evolved trajectories it is hard to see how there can be chaos.

Ironically, this line of argument which stems from an orthodox interpretation of quantum mechanics presents a peculiar challenge to the orthodox interpretation. Through the correspondence principle it is expected that the behavior of a large ensemble of quantum systems should reduce to the behavior of the corresponding classical system, hence a classical system which exhibits chaos should fundamentally stem from an inherently quantum system. But if there can be no chaos in quantum mechanics how is this transition to chaos supposed to occur?

Although complete consensus on this and similar questions concerning quantum chaos has not been achieved there are a few possible explanations and reasons why one might expect to see chaos in quantum systems. One is that the dynamics might actually be described by a nonlinear Schrodinger equation, such as the aforementioned Gross-Pitaevskii equation for BEC. The addition of a nonlinear term allows for the possibility of chaotic behavior. Also, the regular Schrodinger equation does not account for the measurement process, which might introduce the dynamic behavior necessary for chaos.[8] In this report we adopt the former reason, as our system is modelled by the Gross-Pitaevskii equation.

4 Results and method

4.1 Computer Implementations and Measures of Chaotic Behavior

The codes used to compute various quantities of interest were written in Fortran95. The code can be found under the directory

usr/p686g3/projects/project3/official and is attached at the end of the document. The basis of our code was the *spectrumsolver* that has been used for problem 2 the second project.

Starting from there we altered the code such that we obtained the correlation function $\langle\psi(r, 0)|\psi(r, t)\rangle$ and by taking the Fourier transformation we also obtained the energy spectrum.

The **correlation function** was computed at each time step and can be considered as a measure for the movement of the system analogous to trajectories in classical mechanics. Therefore for a chaotic system a slightly different initial condition for two trajectories, e.g. a different position in space or a slightly different form of the potential, should produce correlation functions which diverge. This phenomena was observed qualitatively in our simulation as described below. A quantitative measure of chaos is the Lyapunov factor as described in the previous section. It can be obtained by comparing the difference between two correlation functions over a certain time period. Another measure for chaos, qualitative in nature, is the

energy **spectrum**. We obtain it by taking the Fourier transformation of the correlation function. This can be shown by considering the wave function written as

$$|\psi(r, t)\rangle = \sum_{n=0}^{\infty} c_n |\psi_n\rangle \exp\left(\frac{i}{\hbar} E_n t\right) \quad (10)$$

from which the correlation function is defined as

$$\langle \psi(r, 0) | \psi(r, t) \rangle = \sum_{n=0}^{\infty} |c_n|^2 \exp\left(\frac{i}{\hbar} E_n t\right). \quad (11)$$

Then taking the Fourier transformation leads to

$$FFT(\langle \psi(r, 0) | \psi(r, t) \rangle) = |\psi|^2 \delta(E - E_n) \quad (12)$$

which is a spectrum with peaks at the energy eigenfunctions. In case of chaotic behavior we expect the spectrum to become more continuous which means that more peaks show up while getting closer. We especially expect more peaks at lower energies.[9]

The spectra can only be considered as a qualitative measure for chaos, while the Lyapunov exponents are used to obtain a quantitative result for chaotic behavior. This is done with another algorithm written in Fortran.

A measure of chaos familiar from classical mechanics is the Lyapunov exponent. To calculate it, the program compares two trajectories starting at slightly different initial conditions. In the quantum case, we can actually consider the correlation function as analogous to trajectories. One correlation function is used as the “control function” $\langle \psi_C(0, r) | \psi_C(t, r) \rangle$ while the other correlation function, the “comparator function” is obtained comparing its evolution to the control function’s initial condition, giving it the form $\langle \psi_C(0, r) | \psi(t, r) \rangle$. The initial difference of the two wavefunctions is that the comparator is offset in space by an amount dx as compared to the control.

By definition the Lyapunov exponent is defined as the limit of a quantity as time goes to infinity, but since it is desired to compute the exponent a finite approximation must be made. The correlation functions evolve for a time Δt then their difference is computed and scaled with the initial difference between the functions $\delta(x_0)$ to obtain the stretching factor at a certain time step

$$\alpha_i = \left| \frac{\delta x_i}{\delta x_0} \right| \quad (13)$$

After that the correlation function is reset to the control function and evolves again for a time Δt . This process is implemented n times, so that in the end we obtain

$$\lambda \approx \frac{1}{n\Delta t} \sum_{i=1}^n \ln(\alpha_i). \quad (14)$$

This is the approximate Lyapunov factor which gives a quantitative measure for the chaotic behavior of a system.

4.2 Qualitative Results: Energy Spectra

4.2.1 The Harmonic Oscillator

As the energy spectrum of the harmonic oscillator is well known we first tested the program on a potential of the form

$$V(x) = \omega^2 x^2 \quad (15)$$

such that the Gross Pitaevskii equation becomes

$$i\hbar \frac{\partial}{\partial t} \psi = \left(-\frac{\hbar^2}{2m} + \omega^2 x^2 + u|\psi|^2 \right) \psi \quad (16)$$

First, investigations were made on a **harmonic oscillator without nonlinearity**. We started out with a Gaussian wave packet of the form

$$\psi = \sqrt{\frac{1}{\sigma\sqrt{2\pi}}} e^{-\frac{(x-x_0)^2}{4\sigma^2}} \quad (17)$$

and observe how it evolves. This can be considered a Bose condensate with most the particles in one state and the interaction accounted for in the nonlinear term of the Gross Pitaevskii equation.

