UNIVERSITY OF OSLO
Department of Physics

Bose-Einstein Condensates

Numerical solution of the Gross-Pitaevskii equation using finite elements

Master thesis

Carl Joachim
Berdal Haga

June 2006

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my supervisors for their support throughout my thesis work. Prof. Morten Hjorth-Jensen, my primary supervisor, for his help with physics questions and extensive writing support, as well as unflagging optimism; Prof. Hans Petter Langtangen, who wrote if not *the* book then at least *a* book on the finite element method, and whose intimate knowledge of diffpack saved the day on a few occasions; and Dr. Halvor Møll Nilsen, who knows well how to simulate Bose-Einstein condensates, having done it a lot, and has corrected several of my mistakes, and suggested fruitful methods and avenues of investigation.

Chapter 1

Introduction

Bose-Einstein condensation describes a peculiar state of matter that occurs when certain gases are cooled to a few billionths of a degree above absolute zero temperature.[1] It is characterised by all atoms entering the same quantum state,[2] thus merging to form what has been described as one super-atom.

All matter obey the laws of quantum mechanics. Yet, usually the behaviour that we experience can be described by the simpler classical laws of Newtonian mechanics. A room-temperature gas behaves almost like a swarm of billiard balls, bouncing against each other. At low temperatures, however, the wave nature of the atoms becomes apparent, and the quantum effects start to dominate. This is illustrated in figure 1.1. Atoms that behave as shown in the figure are called bosons, while fermions, the other species of atoms, do not condense in this way.

The three quantum effects that cause Bose-Einstein condensation are the indistinguishability of particles, the discrete nature of the energy of each particle, and that several particles are allowed to be in the same state. The first two are a consequence of the wave nature of atoms; the third is only true for bosons. As we shall see in section 2.3, these effects conspire to make bosons highly social at low temperatures and tend to congregate in the lowest-energy state.

A popular introduction can be found at the BEC Homepage (see bibliography). It has several applets demonstrating what Bose-Einstein condensation is and how it is made. It is worth a visit if the concepts are unclear, since the demonstrations may help to develop a mental model of what goes on. An intermediate-level review of the subject with recent developments is found in (Anglin and Ketterle,

---

[1] Temperature is a measure of the average kinetic energy of the atoms. Atoms in room-temperature air zip around at 500 m/s. At temperatures where Bose-Einstein condensation occurs, they crawl at no more than a few cm per second.

[2] The state of a particle is its position, velocity, and any other properties that are required to describe it.

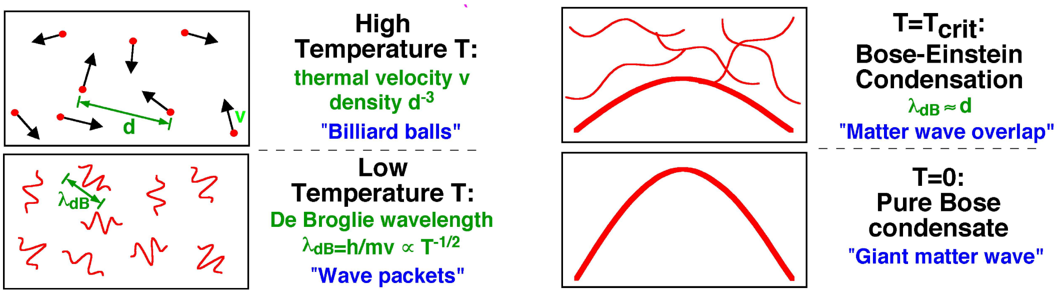


Figure 1.1: Diagram of the properties of particles in a gas at various temperatures. At high temperatures (a cold winter night is “high temperature” in this context), the particles behave like solid balls (upper left). As the temperature drops, the balls become less well defined and their wave nature starts to show (lower left). At the critical temperature where condensation begins, the waves from different particles start to overlap and combine (upper right), and as the temperature drops to absolute zero they overlap completely (lower right). Figures from (Ketterle, 2001).

2002), while (Pethick and Smith, 2001) is a good advanced level introduction to the field.

A number of applications for Bose-Einstein condensates have been proposed. But because the condensate is fragile — even the slightest contact with matter at normal temperature destroys it — practicality is hard to achieve. On the other hand, its sensitivity to disturbances also has its advantages, e.g., for high precision measurements. What has been achieved already is an improved understanding of the quantum world. Because of the relative ease of manipulation and visualisation of the macroscopic wave function of the condensate, quantum mechanical phenomena like for example wave interference, superfluidity and quantised vortices can be more easily investigated experimentally. Precise measurements of some fundamental forces and constants of nature are possible (see for example Harber et al., 2005).

A rather promising approach is to use the condensate as a source of coherent atoms for industrial processes; an atom laser, if you will. Target applications are for example atomic clocks, high precision measurements, atom litography for computer chip production, and nanotechnology (see Ketterle, 1997).

Bose-Einstein condensates have been used as the medium inside which light has been slowed or even stopped, in the sense that the phase of the travelling light beam is imprinted on the condensate to later be released (Dutton et al., 2004). This process has been proposed as a way of storing information in optical communication switches, or even for storing 3D scenes for holography.

Still, applications of the Bose-Einstein condensate are in their infancy. But it is worth considering the fate of another macroscopic scale phenomenon of coherent bosons, the laser.[3] At the time of its invention it met with considerable scepticism and was called, only partly in jest, “a solution looking for a problem” (Townes, 2003). There turned out to be no shortage of problems.

The aims of this master thesis are twofold. The primary purpose at its inception was to investigate the numerical tools for solving quantum mechanical problems, and in particular the nonlinear Schrödinger equation. The choice of the finite element method was then a natural one, because of considerable local knowledge. The finite element method is a standard method in engineering and in many fields of science, but it is not commonly used in the field of quantum mechanics. One reason for this is that in quantum mechanical research, problems seldom have complicated geometries, while in engineering this cannot be avoided. Complicated geometries is one area where the more general finite element method shines. Another reason is that the finite element method is quite hard to understand and implement, compared to other widely used methods such as the finite difference method. Thus, while a finite difference solver can (and often is) coded by the researcher from scratch, an efficient implementation of the finite element method requires the use of a specialised library or tool. Gaining familiarity with such tools, and learning to trust them, takes time. We believe, however, that the payoff can be large, given the flexibility that finite element tools provide.

The secondary (but by no means inferior) purpose was to use these numerical methods to investigate a suitably interesting physical system. Bose-Einstein condensation is such a system; it has been an active research area since first achieved experimentally in 1995. Even though it can be described by a simple partial differential equation, it exhibits a wide variety of phenomena because of its nonlinearity.

The thesis is structured as follows.

Chapter 2 gives a general overview of the field of Bose-Einstein condensation, and tries to explain the interest shown in this field in recent years. It is aimed at a non-specialist but physics-literate audience.

Chapter 3 gives a more formal background of the physics, and derives the mean-field equations that govern the Bose-Einstein condensates under dilute conditions. Chapter 4 introduces the finite element method, and gives an overview of the range of numerical methods used to solve linear and nonlinear problems. These chapters are aimed at the level of a master student of physics.

In chapter 5, the results of this investigation are presented. Results from simulations are compared with earlier simulations, both finite difference calculations using the same mean-field approximations and *ab initio* Monte Carlo simulations. Numerical results on convergence and efficiency are also shown.

---

[3] The discovery and theory of two other such phenomena have also been awarded the Nobel prize: Superconductivity (1913, 1972, 1987, 2003) and superfluidity (1996, 2003).

Chapter 6 concludes the thesis, and discusses the outlook for further work on simulation of the nonlinear Schrödinger equation.

Chapter 2

Background

2.1 History of Bose-Einstein condensation

Bose-Einstein condensation, or BEC, has a long and rich history going back to the early years of quantum mechanics. After being first proposed by Einstein and Bose in 1924, it was not observed until 1995. It draws on fundamental discoveries in diverse fields such as condensed-matter physics, laser cooling and trapping, and fluid mechanics, and the history covers a number of Nobel prize winners and other prominent physicists.

The main reference, unless otherwise noted, of this section is the Nobel lecture by Cornell and Wieman (2002), with supplementary information from Griffin (1998). References to the original papers are found in these texts. Another good, although less detailed, account is from the Nobel Prize press release (Royal Swedish Academy of Sciences, 2001).

2.1.1 Prediction

In 1924, Bose developed the statistics of photons, based on Einstein's idea of photons being discrete particles of light. He thus succeeded in replicating the black-body radiation spectrum. The year after, Einstein published an extension to this work where the number of particles was fixed, as it is in ordinary matter. He remarked upon the possibility of condensation: "A separation is effected; one part condenses, the rest remains a saturated ideal gas".

It is interesting to note that Einstein's work on Bose-Einstein statistics was inspired by de Broglie's idea of matter having a wave nature, thus being complementary to Einstein's showing that light has a particle nature. This all happened early in the development of quantum mechanics; Schrödinger apparently first heard of de Broglie's work through this paper. Schrödinger's wave equations, which are the basis of much of modern quantum mechanics, were published the following year.

But for the next few decades, the prediction was not taken seriously. It was believed that the condensation would disappear from the equations once the interactions were properly accounted for, and Einstein did not write about it again.

2.1.2 Superfluidity in liquid helium

It was known from experiments that liquid ^{4}He became superfluid at low temperatures. In the late 1930s, London and Tisza suggested that Bose-Einstein condensation was the underlying mechanism of the superfluidity, especially since ^{3}He (a fermion) turned out not to become superfluid.[1] Although largely ignored in the beginning, this hypothesis gradually gained currency. But because of the strong interactions in liquid helium, it is hard to treat microscopically — the more successful approach was the hydrodynamical one — and the weakly interacting boson gas was used as a toy model to investigate the condensation effects in isolation from the interaction effects.

Why a toy model? Because BEC treats a gas of bosons, but helium is the only known atom or molecule to even remain liquid at sufficiently low temperatures. All others freeze to solid state. In fact, the gaseous phase that has the required dilute density and low temperature is deep in the thermodynamically forbidden regime, and it was thought that this state was not experimentally realisable. Nevertheless, theoretical work on the toy model continued, and through the 1950s and 1960s a solid theory for both the ground state and excitations was developed by Bogoliubov, Beliaev, Gross, Pitaevskii, Wang, Wu and others.

2.1.3 The search for Bose-Einstein condensation in dilute gases

In 1976, the interest in Bose-Einstein condensation was renewed when Stwalley and Nosanow confirmed Hecht's earlier but little known result, that spin-polarised hydrogen had no bound states and thus would be a good candidate for a dilute-gas BEC. This, together with advances in cryogenic technology that among other things had allowed the discovery of superfluid ^{3}He, started a search for BEC by several experimental groups that would last almost twenty years. By the mid-1980s, polarised hydrogen was brought to within a factor 50 of condensing, but the (at the time) current approach for cooling was then reaching its limit. The hydrogen was confined to a refrigeration cell where the walls were

---

[1] ^{3}He does actually become superfluid, because of particle pairing, but at a much lower temperature. This was discovered in 1972.

Table 2.1: Limits of current cooling techniques for a gas of sodium atoms (Pethick and Smith, 2001; Martikainen, 2006; Leanhardt et al., 2003).

Method	Limiting process	$T_{\min}$
Doppler cooling	The decay rate of the two-level atom	500μK
Sisyphus cooling	The recoil energy of a single photon	10μK at $\lambda = 1\mu$m
(Critical temperature	*See figure 1.1 (p. 2)*	50–500nK *at typical densities)*
Evaporative cooling	Atom loss and thermalisation/recombination rate	< 500pK

covered with superfluid liquid helium to avoid sticking, but this became ineffective at these low temperatures. It was realised that electromagnetic trapping of the atoms was necessary, to allow the condensate to be held in vacuum.

Electromagnetic trapping would also allow cooling through evaporation, where the most energetic atoms are allowed to escape, thus lowering the average energy of the remaining atoms. By 1991, a factor ten improvement had been achieved using this method. But by then other problems surfaced, perhaps most fundamentally the difficulty of measuring these very cold and small clouds.

Meanwhile, the technology of laser cooling was improving at a significant rate. In 1975, Hänsch and Schawlow recognised that a laser can be used to cool free atoms, by tuning the wavelength of the light so that it is only absorbed by atoms that are moving toward the laser. This so-called *Doppler cooling* showed promise in the early 1980s, culminating in two striking experiments in 1985. Phillips managed to capture a moving cloud of particles in a magnetic trap, and Chu set up a three-dimensional configuration of laser beams that slowed all directions of movement simultaneously; an *optical molasses.* Together with Cohen-Tannoudji they discovered and explained the phenomenon of *Sisyphus cooling.*[2] The three of them were awarded the Nobel prize in 1997 for their work on laser cooling and trapping (Royal Swedish Academy of Sciences, 1997).

2.1.4 BEC in alkali atoms

The technology of laser cooling was not very useful for hydrogen atoms, however. The spectrum of hydrogen is strongest in the deep ultraviolet, not in resonance

---

[2] The Sisyphus of Greek myth was punished by the gods to eternally wrestle a stone up a hill, but it always escapes him and he has to start over. Sisyphus cooling thus derived its name from the intuitive image of the atoms always climbing up a slope in the electromagnetic potential, losing kinetic energy, then expending much of its potential energy by releasing a photon and falling down (Phillips, 1998).

with any convenient laser source. At the same time, it became clear that the absence of a bound state is not required. As long as the rate of inelastic (recombinant) collisions can be controlled, the lifetime of a gas can be made long enough for successful experiments; the gas will be in a protracted metastable state.

For evaporative cooling to be effective, the rate of elastic collisions has to be high enough for kinetic equilibrium to be established quickly. The elastic collisions are two-particle collisions, which happen at a rate proportional to the density squared, while inelastic collisions are three-particle collisions which go as density cubed. Thus, diluteness is the factor that controls the lifetime of the unbound state.

The attention of several groups turned to alkali atoms, since they provide many hyperfine levels accessible by smaller and cheaper lasers. The JILA group (Wieman, Cornell) used ^{87}Rb, the MIT group (Kleppner, Pritchard, Ketterle) worked with ^{23}Na, Hulet at Rice chose ^{7}Li; they all eventually proved successful. The first two groups achieved BEC within a few months of each other in 1995, leading to Cornell, Ketterle and Wieman sharing the Nobel prize in 2001.

The lithium atoms turned out to be harder, because their interaction is attractive instead of repulsive.[3] An attractive cloud is unstable and collapses onto itself when the number of atoms is too large. Nevertheless, BEC was achieved in 1997 in a cloud of about 1000 lithium atoms.

Another advantage of using alkali atoms is the ease of imaging. By turning the trap off and letting the cloud expand in free fall for a few milliseconds, the cloud forms what is essentially an image of its original velocity distribution, of the size of perhaps 1mm. This cloud can be directly imaged by resonant light. Figure 2.1 is made in this way.

2.1.5 Further experiments

After the first successful condensations were achieved, the field saw a flurry of activity, and within a few years more than 20 groups had produced BEC in alkali atoms. The spin-polarised hydrogen experiments also finally succeeded in 1998, although direct imaging was not possible.

Since then, experimentalists and theorists have explored a wide range of properties of the condensates. These include collective excitations and nonlinear

---

[3] The sign of the scattering length, thus whether the potential was attractive or repulsive, was not known in advance, but a geometrical argument can be made that 3 out of 4 isotopes have positive scattering lengths (Pethick and Smith, 2001, p. 119). The strategy was to try several and hope for the best.