For a frequency of π , we obtain the correlation function, as well as the energy spectrum shown in the plots below. This corresponds to our expectations: a spectrum with peaks at the energy eigenfunctions

$$E_n = \frac{1}{2} \hbar \omega (n + 1). \quad (18)$$

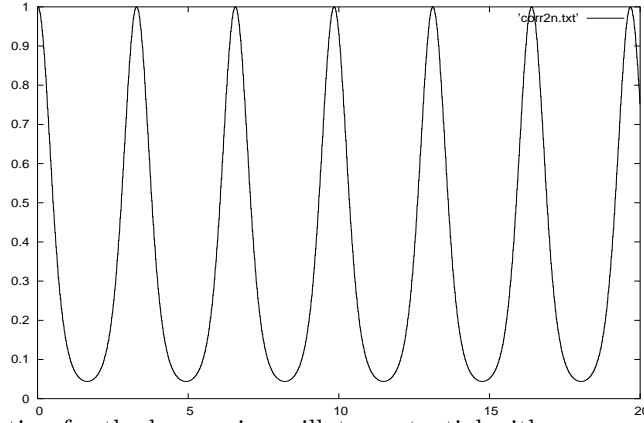


Figure 2: Correlation function for the harmonic oscillator potential with $\omega=\pi$ and initial position $x_0=1.0$

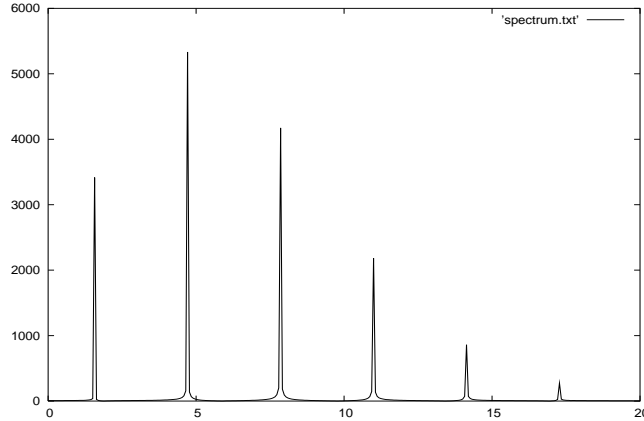


Figure 3: Spectrum for the harmonic oscillator potential with $\omega=\pi$ and an initial x position of $x_0=1.0$, the peaks are at the energy eigenfunctions $E_n = \omega \hbar (n + \frac{1}{2})$

Comparing two correlation functions with slightly different initial position, i.e. at $x_1=0.5$ and $x_2=0.505$ shows that they don't diverge, but that their difference stays the same during several oscillations. This means, from a qualitative point of view, their behavior is not chaotic.

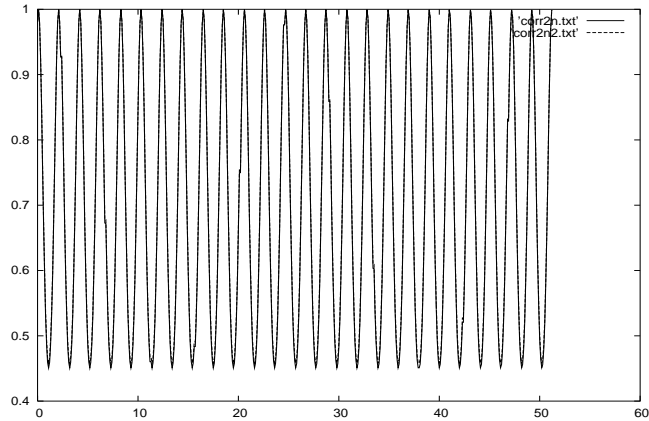


Figure 4: Correlation functions for slightly different initial conditions for the harmonic oscillator potential with $\omega = \pi$

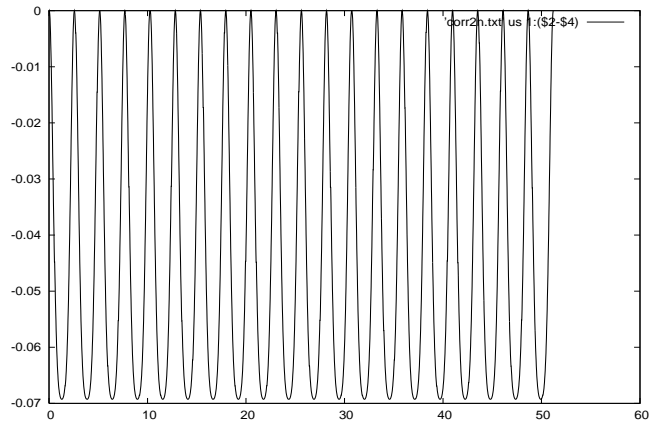


Figure 5: Difference between two correlation functions with slightly initial conditions in a harmonic oscillator potential with $\omega = \pi$

Considering these results we see that the harmonic oscillator without a nonlinear term behaves as a non chaotic system, which was strongly expected.

We want to compare these results to the **harmonic oscillator with nonlinearity**. For the following investigations, the nonlinear term was chosen to be $u=10.0$. Two different correlations were compared with initial conditions $x_{01} = 0.5$ and $x_{02} = 0.51$. The following plots show how the correlation functions diverge and the increasing of the difference between them.

Comparing these results to the harmonic oscillator potential without nonlinearity, it is obvious that the difference in the correlation functions actually increases. This means in some sense that we can see chaotic behavior, or simply that the spectrum is becoming more complicated. For a more rigorous answer to the chaotic behavior question it is necessary to determine the Lyapunov exponent, which is done in the proceeding section.

4.2.2 The Double Well Potential

The other system that was investigated was a Bose condensate in a double well potential. This was the main focus of our project and this is where we expect actual chaotic behavior. Again we consider the correlation functions as well as the spectra to get a qualitative measure of the chaotic nature of the system. Again two

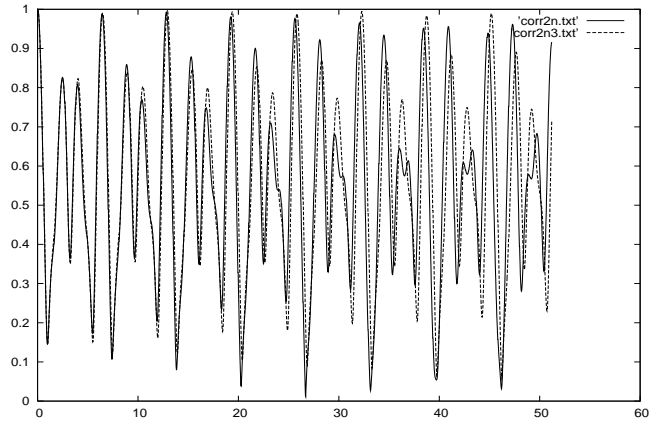


Figure 6: Correlation functions for slightly different initial conditions for the harmonic oscillator potential with a nonlinear term of $u=10.0$ and a frequency of $\omega=\pi$