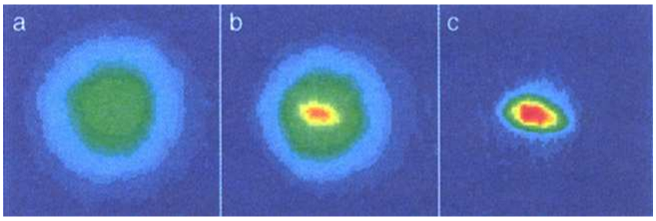


Figure 2.1: Observations of BEC in sodium atoms achieved in the MIT group. Figure (a) is the velocity distribution of a cloud cooled to just below the transition point, (b) just after the condensate appeared, and (c) after further evaporative cooling has left an almost pure condensate. Figure (b) shows the difference between the isotropic thermal distribution and an elliptical core attributed to the expansion of a dense condensate. The figures are from (Davis et al., 1995).

phenomena such as vortex formation[4] and solitons; phase coherence phenomena like fringe and interference patterns; sympathetic cooling and two-component condensates; lower-dimensional and lattice condensates; superfluidity; et cetera. The review article (Anglin and Ketterle, 2002) covers many of these developments.

A striking result of these experiments is how successfully their behaviour is described by the early theoretical work of the 1950s and 1960s. The validity of the Gross-Pitaevskii equation (which is the main focus of later chapters) and Bogoliubov's theory of fluctuations depends on the gas being very dilute, and this fits well with the required experimental conditions.

The behaviour of condensates at higher temperatures and densities requires other treatments, and are also harder to achieve experimentally due to the shortened lifetime of the metastable gaseous state. The exploration of these regimes is at the frontier of many-body physics.

2.2 Quantum particles

In order to understand how Bose-Einstein condensation can occur, we must start by looking at the properties of individual particles, and the many-particle statistics that follow.

Even in classical thermodynamics, we see evidence that particles of the same type are indistinguishable. By this we mean not only that we cannot tell them apart because we do not have accurate enough measurements, but that even in principle they are identical. This is required to explain Gibb's paradox, where otherwise partitioning a container of gas would lower its total entropy.

---

[4] Lattices of up to several hundred vortices has been seen, see for example figure 5.5(b) (p. 88).

Still, in classical physics we could in principle follow the trajectory of a particle, and if it collides with an identical particle then we could make a note of which particle went which way afterwards. Thus we could tell them apart if we knew their histories. Quantum mechanics does not allow this, as their positions and trajectories are known only in a probabilistic sense, and what happens outside of measurements is not a valid question.

Hence we can not distinguish two states where one is created by an interchange of particles in the other. No possible measurement we make can tell them apart. In the following sections we investigate the consequences of this.

2.2.1 Symmetry

We start by looking at the symmetry states of identical particles. The discussion follows (Ballentine, 1998).

Permutation symmetry

Two physical situations that differ only in the interchange of identical particles are indistinguishable. We look at how this affects the eigenstates of the Hamiltonian of the system.

Consider a system of two identical particles. We construct a basis vector as a product of single particle basis vectors with quantum numbers α and β,

$$|\alpha\beta\rangle \equiv |\alpha\rangle \otimes |\beta\rangle, \tag{2.1}$$

and an arbitrary two-particle state as a linear combination of these,

$$|\Psi\rangle = \sum_{\alpha\beta} c_{\alpha\beta} |\alpha\beta\rangle. \tag{2.2}$$

We then define the permutation operator

$$P_{12}|\alpha\beta\rangle = |\beta\alpha\rangle. \tag{2.3}$$

From the permutation invariance of the Hamiltonian, it follows that $[P_{12}, H] = 0$ and thus that H and P_{12} share a complete set of eigenvectors:

$$P_{12}|\alpha\beta\rangle = \lambda|\alpha\beta\rangle. \tag{2.4}$$

We can construct the state wave function as

$$\Psi(\mathbf{x}_1, \mathbf{x}_2) = (\langle \mathbf{x}_1| \otimes \langle \mathbf{x}_2|)|\Psi\rangle = \sum_{\alpha\beta} c_{\alpha\beta} \langle \mathbf{x}_1|\alpha\rangle\langle \mathbf{x}_2|\beta\rangle, \tag{2.5}$$

thus the action of the permutation operator on the wave function is to exchange the particle coordinates.

It is obvious from equation (2.3) that $(P_{12})^2 = 1$, and thus $\lambda^2 = 1$. The eigenvalues λ of P_{12} are then either 1 (symmetric eigenfunction) or -1 (antisymmetric eigenfunction),

$$\Psi(\mathbf{x}_1, \mathbf{x}_2) = \pm\Psi(\mathbf{x}_2, \mathbf{x}_1). \tag{2.6}$$

For a state of more than two particles, the situation is more complicated. Consider a state of three identical particles, $|\alpha\beta\gamma\rangle$. Then

$$P_{12}|\alpha\beta\gamma\rangle = |\beta\alpha\gamma\rangle, \quad P_{23}|\alpha\beta\gamma\rangle = |\alpha\gamma\beta\rangle, \quad P_{31}|\alpha\beta\gamma\rangle = |\gamma\beta\alpha\rangle. \tag{2.7}$$

These operators clearly do not commute, e.g.,

$$P_{12}P_{23}|\alpha\beta\gamma\rangle = |\gamma\alpha\beta\rangle \neq |\beta\gamma\alpha\rangle = P_{23}P_{12}|\alpha\beta\gamma\rangle, \tag{2.8}$$

thus they cannot all have a complete set of common eigenvectors with H. We can, however, explicitly construct permutation invariant subspaces of $\mathcal{H}^3$. The invariant subspaces are in this case spanned by the basis vectors

- Symmetric subspace

$$|\Psi\rangle_{\mathrm{S}} = \frac{1}{\sqrt{6}}\left(|\alpha\beta\gamma\rangle + |\beta\alpha\gamma\rangle + |\alpha\gamma\beta\rangle + |\gamma\beta\alpha\rangle + |\gamma\alpha\beta\rangle + |\beta\gamma\alpha\rangle\right). \tag{2.9}$$

- Antisymmetric subspace

$$|\Psi\rangle_{\mathrm{AS}} = \frac{1}{\sqrt{6}}\left(|\alpha\beta\gamma\rangle - |\beta\alpha\gamma\rangle - |\alpha\gamma\beta\rangle - |\gamma\beta\alpha\rangle + |\gamma\alpha\beta\rangle + |\beta\gamma\alpha\rangle\right). \tag{2.10}$$

- Partially symmetric (one of several)

$$\begin{aligned}&\frac{1}{2}\left(|\alpha\gamma\beta\rangle - |\gamma\beta\alpha\rangle + |\gamma\alpha\beta\rangle - |\beta\gamma\alpha\rangle\right),\\ &\frac{1}{\sqrt{12}}\left(2|\alpha\beta\gamma\rangle - 2|\beta\alpha\gamma\rangle - |\alpha\gamma\beta\rangle - |\gamma\beta\alpha\rangle + |\gamma\alpha\beta\rangle + |\beta\gamma\alpha\rangle\right).\end{aligned} \tag{2.11}$$

We see that for many-particle systems, we have to impose the permutation invariance of H explicitly, and the symmetry state is arbitrary. Experimental evidence, however, overwhelmingly supports the statement that only the first two symmetry states exist for identical particles in nature, and that the symmetry depends on the total spin of the particle. This leads to an additional postulate:

The symmetrisation postulate

(a) Particles whose spin is an integer multiple of $\hbar$ have only symmetric states. These particles are called *bosons*.

(b) Particles whose spin is a half odd-integer multiple of $\hbar$ have only antisymmetric states. These particles are called *fermions*.

(c) Partially symmetric states do not exist.[5]

It is important to note that even though all baryonic elementary particles (particles of matter) have half-integral spin, a compound particle can still have integral spin and thus be bosonic. For example, neutral atoms are bosons if they have an even number of neutrons, since the sum of protons and electrons is always even and thus has integral total spin.