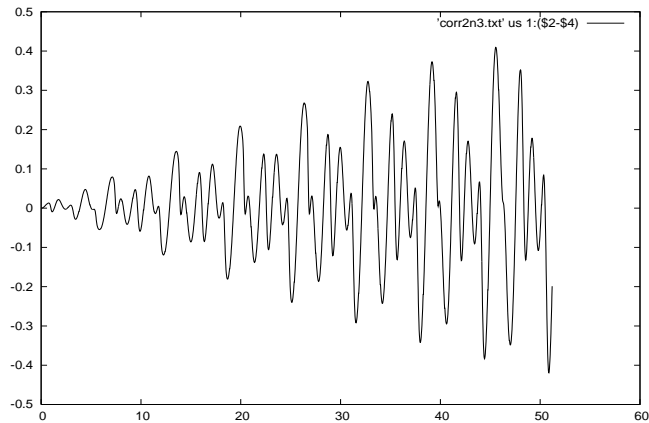


Figure 7: Difference between the correlation functions for slightly different initial conditions for the harmonic oscillator potential with a nonlinear term of $u=10.0$ and a frequency of $\omega=\pi$

correlation functions starting out at slightly different initial conditions were compared. We consider different nonlinear coupling terms u and want to see where chaotic behavior starts to occur.

The double well potential that was used was of the form

$$V(x) = \omega^2(x^4 - ax^2) \quad (19)$$

where the constants ω and a could be varied in the parameter file.

First we run a simulation for the **double well potential with $u=0$** . The results are similar those of the harmonic oscillator, two correlation functions with slightly different initial conditions don't diverge but obtain their difference. The following plots are for two different wave functions in a potential with $\omega = \pi$, $a = 0.03$, starting out at $x_{01}=0.2$ and $x_{02}=0.205$.

The double well potential without nonlinearity therefore doesn't show any form of chaotic behavior.

Proceeding with a **double well potential with a nonlinear term of $u=10$** , again we compare the correlation functions of Gaussians starting off at $x_{01}=0.2$ and $x_{02}=0.205$ with a nonlinear term of $u=10.0$ and the same double well potential as before. Qualitatively examining the result for the correlation functions,

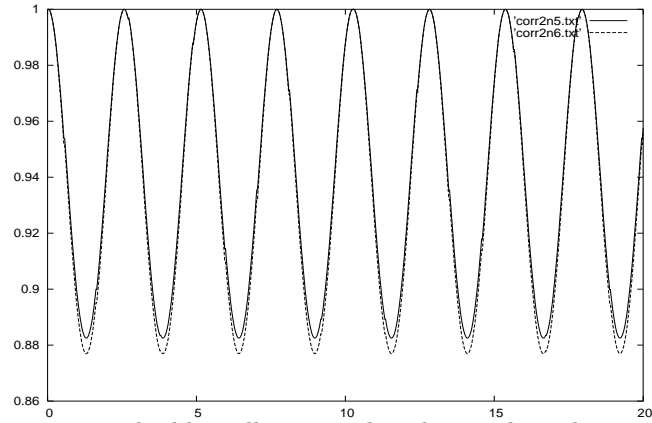


Figure 8: Correlation functions in a double well potential with initial conditions $x_01=0.2$ and $x_02=0.205$, $\omega = \pi$, $a = 0.03$ and $u=0$

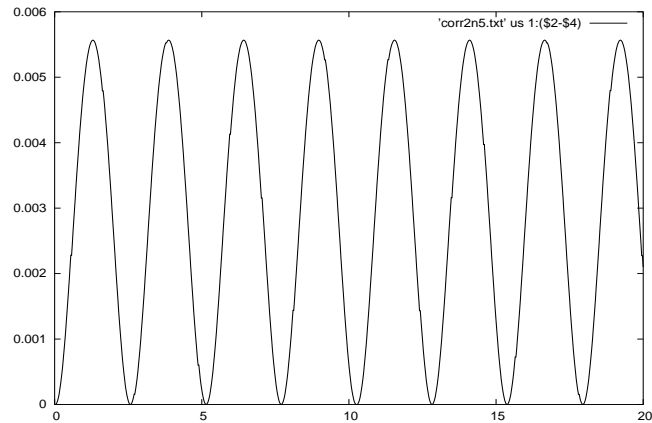


Figure 9: Difference between the correlation functions in a double well potential with initial conditions $x_01=0.2$ and $x_02=0.205$, $\omega = \pi$, $a = 0.03$ and $u=0$

their behavior looks chaotic:

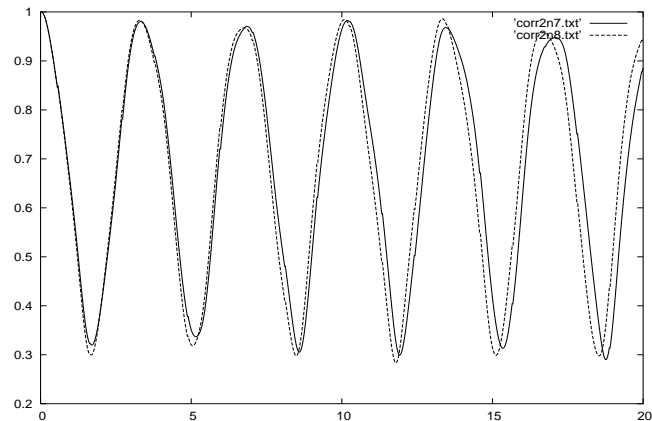


Figure 10: Difference between the correlation functions in a double well potential with initial conditions $x_01=0.2$ and $x_02=0.205$, $\omega = \pi$, $a = 0.03$ and $u=10.0$

Compared to the double well potential with a nonlinear term, a divergence of the two correlation functions was observed.

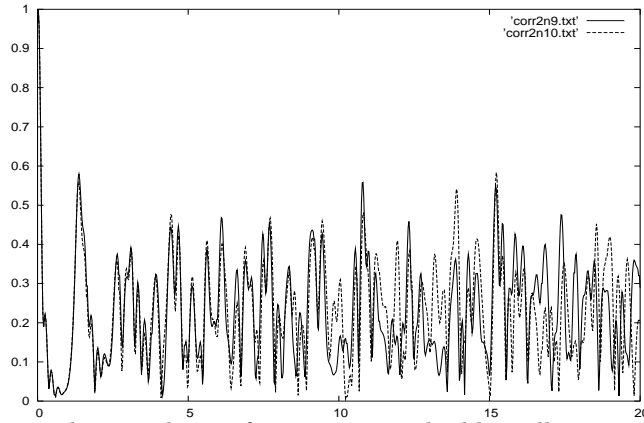


Figure 11: Difference between the correlation functions in a double well potential with initial conditions $x_{01}=0.2$ and $x_{02}=0.205$, $\omega = \pi$, $a = 0.03$ and $u=30.0$

If the nonlinearity is raised to $u=30$ the correlation functions start to drift apart even more.

We consider the spectra, which are another qualitative measure for chaotic behavior, which we expect to see become more continuous, i. e. to have more peaks, in a chaotic regime of the system. Especially the number of peaks at low energies should increase for a chaotic system. This is because aperiodic points in a finite data set appear as points with long periods and therefore in the spectrum as points at low energies.[9] The following plots show a comparison of spectra for three different nonlinear terms, $u=0$, $u=30$ and $u=100$. All the plots are on a logarithmic scale. Comparing the spectra for increasing nonlinear terms, they become noisier in the low frequency terms, which fulfills our expectation about an increasing chaotic behavior for an increasing value of the nonlinearity u .

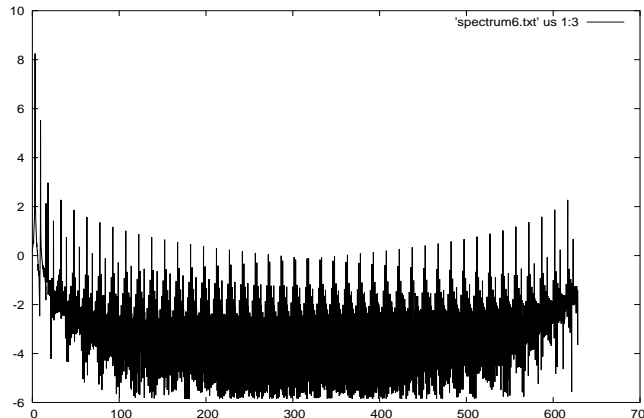


Figure 12: Spectrum of the double well potential for $\omega = \pi$, $a = 0.5$ and $u= 0$

The qualitative explorations indicate an increasing chaotic behavior for an increasing nonlinear term.

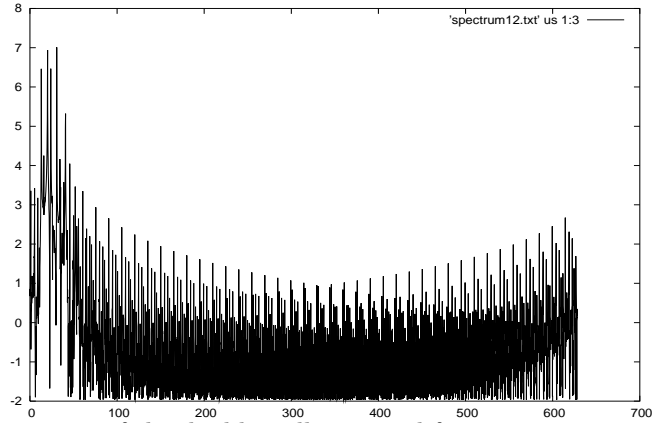


Figure 13: Spectrum of the double well potential for $\omega = \pi$, $a = 0.5$ and $u=30$

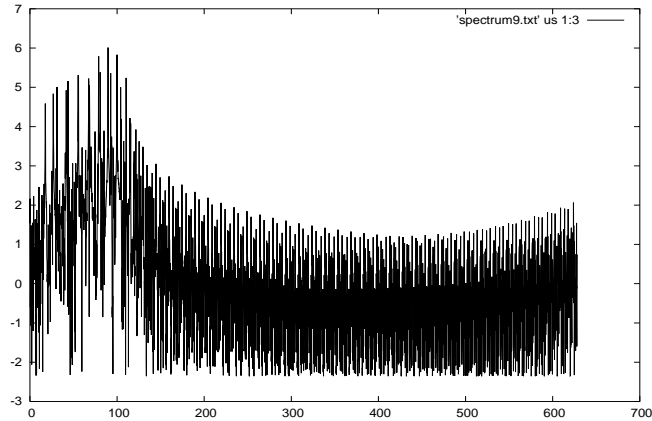


Figure 14: Spectrum of the double well potential for $\omega = \pi$, $a = 0.5$ and $u=100$

4.3 Quantitative Measure of Chaos: the Lyapunov factor

4.3.1 The Harmonic Oscillator

To obtain a quantitative measure of the system the Lyapunov factor was determined with the method described above. The rescaling of the correlation functions was done for every time step as this gives the highest precision. First, we explored **the harmonic oscillator**. The Lyapunov exponent λ was determined for four different final times. The results are presented in the table below.

u	tfinal	λ	x
0	100	0.3426	0.71
0	200	-0.3294	0.71
0	300	-0.3296	0.71
0	400	-0.3300	0.71
1	100	0.8230	0.71
1	1000	-0.0028	0.71

Table 1: Lyapunov exponents for the harmonic oscillator for different nonlinear terms and different time step

The exponent λ as a measure of chaos should be negative for a stable system and positive for a chaotic system. As it is visible in the data, large final times are necessary to obtain converging results. The Lyapunov factor

is negative for a nonlinear term $u=0$ and eventually converges within a time of 400. Therefore the harmonic oscillator system without nonlinearity is stable, which corresponds to our expectations.

Furthermore, for a nonlinear term of $u=1$, we still obtain a stable system although it is harder to see convergence in the Lyapunov factor as the necessary running times become very high. Still, as a general result, we state that we do not observe chaotic behavior for the harmonic oscillator potential.