2.2.2 The Pauli exclusion principle

If we insert $\alpha = \beta$ into equation (2.10), or $\mathbf{x}_1 = \mathbf{x}_2$ into the antisymmetric variant of equation (2.6), the state must be equal to 0. The same clearly applies to the antisymmetric state of any number of particles. This is known as the Pauli exclusion principle, and the consequence is that any two fermions may not have the same quantum numbers.

Bosons are not bound by this principle, a fact that we shall see has radical consequences at low temperatures.

2.3 Particle statistics

We now look at the behaviour of a large number of particles by deriving their distributions — that is, if the temperature is the average energy of the particles, the distribution is the range and spread of the particle energies.

2.3.1 Maxwell-Boltzmann statistics

The Maxwell-Boltzmann distribution governs the statistics of classical particles. We include it here because its simple form, and to demonstrate its validity at higher temperatures, where it becomes similar to the quantum mechanical distributions below.

---

[5] Not in 3D (real) space. They may exist in lower-dimensional space, and are then called *paraparticles* or *anyons*.

The probability of a single particle being in a state of energy ϵ is (Landau and Lifshitz, 1969)

$$\mathcal{P}_\epsilon = \frac{1}{Z_1} e^{-\epsilon/kT} \tag{2.12}$$

independent of the particle. Therefore the average is just multiplied by the total number of particles, N.

$$\overline{n}_{\mathrm{MB}} = N\mathcal{P}_\epsilon = \frac{N}{Z_1} e^{-\epsilon/kT}. \tag{2.13}$$

The chemical potential is given by $\mu = -kT \ln(Z_1/N)$, thus $N/Z_1 = e^{\mu/kT}$:

$$\overline{n}_{\mathrm{MB}} = \frac{1}{e^{(\epsilon-\mu)/kT}}. \tag{2.14}$$

2.3.2 Fermi-Dirac statistics

The Fermi-Dirac distribution concerns itself with quantum particles that cannot be in the same state because of the Pauli exclusion principle.

Using the grand partition function, the probability of a single state of energy ϵ being occupied by n particles is

$$\mathcal{P}_\epsilon(n) = \frac{1}{\mathcal{Z}} e^{-n(\epsilon-\mu)/kT}, \tag{2.15}$$

and $\mathcal{Z}$ is the sum of the Gibbs factor for all possible n, so that the sum of probabilities is 1. When the particles are fermions, a single particle state can only be occupied by 0 or 1 particles,

$$\mathcal{Z} = 1 + e^{-(\epsilon-\mu)/kT}, \tag{2.16}$$

and the average number of particles in a state is

$$\overline{n}_{\mathrm{FD}} = \langle n \rangle_\epsilon = \sum_{n=0}^{1} n\mathcal{P}_\epsilon(n) = \frac{1}{e^{(\epsilon-\mu)/kT} + 1}. \tag{2.17}$$

2.3.3 Bose-Einstein statistics

When the particles are bosons, a state may be occupied by any number of particles. The grand partition function becomes

$$\begin{aligned}\mathcal{Z} &= \sum_{n=0}^{\infty} \left(e^{-(\epsilon-\mu)/kT}\right)^n \\ &= \frac{1}{1 - e^{-(\epsilon-\mu)/kT}}. \end{aligned} \tag{2.18}$$

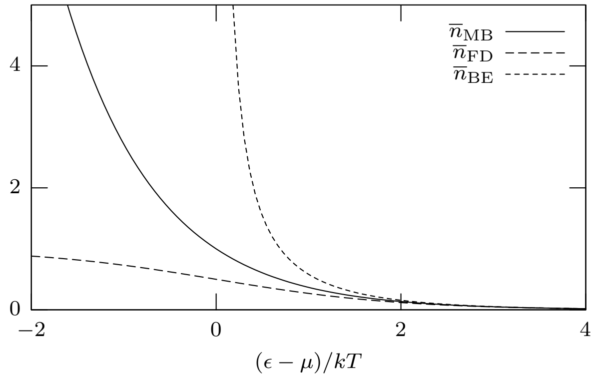


Figure 2.2: Comparison of the Boltzmann, Fermi-Dirac and Bose-Einstein distributions. The three distributions are equal for high-energy states, but very different for low-energy states.

To find the distribution, it is convenient to notice that

$$\frac{\partial}{\partial\chi}\sum_n e^{-n\chi}=\sum_n -ne^{-n\chi}, \tag{2.19}$$

so that with $\chi=(\epsilon-\mu)/kT$ and $\mathcal{P}_\epsilon(n)$ defined as in equation (2.15), the distribution can be written as

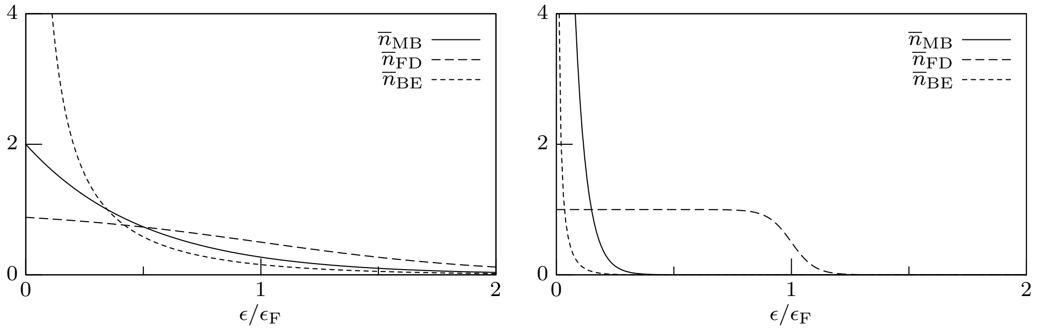
$$\overline{n}_{\mathrm{BE}}=\sum_n n\mathcal{P}_\epsilon(n)=-\frac{1}{\mathcal{Z}}\frac{\partial\mathcal{Z}}{\partial\chi}=\frac{1}{e^{(\epsilon-\mu)/kT}-1}. \tag{2.20}$$

We note that $\epsilon\geq\mu$, because the distribution must be positive everywhere; and furthermore, that for $\langle n\rangle_{\epsilon_0}$ — the probability of finding a particle in the lowest energy state — to remain finite when $kT\rightarrow 0$, it must be the case that $\mu\rightarrow\epsilon_0$

2.3.4 Comparison at $T\rightarrow 0$

The main difference between the statistics of classical distinguishable particles and fermions is this: at low temperatures, most classical particles occupy the few lowest energy states, and at $T=0$ all particles occupy the same states. The Pauli principle disallows this for fermions, as only one particle can be in a given state; so even at $T=0$, a range of energies up to ϵ_{F} (the Fermi energy) are singly occupied. This tendency is clear in figure 2.3(b).

The difference between classical particles and bosons is more subtle. The bosons fall into the lowest energy state at higher temperatures, so the condensation is more pronounced. The indistinguishability in effect boosts the relative entropy of the lowest state. Also, the energy gap between the lower states is relatively wide.



(a) $T = 0.5$

(b) $T = 0.05$

Figure 2.3: Plot of the Boltzmann, Fermi-Dirac and Bose-Einstein distributions relative to their lowest-energy state at successively falling temperatures. In (a) the distributions have started diverging from each other, with the Fermi-Dirac distribution quite flat and the Bose-Einstein distribution growing fast near $\epsilon = 0$. As the temperature falls further (b) the Fermi-Dirac distribution reaches a plateau where all energy states below ϵ_{F} are filled uniformly, while the Bose-Einstein distribution condenses with almost no probability for $\epsilon > 0$. The classical Boltzmann distribution lies in-between.

To contrast, consider distinguishable particles in a system of two levels. Then the state $(N_1 = N, N_2 = 0)$ with all particles in the lowest state has multiplicity 1. The state $(N_1 = N - 1, N_2 = 1)$, where one particle is in the upper level, has multiplicity N, since each particle to move up would be a distinct state. Thus we are N times more likely to find the system in this second state. If the particles are identical, both states have multiplicity 1, and are equally likely.