4.3.2 The Double Well Potential

The main focus of this project is on the double well potential. We searched for chaotic behavior at two different initial position. The parameters that were hold constant throughout all the runs are $N_x=1024$, $dx=0,01$, $dt=0.001$, $a=0.5$ and $\omega = \pi$. First, we start off at an initial position of $x_0=0.75$, which is slightly above the energy of the saddle point. At this point we expect chaotic behavior as the energy of the Bose condensate is high enough.

Our result is in agreement with this expectation: for a nonlinearity of zero, the system stays stable with a negative Lyapunov factor. If the nonlinear term is increased to $u=10$, we obtain a positive Lyapunov factor, which indicates chaos. We don't see the convergence of the Lyapunov factors yet, longer running times would be necessary to obtain more reliable results, but due to time limitations on this project we were not able to pursue convergence satisfactorily. Although we didn't find sufficient convergence λ seemed to stabilize for nonlinear evolution to some degree for a final time of around 1000, hence the time 1000 was taken to be a sufficient time in which to find an adequate approximation to λ . Because of this all further final times for the double well were taken to 1000.

u	tfinal	λ	x
0	1000	-0.4180	0.75
10	100	1.6830	0.75
10	200	0.2693	0.75
10	400	0.8560	0.75
10	800	1.1969	0.75
10	1200	0.8000	0.75

Table 2: Lyapunov exponents for the double well potential for different nonlinear terms and different time steps

The next investigation was made with the same parameters as above, but started at $x_0=0.65$ which is below the energy of the saddle point. This is an interesting point to start, because it is not quite clear whether we should expect chaos or not, as we do have a nonlinear term, but the initial energy of the condensate might be too small to evolve chaotically.

u	tfinal	λ	x
0	1000	-0.206	0.65
10	1000	0.4704	0.65

Table 3: Lyapunov exponents for the double well potential for different nonlinear terms and different time steps

The data1 obtained indicated that the system is also chaotic for this initial condition when nonlinearity is present. The Lyapunov factor is smaller than for the starting point at 0.75, but still doesn't become negative for a large final time $t_{final}=1000$. Therefore, we conclude that chaos occurs in this regime as well. As expected, when no nonlinear is present λ is negative, signifying a lack of chaotic behavior.

5 Conclusions

Summing up, our results are in good accordance with our expectations. For the **harmonic oscillator** we did not expect any chaos. Our results for the Lyapunov exponent for $u=0$ did converge well in the chosen time range and are therefore reliable. Also for a nonlinear term $u=1$, our results for the Lyapunov factor converged. Therefore, the conclusion that we don't observe chaotic behavior in this potential is reliable.

For the double well potential it was harder to get convergence in the Lyapunov exponents. The running times in this regime are a lot higher and our data is therefore less reliable. More runs at much higher final times would be needed, but as time is limited...

Still, the result we obtained for running times of $t_{\text{final}}=1000$ (as our previous observations indicated that this was the order of magnitude where λ would converge) are within our range of expectations. For an initial point $x_0=0.75$ of the wave packet, which is slightly above the saddle point energy of the double well potential, the system remained stable for $u=0$, while we got clear indicators of chaotic behavior, a Lyapunov factor larger than zero for $u=10$, for a nonlinear term unequal to zero. The other interesting result we obtained is that the condensate's energy does not necessarily have to be above the saddle point in order to lead to chaos: we observed a positive Lyapunov factor for $x_0=0.65$ and $u=10$.

Chaotic behavior in Bose Einstein condensates therefore occurs in double well potentials both for energies above and below the saddle point energy.