Chapter 3

Theory of Dilute Bose Gases

In this chapter we will show that a dilute Bose gas is well approximated by a mean field description, where the interaction between particles can be described by the action of an averaged background field, set up by the gas, on each particle independently. Furthermore, even if the interaction is long-range, it can be described accurately by a simple hard-sphere effective potential provided the inter-particle separation is much larger than the scattering length. Thus the interaction can be effectively modelled using only one short range parameter, even when the physical potential is long-range and multi-parameter. We will quantify these ideas in the following sections.

Once the mean field interaction is established, an energy functional is easily derived. The Gross-Pitaevskii, or GP, equation (and its slightly-higher-density sibling, the Modified GP) follow by treating the local wave function energy as that in an equivalent uniform gas at the same density. These are the nonlinear Schrödinger equations that we use for the numerical treatment of Bose-Einstein condensation in later chapters.

The present exposition follows that in (Fetter, 1998); for more details, consult (Fetter and Walecka, 1971) by the same author.

3.1 Many-particle theory

We start by describing some basics of many-particle theory. In particular, why many-particle systems can be approximated by collections of harmonic oscillators, and how canonical quantisation allows us to view the states of a many-particle system as a product of abstract particle creation and annihilation operators.

3.1.1 The harmonic oscillator

The harmonic oscillator is both instructive and of immense practical value, both in classical and in quantum mechanical applications. The reason for this is that any system fluctuating by sufficiently small amounts around a configuration of a stable equilibrium may be described by a system of decoupled harmonic oscillators, regardless of the shape of the actual potential.

Why is this so? Consider the Taylor series expansion of a particle in an arbitrary potential $\tilde{V}(x)$, with a minimum at x_0. We define, for simplicity, $V(x) = \tilde{V}(x - x_0) - \tilde{V}(x_0)$ so that $V(0) = 0$ is a minimum. Then the two first terms in the Taylor series vanish, and we are left with

$$V(x) = \frac{1}{2} \frac{\mathrm{d}^2 V}{\mathrm{d}x^2}\bigg|_0 x^2 + \sum_{n=3}^{\infty} \frac{1}{n!} \frac{\mathrm{d}^n V}{\mathrm{d}x^n}\bigg|_0 x^n. \tag{3.1}$$

The first term is the harmonic oscillator potential

$$V(x) \approx \frac{x^2}{2} \frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(\frac{1}{2} m\omega^2 x^2\right) = \frac{1}{2} m\omega^2 x^2, \tag{3.2}$$

and for a system near equilibrium, the fluctuations $|x|$ are small and we may disregard the other (anharmonic) terms of $\mathcal{O}(x^3)$.

We note that the harmonic oscillator with N degrees of freedom can be solved as a product of N decoupled oscillators. The same cannot be said for the anharmonic terms of equation (3.1), which couple the different degrees of freedom.

3.1.2 Energy basis quantisation

We construct the energy basis quantisation of the harmonic oscillator, following (Fetter and Walecka, 1971).

The quantum mechanical harmonic oscillator Hamiltonian is, if we start with the classical Hamiltonian and let the coordinates go to operators,

$$H = \frac{\hat{p}^2}{2m} + \frac{m\omega^2 \hat{x}^2}{2}, \tag{3.3}$$

which can be written as

$$H = \hbar\omega(a^\dagger a + 1/2), \tag{3.4}$$

by substituting

$$a = \left(\frac{m\omega}{2\hbar}\right)^{1/2} \hat{x} + i \left(\frac{1}{2m\hbar\omega}\right)^{1/2} \hat{p}, \tag{3.5}$$

and its Hermitian conjugate $a^\dagger$. From the commutator $[x,p] = i\hbar$, this implies that the commutator of a and $a^\dagger$ is

$$[a, a^\dagger] = 1. \tag{3.6}$$

But for now, let us look at the scaled Hamiltonian $H' \to H/\hbar\omega$, with energy in units of $\hbar\omega$, for presentation clarity.

The commutators are then

$$[a, H] = [a, a^\dagger a] = a, \tag{3.7}$$

$$[a^\dagger, H] = [a^\dagger, a^\dagger a] = -a^\dagger. \tag{3.8}$$

The utility of these operators lie in that we can use them to construct an arbitrary state. Compare the state with energy ϵ,

$$H|\epsilon\rangle = \epsilon|\epsilon\rangle, \tag{3.9}$$

with the states constructed by acting on $|\epsilon\rangle$ with a and $a^\dagger$:

$$H(a|\epsilon\rangle) = (aH - [a, H])|\epsilon\rangle = (\epsilon - 1)(a|\epsilon\rangle), \tag{3.10}$$

$$H(a^\dagger|\epsilon\rangle) = (a^\dagger H - [a^\dagger, H])|\epsilon\rangle = (\epsilon + 1)(a^\dagger|\epsilon\rangle). \tag{3.11}$$

This shows that they work on the state so as to raise or lower it by one quantum of energy, and that the resulting state is an eigenstate of H.

Since the energy spectrum of the harmonic oscillator is positive, there must be a lowest state such that

$$a|\epsilon_0\rangle = 0. \tag{3.12}$$

Thus the state $|\epsilon_0\rangle$ has energy

$$H|\epsilon_0\rangle = \left(a^\dagger a + \frac{1}{2}\right)|\epsilon_0\rangle = \frac{1}{2}|\epsilon_0\rangle, \tag{3.13}$$

and an arbitrary state can be constructed by applying $a^\dagger$ repeatedly, constructing an equidistant ladder of states. These states are unique, because the ground state is unique.

We make one important observation at this point: Using only equations (3.4) and (3.6), we can find the ground state and the spectrum of the system, without explicit reference to the harmonic oscillator potential or the form of the operators as defined in equation (3.5).

3.1.3 Canonical quantisation

We have not yet discussed how these states represent the particles in a many-particle system. In general, we relate them through canonical quantisation; that is, quantisation of the field.

A *mode* of the field refers to a harmonic oscillator of a particular wave number $\mathbf{k}$, and the field consists of all these modes. Adding a quantum to a mode means adding a particle in the single-particle state $|\psi_{\mathbf{k}}\rangle$ to the field. We therefore number the creation and annihilation operators with the mode that they act on, as

$$a_{\mathbf{k}}^{\dagger}, a_{\mathbf{k}}, \tag{3.14}$$

with the knowledge that the operators for different $\mathbf{k}$ are independent, as the harmonic oscillators are decoupled.

We do not need to know the exact form of these operators to go on. We only need to know that they obey the canonical commutation relations for independent $\mathbf{k}$,

$$[a_{\mathbf{k}}, a_{\mathbf{k}'}^{\dagger}] = \delta_{\mathbf{k},\mathbf{k}'}, \tag{3.15}$$

$$[a_{\mathbf{k}}, a_{\mathbf{k}'}] = 0. \tag{3.16}$$

3.1.4 Fock space representation

We write a Fock state in the occupancy number representation as

$$|n_0, n_1, \ldots, n_N\rangle, \tag{3.17}$$

where n_i is the number of particles in the single-particle state $|\psi_i\rangle$. We enumerate the single-particle states by their wave number $\mathbf{k}$, as above. This representation is particularly suitable for use with the ladder operators, as the following relations show,

$$a_{\mathbf{k}}|\ldots, n_{\mathbf{k}}, \ldots\rangle = \sqrt{n_{\mathbf{k}} - 1}\,|\ldots, n_{\mathbf{k}} - 1, \ldots\rangle, \tag{3.18}$$

$$a_{\mathbf{k}}^{\dagger}|\ldots, n_{\mathbf{k}}, \ldots\rangle = \sqrt{n_{\mathbf{k}}}\,|\ldots, n_{\mathbf{k}} + 1, \ldots\rangle. \tag{3.19}$$

The number operator $N_{\mathbf{k}} = a_{\mathbf{k}} a_{\mathbf{k}}^{\dagger}$ is diagonal in this basis,

$$a_{\mathbf{k}} a_{\mathbf{k}}^{\dagger}|\ldots, n_{\mathbf{k}}, \ldots\rangle = n_{\mathbf{k}}\,|\ldots, n_{\mathbf{k}}, \ldots\rangle. \tag{3.20}$$

3.1.5 A note on quasi-particles

In moving to second quantisation, we described how the field is excited by adding particles to it, and how the excitation of a particular mode of the field comes from the addition of a particle in the corresponding state.

But the identification of modes with real particles is not the only one that makes sense. Any discrete energy bundle can be described by the excitation of its corresponding mode in some field. By analogy with real particles, these field excitations are called quasi-particles. Quasi-particles can be treated just like real particles and obey the same rules like momentum preservation.

Examples of quasi-particles are phonons (quanta of sound) or rotons (quanta of rotation).

3.2 Homogenous Bose gas

We investigate a gas of bosons that has uniform density, and calculate its energy. We begin by considering the ideal gas of non-interacting particles, and proceed to the weakly interacting gas

3.2.1 The uniform, ideal gas

Consider a gas of N particles in a box of volume V. The density of particles inside this box is $n = N/V$. We can trivially solve the Schrödinger equation for the single-particle states, and thus find their energy spectrum

$$\epsilon_k = \frac{\hbar^2 k^2}{2m}. \tag{3.21}$$

for plane waves with wave number $\mathbf{k} = 2\pi V^{-1/3}(n_x, n_y, n_z)$.

In the classical limit, we can use the Boltzmann distribution, equation (2.14), combined with the de Broglie wavelength $\lambda = h/p$, to find the average wavelength at a given temperature; the so-called thermal de Broglie wavelength. It is (Schroeder, 2000, p. 253)

$$\Lambda_T = \left(\frac{2\pi\hbar^2}{mk_\mathrm{B}T}\right)^{1/2}. \tag{3.22}$$

Only when this wavelength is much smaller than the typical particle separation $l \approx n^{-1/3}$,

$$\Lambda_T \ll n^{-1/3}, \tag{3.23}$$

can we ignore quantum effects.

We define the critical temperature T_C to be the temperature where the thermal de Broglie wavelength is of the same order as the particle separation, $n\Lambda_{T_\mathrm{C}}^3 \sim 1$. At this temperature the occupation of the lowest-lying state becomes macroscopic.[1] In other words, it condenses.

---

[1] Detailed calculation shows $n\Lambda_{T_\mathrm{C}}^3 = \zeta(3/2) \approx 2.6$ to be the precise critical ratio (Landau and Lifshitz, 1958, p. 159).

We use the Bose-Einstein distribution, equation (2.20), to find the occupation of the energy levels. Above T_C, we can approximate the sum over levels, and thus the total number of particles, as an integral, but below this temperature the lowest-lying state must be removed from the integral. The remaining terms gives us the number of non-condensed particles. We can then calculate the fraction of particles in the condensed state,

$$\frac{N_0}{N} = 1 - \left(\frac{T}{T_\mathrm{C}}\right)^{3/2}. \tag{3.24}$$

At $T = 0$, all particles are in the lowest state and $N_0 = N$. In the following, we consider systems where nearly all particles are in the lowest single-particle state,

$$\frac{N - N_0}{N} \ll 1, \tag{3.25}$$

which from equation (3.24) implies $T \ll T_\mathrm{C}$.

3.2.2 The weakly interacting gas

In canonical quantisation, we can write the Hamiltonian of the system as, including one- and two-particle terms only,

$$H = \underbrace{\sum_{\mathbf{k}} \epsilon_k a^\dagger_{\mathbf{k}} a_{\mathbf{k}}}_{H_\mathrm{self}} + \underbrace{\frac{1}{2V} \sum_{\mathbf{k}_1\mathbf{k}_2\mathbf{k}_3\mathbf{k}_4} \tilde{V}_{|\mathbf{k}_1-\mathbf{k}_2|} a^\dagger_{\mathbf{k}_1} a^\dagger_{\mathbf{k}_2} a_{\mathbf{k}_3} a_{\mathbf{k}_4} \delta_{\mathbf{k}_1+\mathbf{k}_2,\mathbf{k}_3+\mathbf{k}_4}}_{H_\mathrm{int}} \tag{3.26}$$

where $\tilde{V}$ is the momentum representation of the potential $V(\mathbf{r})$ and δ is the Kronecker delta, ensuring that only momentum conserving wave numbers are included.

The ideal gas has no interaction, $H_\mathrm{int} = 0$. For a sufficiently dilute gas at $T \ll T_\mathrm{C}$, the difference from an ideal gas is small. The interactions are dominated by two-body collisions with small momentum transfer, which are characterised by the s-wave scattering length a.

We briefly try to justify the assumptions above. The unconvinced reader should consult (Pethick and Smith, 2001, ch. 5).

- The inter-particle separation is much larger than the scattering length, thus interactions are dominated by two-particle interactions. Some typical lengths are shown in table 3.1 for reference.
- In alkali atoms in particular, the long-range interactions are relatively strong ($a \sim 100a_0$). A low-energy collision sees the more complicated short-range interactions only in an average sense.

Table 3.1: Lengths scales under typical dilute conditions, using alkali atoms at density $na^3 \approx 10^{-4}$, at $T = 0$. These numbers are an example, and vary in different experiments.

Scale		Typical length
Short-range interaction radius	$\sim 10a_0$	0.5nm
S-wave scattering length	$a \sim 100a_0$	5nm
Particle separation	$n^{-1/3}$	100nm
Healing length	$\xi = (8\pi na)^{-1/2}$	100nm
Trap dimension	$d_0 = (\hbar/m\omega_0)^{1/2}$	1000nm
Wave length	$R \sim R_{\mathrm{TF}}$	5000nm

- Although the short-range interactions are indeed important, the average effect that a low-energy two-particle collision observes can be integrated out, and compensated for by replacing the interaction with an effective interaction that includes the short-range part using a single parameter. The effective interaction will be a good approximation as long as we only consider low-energy interactions.

Thus, *sufficiently dilute* means that collisions are relatively infrequent. We assume that the s-wave scattering length is much shorter than the typical particle separation. The diluteness of the gas can be described by the gas parameter na^3, and we require that

$$na^3 \ll 1. \tag{3.27}$$

The low-energy condition is taken into account by the requirement that $T \ll T_{\mathrm{C}}$.

In this regime, we can approximate the interaction part of (3.26) by the constant two-particle effective interaction term,

$$H_{\mathrm{int}} \approx H_{\mathrm{2p}} \approx \frac{g}{2V} \sum_{\mathbf{k}_1\mathbf{k}_2\mathbf{k}_3\mathbf{k}_4} a^\dagger_{\mathbf{k}_1} a^\dagger_{\mathbf{k}_2} a_{\mathbf{k}_3} a_{\mathbf{k}_4} \delta_{\mathbf{k}_1+\mathbf{k}_2,\mathbf{k}_3+\mathbf{k}_4}. \tag{3.28}$$

3.2.3 Scattering theory

To connect the effective interaction in equation (3.28) with the real interaction, we require that the approximated Hamiltonian reproduces the correct two-body scattering in vacuum.