A Appendix A: Used code

The code use for this projects was based on the project 2, problem 2 code. We only included the code that has been changed for this project.

```
=====
!
! specsol.f90
!
! Computes evolution of Schrodinger equation in one dimension for two
! wavefuntions with differing initial condition by split-operator
! exponentials, generates a correlation function
! with the initial wavefunction at each output time step, then
! uses this to construct the energy eigenspectrum of this
!
!
! The equation can be found in the file splitop_evolver.f90
!
! Evolution is by explicit split-operator
! exponentials, to give a second order method extendable to
! 4th, 6th, etc orders.
!
! Main output is to 2 text files:
! 1. corr2n.txt
! first column is the time, the other column is the correlation
! value for the first wavefunction
! 1. corr2n2.txt
! first column is the time, the other column is the correlation
! value for the second wavefunction
!
!=====

program specsol

  use globals
  use output
  use param_parser
  use splitop_evolver
  use timing
  use spline

  implicit none

  integer :: j, jj, q, qq, ll
  integer :: ti, tiout
  integer :: power
  real(vp) :: stretchsum

  !array of stretch parameters
  real(vp), allocatable, dimension(:) :: stretch
  real(vp) :: fircorr, lacorr, fircorr2, lacorr2, lambda

  !declares storage arrays for the initial wavefunction, the correlation
  !function, and the energy spectrum
  complex(vp), allocatable, dimension(:) :: psinaught, corr, &
  spectrum, corr2n
  complex(vp), allocatable, dimension(:) :: psinaught2, corr2, &
  spectrum2, corr2n2
  real(vp), allocatable, dimension(:) :: tgrid, tgrid2n, corr2nre, corr2nim
  real(vp), allocatable, dimension(:) :: tgrid2, tgrid2n2, corr2nre2, corr2nim2

  ! get parameters from parameter file
  call parse_params()

  allocate(psinaught(Nx))
  allocate(psinaught2(Nx))

  !the stretch factors array
  allocate(stretch(int(tfinal/dt) +1) )

  stretch = 1.0

  !set initial conditions
  psinaught = exp( -(x-x0)**2/(4*sigma_x**2) ) / sqrt(sqrt(2*pi)*sigma_x)
  psinaught2 = exp( -(x-(x0+dx))**2/(4*sigma_x**2) ) / sqrt(sqrt(2*pi)*sigma_x)

  psi = psinaught
  psi2 = psinaught2

  ! output initial condition
  call open_output_files()
  call output_stuff()
  call output_dists(first_time=.true.)

  call print_norm()
  if ( time_trace ) call print_time('s')

  !allocate correlation function array
  !because it needs to be a power of 2 for the fft routine to be used on it
  !must make sure that it gets padded with enough extra zeros to make it work
  ! or interpolation may be used to make the array the proper size

  allocate(corr(2*Ntout+1))
  allocate(corr2(2*Ntout+1))

  power = Int(log(Real(2*Ntout+1))/log(2.0)) + 1
```



```

!!allocate the time grid to feed the spline function
!! and the new power-of-two sized grid well we're at it

allocate(tgrid(2*Ntout+1))
allocate(tgrid2n(int(2**power)))
allocate(corr2n(int(2**power)))
allocate(corr2nre(int(2**power)))
allocate(corr2nim(int(2**power)))

allocate(tgrid2(2*Ntout+1))
allocate(tgrid2n2(int(2**power)))
allocate(corr2n2(int(2**power)))
allocate(corr2nre2(int(2**power)))
allocate(corr2nim2(int(2**power)))

! Main evolution loop
t = 0.0_wp
j = 1
jj = 0

!first correlation function values, used for calculating the initial stretch factor
fircorr = dot_product(conjg(psi), psinaught)*dx
fircorr2 = dot_product(conjg(psi2), psinaught)*dx

!set and write the initial value of the correlation function
corr(int(0.5*(size(corr)+1))) = fircorr
corr(int(0.5*(size(corr)+1))) = fircorr2

do tiout = 1, Ntout ! this is the loop over output times
do ti = 1, toutevery ! this is a loop over times between outputs
t = t + dt
j = j + 1
!call splitop_onestep() ! this evolves psi from t to t+dt

call splitop_ord2_onestep(t, dt, psi)
call splitop_ord2_onestep(t, dt, psi2)

!rescale on every single time step.

jj = jj + 1

lacorr=dot_product(conjg(psi), psinaught)*dx
lacorr2=dot_product(conjg(psi2), psinaught)*dx

stretch(jj) = abs( (lacorr - lacorr2)/(fircorr-fircorr2) )

psi2=cshift(psi,-1)

fircorr = dot_product(conjg(psi), psinaught)*dx
fircorr2 = dot_product(conjg(psi2), psinaught)*dx

end do

if (mod(tiout,100).eq.0) write(0,*) 100.0*tiout/Ntout, ' % done '

!build correlation function value at each time step, store in corr array
corr(int(0.5*(size(corr)+1))+tiout) = dot_product(conjg(psi), psinaught)*dx
corr(int(0.5*(size(corr)+1))-tiout) = conjg(corr(int(0.5*(size(corr)+1))+tiout))

corr2(int(0.5*(size(corr2)+1))+tiout) = dot_product(conjg(psi2), psinaught)*dx
corr2(int(0.5*(size(corr2)+1))-tiout) = conjg(corr2(int(0.5*(size(corr2)+1))+tiout))

! note: for a stochastic code, you may want to make many more
! calls to output_stuff() than to output_dists()
call output_stuff() ! this writes expectation values, etc
call output_dists() ! this writes wave functions; *always* call
! *after* output_stuff()

!!!commented out this for efficiency
! if ( time_trace ) then
! write(0,*) 'Wave function computed at t = ', t
! write(0,*) ' (step ', tiout, ' of ', Ntout, ' )'
! end if
! call print_norm() ! print wave function norm to std err
! if ( time_trace ) call print_time('m') ! print elapsed time to std err
end do
! end main evolution loop

!fill tgrid and tgrid2n
do qq = 1, size(tgrid)
tgrid(qq) = qq
end do
do ll = 1, size(tgrid2n)
tgrid2n(ll) = 1+(ll-1)*size(tgrid)/size(tgrid2n)
end do

do qq = 1, size(tgrid2)
tgrid2(qq) = qq
end do
do ll = 1, size(tgrid2n2)
tgrid2n2(ll) = 1+(ll-1)*size(tgrid2)/size(tgrid2n2)
end do

!use the spline interpolation routine to get the correlation function on
!an array of the needed power-of-two length
!call cubic_spline(real(tgrid), real(corr), real(tgrid2n))
corr2nre = cubic_spline(tgrid, real(corr), tgrid2n)
!call cubic_spline(real(tgrid), aimag(corr), real(tgrid2n))

```

```

corr2nim = cubic_spline(tgrid, aimag(corr), tgrid2n)
corr2n = corr2nre + i*corr2nim
!use the spline interpolation routine to get the correlation function on
!an array of the needed power-of-two length
!call cubic_spline(real(tgrid), real(corr), real(tgrid2n))
corr2nre2 = cubic_spline(tgrid2, real(corr2), tgrid2n2)
!call cubic_spline(real(tgrid), aimag(corr), real(tgrid2n))
corr2nim2 = cubic_spline(tgrid2, aimag(corr2), tgrid2n2)
corr2n2 = corr2nre2 + i*corr2nim2

!The exponent!
stretchsum=0.0

write(0,*) jj, 'is the # of stretch factors'
do j=1, jj
!write(0,*) stretch(j), j
stretchsum = stretchsum + log(stretch(j))
end do

lambda=1/(jj*dt)*stretchsum

write(0,*) 'Lyapunov exponent is... ',lambda

!write correlation functions
open(unit=63, file='corr2n.txt', status='unknown')

do q=int(0.5*size(corr2n)), size(corr2n)-1
write(63,*) (q-int(0.5*size(corr2n)))*tfinal/Ntout, abs(corr2n(q+1))
end do

open(unit=64, file='corr2n2.txt', status='unknown')

do q=int(0.5*size(corr2n2)), size(corr2n2)-1
write(64,*) (q-int(0.5*size(corr2n2)))*tfinal/Ntout, abs(corr2n2(q+1))
end do

close(unit=63)
close(unit=64)

allocate(spectrum(size(corr)))
allocate(spectrum2(size(corr2)))

spectrum = fftsf(1,corr2n)
spectrum2 = fftsf(1,corr2n2)

!write spectrums
open(unit=15, file='spectrum.txt', status='unknown')
do q=0, size(spectrum)-1
write(15,*) 2.0*pi*q*tfinal/Ntout, abs(spectrum(q+1)), log(abs(spectrum(q+1)))
end do

open(unit=16, file='spectrum2.txt', status='unknown')
do q=0, size(spectrum2)-1
write(16,*) 2.0*pi*q*tfinal/Ntout, abs(spectrum2(q+1)), log(abs(spectrum2(q+1)))
end do

! close output files
call cleanup()
end program specsol

=====
!
! splitop_evolver.f90
!
! Routines to evolve the wave function psi by one time step dt
! by split-operator Fourier transforms.
!
=====

module splitop_evolver

use globals
use ffts

implicit none

contains

!-----
!
! subroutine splitop_onestep()
!
! Driver for split operator method. The real work is done in
! splitop_ord2_onestep. This driver is basically a wrapper that
! can combine the single steps to do higher order methods.
!
!-----

subroutine splitop_onestep()

implicit none

complex(kind=wp), allocatable, save, dimension(:) :: psi2, psi3

logical, save :: initflag = .true.

```

```

if ( initflag ) then
  initflag = .false.
  if ( order .ge. 4 ) allocate( psi2(Nx) )
  if ( order .ge. 6 ) allocate( psi3(Nx) )
end if

if ( order .eq. 2 ) then
  call splitop_ord2_onestep(t, dt, psi)
else if ( order .eq. 4 ) then
  psi2 = psi
  call splitop_ord2_onestep(t, dt/2, psi2)
  call splitop_ord2_onestep(t+dt/2, dt/2, psi2)
  call splitop_ord2_onestep(t, dt, psi)
  psi = (4.0_wp/3.0_wp) * psi2 - (1.0_wp/3.0_wp) * psi
else if ( order .eq. 6 ) then
  psi3 = psi
  call splitop_ord2_onestep(t, dt/3, psi3)
  call splitop_ord2_onestep(t+dt/3, dt/3, psi3)
  call splitop_ord2_onestep(t+dt*2/3, dt/3, psi3)
  psi2 = psi
  call splitop_ord2_onestep(t, dt/2, psi2)
  call splitop_ord2_onestep(t+dt/2, dt/2, psi2)
  call splitop_ord2_onestep(t, dt, psi)
  psi = (1.0_wp/24.0_wp) * psi &
    - (16.0_wp/15.0_wp) * psi2 &
    + (81.0_wp/40.0_wp) * psi3
else
  write(0,*) 'Error (SPLITOP_ONESTEP): invalid order = ', order
  stop
end if

end subroutine splitop_onestep

!-----
!
! subroutine splitop_ord2_onestep(tloc, dtloc, psiloc)
!
! Evolves psi by one step via the second-order explicit split-operator
! method.
!-----

subroutine splitop_ord2_onestep(tloc, dtloc, psiloc)

  implicit none

  complex(kind=wp), dimension(Nx), intent(inout) :: psiloc
  real(kind=wp), intent(in) :: tloc, dtloc

  real(kind=wp), allocatable, save, dimension(:) :: Top, Vop
  complex(kind=wp), allocatable, save, dimension(:) :: psitemp

  logical, save :: initflag = .true.

  integer :: j, k

  ! set up the coefficients and operators only once
  if ( initflag ) then
    initflag = .false.
    allocate(psitemp(Nx))
    allocate(Vop(Nx), Top(Nx))

    ! potential energy
    !Vop = wx**2 * x**2 + unonlin*abs(psiloc)**2
    Vop = wx**2 * (x**4 - ax* x**2) + unonlin*abs(psiloc)**2

    ! kinetic energy operator (in momentum basis)
    Top = p*p
  end if

  ! first take half of spatial step
  psiloc = psiloc * exp(- i / (2.0_wp*hbar) * Vop * dtloc * 0.5_wp )

  ! now transform to p space
  call fftsid(-1, psiloc)

  ! apply drift part
  psiloc = psiloc * exp(- i / (2.0_wp*hbar) * Top * dtloc)

  ! check numerics (make sure not much population at edges of p grid)
  if ( maxval( (/ abs(psiloc(Nx/2)), abs(psiloc(Nx/2+1)) /) )**2 &
    .gt. 1e-5 * maxval( abs(psiloc) )**2 ) then
    write(0,*) 'Warning (SPLITOP_EVOLVER): possible aliasing effects (p)'
  end if

  ! back to x space
  call fftsid(1, psiloc)
  psiloc = psiloc/(Nx)

  ! do second half of spatial step
  psiloc = psiloc * exp(- i / (2.0_wp*hbar) * Vop * dtloc * 0.5_wp )

  ! check numerics (make sure not much population at edges of x grid)
  if ( maxval( (/ abs(psiloc(1)), abs(psiloc(Nx)) /) )**2 &
    .gt. 1e-5 * maxval( abs(psiloc) )**2 ) then
    write(0,*) 'Warning (SPLITOP_EVOLVER): possible aliasing effects (x)'
  end if

```

```

end subroutine splitop_ord2_onestep

end module splitop_evolver

!-----
!
! module param_parser
!
! Contains the parse_params subroutine for loading parameters from
! the parameter (.param) file. Also does some setup for grids,
! etc.
!-----

module param_parser

  use ffts
  use globals
  use utilities
  use timing

contains

!-----
!
! subroutine parse_params()
!
! Loads parameters from the parameter file, which is given on the
! command line. Also constructs a few other static parameters.
!
! To get the parameters, use one of 3 functions:
! get_real_param(p_file, 'str', var, trace)
! get_int_param(p_file, 'str', var, trace)
! get_log_param(p_file, 'str', var, trace)
! get_str_param(p_file, 'str', var, trace)
!
! for reading in parameters of type real, integer, logical, or
! string, respectively. Arguments:
! p_file (string) -> file name of parameter file
! 'str' (string) -> name of parameter to get as written in
! param file
! var (various) -> value of parameter from file is returned here
! trace (logical) -> whether or not to print out the parsed
! value of the param to std err
!
! see sample parameter file for more info on parameters
!-----

subroutine parse_params()

  implicit none

  integer :: ierr, j
  character(64) :: p_file

  ! parse command line argument (for parameter file name)
  if ( command_argument_count() .ne. 1 ) then
    ierr = 1
    do while ( ierr .ne. 0 )
      write(*, '(30A)', advance='no') 'Enter name of parameter file: '
      read(*, '(A)', iostat=ierr) p_file
    end do
  else
    call get_command_argument(1, p_file)
  end if
  p_file = trim(p_file)

  ! get display modes
  call get_log_param(p_file, 'param_trace', param_trace, .false.)
  if ( param_trace ) then
    write(0,*) 'Reading parameters from file ', p_file(1:indlnb(p_file))
  end if
  call get_log_param(p_file, 'time_trace', time_trace, param_trace)
  if ( time_trace ) call initialize_timer()
  call get_log_param(p_file, 'norm_trace', norm_trace, param_trace)

  ! get grid parameters
  call get_int_param(p_file, 'Nx', Nx, param_trace)
  ! if you bitwise-AND Nx with Nx-1, then the result = 0 iff Nx is a power of 2
  ! cool, eh?
  if ( iand(Nx, Nx-1) .ne. 0 ) then
    write(0,*) 'Error (PARSE_PARAMS): Nx must be a power of 2'
    stop
  end if
  call get_real_param(p_file, 'dx', dx, param_trace)

  ! get time evolution parameters
  call get_real_param(p_file, 'dt', dt, param_trace)