We write the Schrödinger equation for the scattering in a centre-of-mass system with $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$, $m_r = m/2$,

$$-\frac{\hbar^2}{m}\nabla^2\psi_{\mathbf{k}}(\mathbf{r}) + V(\mathbf{r})\psi_{\mathbf{k}}(\mathbf{r}) = 2\epsilon_k\psi_{\mathbf{k}}(\mathbf{r}), \tag{3.29}$$

with the solution, using the Helmholtz equation with an outgoing-wave Green's function (Fetter and Walecka, 1971, sec. 11) as an incoming plane wave and a scattered wave,

$$\psi_{\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}} - \frac{m}{4\pi\hbar^2}\int \mathrm{d}^3r' \, \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} V(\mathbf{r}')\psi_{\mathbf{k}}(\mathbf{r}'). \tag{3.30}$$

If we assume that the potential has finite range, the asymptotic form ($|\mathbf{r}-\mathbf{r}'| \to r$) is

$$\psi_{\mathbf{k}}(\mathbf{r}) \approx e^{i\mathbf{k}\cdot\mathbf{r}} + f(\mathbf{k}',\mathbf{k})\frac{e^{ikr}}{r} \tag{3.31}$$

with the scattering amplitude for $\mathbf{k} \leadsto \mathbf{k}'$

$$f(\mathbf{k}',\mathbf{k}) \equiv -\frac{m}{4\pi\hbar^2}\int \mathrm{d}^3r' \, e^{-i\mathbf{k}'\cdot\mathbf{r}'} V(\mathbf{r}')\psi_{\mathbf{k}}(\mathbf{r}') \tag{3.32}$$

This solution is valid also for a singular potential because the wave function vanishes at the singularity.

If the potential is spherically symmetric, energy conservation requires $k' = k$, and the partial wave expansion

$$f(k,\theta) = \sum_{l=0}^{\infty} \frac{2l+1}{k} e^{i\delta_l} \sin\delta_l P_l(\cos\theta), \tag{3.33}$$

where P_l is the Legendre polynomial, and δ_l is the phase shift. In the region outside the potential the Schrödinger equation can be solved, and thus the phase shifts obtained explicitly, with the boundary condition that the wave function vanishes inside the hard core.

In the low energy limit ($k \to 0$), the phase shifts are proportional to k^{2l+1} (Landau and Lifshitz, 1958) so only the $l = 0$ term contributes significantly; it is

$$\delta_0 = -ka, \tag{3.34}$$

which also defines the s-wave scattering length a. In this limit, the scattering amplitude becomes constant,

$$\lim_{k=k'\to 0} f(\mathbf{k}',\mathbf{k}) = -a. \tag{3.35}$$

We transform equations (3.32) and (3.30) to momentum space, using the outgoing-wave Green's function

$$-\frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} = \int \frac{\mathrm{d}^3q}{(2\pi)^3} \frac{e^{i\mathbf{q}\cdot(\mathbf{r}-\mathbf{r}')}}{k^2 - q^2 + i\eta}, \tag{3.36}$$

where $\eta \to 0^+$ represents the adiabatic limit.

The scattering amplitude, equation (3.32), becomes

$$f(\mathbf{k}',\mathbf{k}) = -\frac{m}{4\pi\hbar^2}\int \frac{\mathrm{d}^3p}{(2\pi)^3}\tilde{V}_{\mathbf{k}-\mathbf{p}}\tilde{\psi}_{\mathbf{k}}(\mathbf{p}), \tag{3.37}$$

and equation (3.30) combined with equation (3.36), gives us the scattering wave function

$$\tilde{\psi}_{\mathbf{k}}(\mathbf{p}) = \underbrace{(2\pi)^3\delta(\mathbf{k}-\mathbf{p})}_{\text{incident wave}} - \underbrace{\frac{4\pi f(\mathbf{p},\mathbf{k})}{k^2 - p^2 + i\eta}}_{\text{scattered wave}}. \tag{3.38}$$

We insert this into equation (3.37) to get

$$-\frac{4\pi\hbar^2}{m}f(\mathbf{k}',\mathbf{k}) = \tilde{V}_{\mathbf{k}'-\mathbf{k}} - 4\pi\int \frac{\mathrm{d}^3p}{(2\pi)^3}\frac{\tilde{V}_{\mathbf{k}'-\mathbf{p}}f(\mathbf{p},\mathbf{k})}{k^2 - p^2 + i\eta}. \tag{3.39}$$

The off-shell energies $p^2 \neq k^2$ are included in the integral. The leading term is the first Born approximation, and iterating by inserting this approximation on the right hand side gives the first correction

$$-\frac{4\pi\hbar^2}{m}f(\mathbf{k}',\mathbf{k}) = \tilde{V}_{\mathbf{k}'-\mathbf{k}} - \frac{m}{\hbar^2}\int \frac{\mathrm{d}^3p}{(2\pi)^3}\frac{\tilde{V}_{\mathbf{k}'-\mathbf{p}}\tilde{V}_{\mathbf{k}-\mathbf{p}}}{k^2 - p^2 + i\eta} + \dots \tag{3.40}$$

In the limit $k'^2 = k^2 \to 0$, this equation becomes a perturbation expansion for the scattering length in powers of $\tilde{V}$,

$$\frac{4\pi\hbar^2 a}{m} = \tilde{V}_0 - \frac{m}{\hbar^2}\int \frac{\mathrm{d}^3p}{(2\pi)^3}\frac{|\tilde{V}_{\mathbf{p}}|^2}{p^2} + \dots, \tag{3.41}$$

with the first correction finite if $\tilde{V}_{\mathbf{p}}$ vanishes rapidly enough.

However, if we use our constant effective interaction, $\tilde{V} = g$, then equation (3.41) reduces to

$$\frac{4\pi\hbar^2 a}{m} \approx g - \frac{mg^2}{\hbar^2}\int \frac{\mathrm{d}^3p}{(2\pi)^3}\frac{1}{p^2} + \dots \tag{3.42}$$

which unfortunately is divergent in the second term. We shall see in section 3.2.6 that we can use this term to cancel a divergence in the ground state energy, thus validating our choice of potential.

3.2.4 Canonical quantisation

The presence of interactions implies that two particles scatter out of the ground state $\mathbf{k} = 0$ to $\pm\mathbf{k} \neq 0$. For an interacting Bose gas, $N_0/N < 1$ even at $T = 0$, but we assume N_0/N finite in the thermodynamical limit.

The noninteracting ground state can be written using the zero-momentum creation operator $a_0^\dagger$ as

$$|\Phi(N)\rangle = |N, 0, 0, \ldots\rangle \equiv (a_0^\dagger)^N (N!)^{-1/2}\, |0, 0, 0, \ldots\rangle, \tag{3.43}$$

thus the zero-momentum mode operators act as

$$a_0|\Phi(N)\rangle = \sqrt{N}\, |\Phi(N-1)\rangle, \tag{3.44}$$
$$a_0^\dagger|\Phi(N)\rangle = \sqrt{N+1}\, |\Phi(N+1)\rangle. \tag{3.45}$$

Both of these operators, when acting on the condensed state, have much larger effect than the commutator between them, which is $[a_0, a_0^\dagger] = 1$. Thus the commutator is insignificant for thermodynamical purposes and we can replace the raising and lowering operators with c numbers.

As long as the lowest mode is macroscopically occupied, we can extend this notion to the weakly interacting gas. We therefore let

$$a_0, a_0^\dagger \rightarrow \sqrt{N_0}. \tag{3.46}$$

The other operators, for the modes $\mathbf{k} \neq 0$, remain as they were, and we do not replace them with c numbers.