```

```

call get_real_param(p_file,'tfinal',tfinal,param_trace)
Nt = int(tfinal/dt)
call get_int_param(p_file,'Ntout',Ntout,param_trace)
if ( Ntout .gt. Nt ) then
  Ntout = Nt
  write(0,*) 'Warning: Ntout reset to ', Ntout
end if
toutevery = Nt/Ntout

call get_int_param(p_file,'order',order,param_trace)

! get Schrodinger equation parameters
call get_real_param(p_file,'wx',wx,param_trace)
call get_real_param(p_file,'hbar',hbar,param_trace)

! get nonlinear term of GPE
call get_real_param(p_file,'unonlin',unonlin,param_trace)
call get_real_param(p_file,'ax',ax,param_trace)

! get initial condition parameters
call get_real_param(p_file,'sigma_x',sigma_x,param_trace)
call get_real_param(p_file,'x0',x0,param_trace)

! output file stuff
call get_str_param(p_file,'base_file_tag',base_file_tag,param_trace)

! momentum grid stuff
dp = 2.0_wp * pi / real(Nx, kind=wp) / dx
pmax = dp * real(Nx, kind=wp) * 0.5_wp

if ( param_trace ) then
  write(0,*)
  write(0,*) 'Grid statistics:'
  write(0,*) ' maximum x value: ', real(Nx/2,kind=wp)*dx
  write(0,*) ' total time steps: ', Nt
  write(0,*) ' time steps between outputs: ', toutevery
  write(0,*) ' maximum p value: ', pmax
end if

if ( time_trace ) then
  write(0,*)
  write(0,*) 'Initializing ...'
end if

! allocate grids and wave function arrays
allocate(psi(Nx), phi(Nx), xdist(Nx), pdist(Nx), x(Nx), p(Nx))
allocate(psi2(Nx), phi2(Nx), xdist2(Nx), pdist2(Nx), x2(Nx), p2(Nx))
write(0,*) 'after allocate in param'
! x grid
x = (/ (real(j-Nx/2,kind=wp)*dx, j=0,Nx-1) /)

! shift p vector so we don't need fftshifts
p = (/ (real(j-Nx/2,kind=wp)*dp, j=0, Nx-1) /)
p = cshift(p, Nx/2)

write(0,*) 'got end of param'
end subroutine parse_params
end module param_parser

=====
!
! globals.f90
!
! Module that contains global variables for the schid simulation.
!
=====

module globals

  implicit none

  ! character
  character :: filename1, filename2

  ! parameters
  integer, parameter :: wp = selected_real_kind(p=14)
  complex(wp), parameter :: i = (0.0_wp, 1.0_wp)

  ! this would be a parameter, but there is a bug in g95 that prevents this
  real(wp) :: pi = 3.141592653589793_wp

  !
  ! control parameters
  logical :: param_trace, time_trace, norm_trace

  ! Schrodinger equation parameters
  real(wp) :: wx, hbar

  ! nonlinear term of GPE

```

```

real(wp) :: unonlin
real(wp) :: ax

! grid parameters
real(kind=wp) :: dx, dp, pmax, dt, t, tfinal
integer :: Nx, Nt, Ntout, toutevery, order

! initial condition parameters
real(kind=wp) :: sigma_x, x0

! grids
complex(wp), allocatable, dimension(:) :: psi, initpsi, phi
complex(wp), allocatable, dimension(:) :: psi2, initpsi2, phi2
real(wp), allocatable, dimension(:) :: xdist, pdist
real(wp), allocatable, dimension(:) :: x, p

! file parameter: files are named <base_file_tag>.<ext>,
! where 'ext' is different for each file (but generally ends in .txt)
character(64) :: base_file_tag

end module globals

```

References

- [1] Gardiner, S.A., D Jaksch, R. Dum, J.I. Cirac, and P. Zoller, 'Nonlinear matter wave dynamics with a chaotic potential, *Physical Review A*, Vol. 62, 2000
- [2] Stenholm, Stig, 'Validity of the Gross-Pitaevski Equation Describing Bosons in a Trap', *Physical Review A*, V54 4, 1998
- [3] Toner, John, lecture notes from the course 'statistical mechanics', University of Oregon, Spring 2006
- [4] Wolfram, S., *A New Kind of Science*, Wolfram Media, New York 2002
- [5] Weisstein, Eric W. *Chaos*. [http: //mathworld.wolfram.com/Chaos.html](http://mathworld.wolfram.com/Chaos.html)
- [6] Wikipedia., *Chaos Theory*. [http: //en.wikipedia.org/wiki/Chaos.theory](http://en.wikipedia.org/wiki/Chaos.theory)
- [7] Touchey, P., *Yet Another Definition of Chaos*, *The American Mathematical Monthly*, **104**, 411-414, 1997
- [8] Casati, G., Chirikov, B., *Quantum Chaos*, Cambridge University press, Cambridge, 1995
- [9] Neil Rasband, *Chaotic dynamics of nonlinear systems*, John Wiley and sons, New York, 1990