From equation (3.26), we can identify the different cases of two-particle interaction in H_{int} that are consistent with conservation of momentum:

(a) Two particles in the condensate state interact ($\mathbf{k}_1 = \mathbf{k}_2 = \mathbf{k}_3 = \mathbf{k}_4 = 0$):

$$a_0^\dagger a_0^\dagger a_0 a_0 \rightarrow N_0^2, \tag{3.47}$$

(b) a particle in the condensate state interacts with a particle that is above the condensate state($\mathbf{k}_1 = \mathbf{k}_3 = 0, \mathbf{k}_2 = \mathbf{k}_4$; $\mathbf{k}_1 = \mathbf{k}_4 = 0, \mathbf{k}_2 = \mathbf{k}_3$; $\mathbf{k}_2 = \mathbf{k}_3 = 0, \mathbf{k}_1 = \mathbf{k}_4$; $\mathbf{k}_2 = \mathbf{k}_4 = 0, \mathbf{k}_1 = \mathbf{k}_3$):

$$4 \sum_{\mathbf{k}\neq 0} a_0^\dagger a_0 a_{\mathbf{k}}^\dagger a_{\mathbf{k}} \rightarrow 2N_0 \sum_{\mathbf{k}\neq 0} (a_{\mathbf{k}}^\dagger a_{\mathbf{k}} + a_{-\mathbf{k}}^\dagger a_{-\mathbf{k}}), \tag{3.48}$$

(c) two particles are both excited from or both fall into the condensate state ($\mathbf{k}_1 = \mathbf{k}_2 = 0, \mathbf{k}_3 = -\mathbf{k}_4$; $\mathbf{k}_3 = \mathbf{k}_4 = 0, \mathbf{k}_1 = -\mathbf{k}_2$):

$$\sum_{\mathbf{k}\neq 0} (a_0^\dagger a_0^\dagger a_{\mathbf{k}} a_{-\mathbf{k}} + a_{\mathbf{k}}^\dagger a_{-\mathbf{k}}^\dagger a_0 a_0) \rightarrow N_0 \sum_{\mathbf{k}\neq 0} (a_{\mathbf{k}} a_{-\mathbf{k}} + a_{\mathbf{k}}^\dagger a_{-\mathbf{k}}^\dagger), \tag{3.49}$$

(d) a particle is excited from the condensate state and does not return, or falls into the condensate state; the other particle stays above the condensate state ($\mathbf{k}_1 = 0, \mathbf{k}_2 = \mathbf{k}_3 + \mathbf{k}_4$; $\mathbf{k}_2 = 0, \mathbf{k}_1 = \mathbf{k}_3 + \mathbf{k}_4$; $\mathbf{k}_3 = 0, \mathbf{k}_4 = \mathbf{k}_1 + \mathbf{k}_2$; $\mathbf{k}_4 = 0, \mathbf{k}_3 = \mathbf{k}_1 + \mathbf{k}_2$):

$$\begin{aligned} &2\sum_{\mathbf{k}_1\mathbf{k}_2\neq 0}(a_0^\dagger a_{\mathbf{k}_1+\mathbf{k}_2}^\dagger a_{\mathbf{k}_1}a_{\mathbf{k}_2} + a_{\mathbf{k}_1}^\dagger a_{\mathbf{k}_2}^\dagger a_{\mathbf{k}_1+\mathbf{k}_2}a_0) \\ &\qquad \rightarrow 2\sqrt{N_0}\sum_{\mathbf{k}_1\mathbf{k}_2\neq 0}(a_{\mathbf{k}_1+\mathbf{k}_2}^\dagger a_{\mathbf{k}_1}a_{\mathbf{k}_2} + a_{\mathbf{k}_1}^\dagger a_{\mathbf{k}_2}^\dagger a_{\mathbf{k}_1+\mathbf{k}_2}), \end{aligned} \tag{3.50}$$

or

(e) neither particle is in the condensate state before or after the interaction ($\mathbf{k}_1 \neq 0, \mathbf{k}_2 \neq 0, \mathbf{k}_3 \neq 0, \mathbf{k}_4 \neq 0$):

$$\sum_{\mathbf{k}_1\mathbf{k}_2\mathbf{k}_3\mathbf{k}_4\neq 0} a_{\mathbf{k}_1}^\dagger a_{\mathbf{k}_2}^\dagger a_{\mathbf{k}_3}a_{\mathbf{k}_4}. \tag{3.51}$$

We follow the Bogoliubov prescription of treating the zero-momentum mode operators as c numbers as in equation (3.46), and of only retaining the terms that are at least proportional to N_0. Thus, case (c) and (d) are dropped. We have to be a bit careful with case (c), since it does not preserve the particle number. This leaves us with the much simpler interaction Hamiltonian,

$$H_{\text{int}} \approx \frac{gN_0}{2V}\left[N_0 + 2\sum_{\mathbf{k}\neq 0}(a_\mathbf{k}^\dagger a_\mathbf{k} + a_{-\mathbf{k}}^\dagger a_{-\mathbf{k}}) + \sum_{\mathbf{k}\neq 0}(a_\mathbf{k}^\dagger a_{-\mathbf{k}}^\dagger + a_\mathbf{k}a_{-\mathbf{k}})\right]. \tag{3.52}$$

We ensure particle preservation by treating N as a parameter of the system. Later, when we look at trapped dilute Bose gases, we will introduce a Lagrange multiplier for this purpose; but here, it is simpler to use the eigenvalue of the number operator.

The number operator, written in full, is

$$\hat{N} = N_0 + \frac{1}{2}\sum_{\mathbf{k}\neq 0}(a_\mathbf{k}^\dagger a_\mathbf{k} + a_{-\mathbf{k}}^\dagger a_{-\mathbf{k}}), \tag{3.53}$$

with eigenvalue N. Replacing N_0 in equation (3.52) with N from this relation, the particle number is reintroduced as a parameter, and the full Hamiltonian of equation (3.26) becomes

$$H \approx \frac{gN^2}{2V} + \frac{1}{2}\sum_{\mathbf{k}\neq 0}(\epsilon_k + ng)(a_\mathbf{k}^\dagger a_\mathbf{k} + a_{-\mathbf{k}}^\dagger a_{-\mathbf{k}}) + ng(a_\mathbf{k}^\dagger a_{-\mathbf{k}}^\dagger + a_\mathbf{k}a_{-\mathbf{k}}), \tag{3.54}$$

This Hamiltonian is of quadratic form in $a^\dagger, a$ and can be diagonalised with a canonical transformation.

3.2.5 Canonical (Bogoliubov) transformation

We define a linear combination of the ladder operators

$$a_{\mathbf{k}} = u_k \alpha_{\mathbf{k}} - v_k \alpha^\dagger_{-\mathbf{k}} \tag{3.55}$$

$$a^\dagger_{-\mathbf{k}} = u_k \alpha^\dagger_{-\mathbf{k}} - v_k \alpha_{\mathbf{k}}, \tag{3.56}$$

with real coefficients u_k, v_k (both reduce momentum by $\hbar k$). The new operators obey the canonical commutation relations for bosons,

$$[\alpha_{\mathbf{k}}, \alpha^\dagger_{\mathbf{k}'}] = \delta_{\mathbf{k},\mathbf{k}'} \tag{3.57}$$

$$[\alpha_{\mathbf{k}}, \alpha_{\mathbf{k}'}] = [\alpha^\dagger_{\mathbf{k}}, \alpha^\dagger_{\mathbf{k}'}] = 0, \tag{3.58}$$

for $\mathbf{k} \neq 0$. We see that this requirement fixes the coefficients as

$$u_k^2 - v_k^2 = 1, \tag{3.59}$$

which we can parameterise as

$$u_k = \cosh\theta_k \tag{3.60}$$

$$v_k = \sinh\theta_k. \tag{3.61}$$

The remaining degree of freedom, θ_k, can be chosen to diagonalise H:

$$\begin{aligned} H =& \frac{1}{2} g n^2 V + \sum_{\mathbf{k}\neq 0} \left[(\epsilon_k + ng) v_k^2 - ng u_k v_k \right] + \\ &+ \frac{1}{2} \sum_{\mathbf{k}\neq 0} \left[(\epsilon_k + ng)(u_k^2 + v_k^2) - 2ng u_k v_k \right] (\alpha^\dagger_{\mathbf{k}} \alpha_{\mathbf{k}} + \alpha^\dagger_{-\mathbf{k}} \alpha_{-\mathbf{k}}) + \\ &+ \frac{1}{2} \sum_{\mathbf{k}\neq 0} \left[ng(u_k^2 + v_k^2) - 2(\epsilon_k + ng) u_k v_k \right] (\alpha^\dagger_{\mathbf{k}} \alpha^\dagger_{-\mathbf{k}} + \alpha_{\mathbf{k}} \alpha_{-\mathbf{k}}). \end{aligned} \tag{3.62}$$

The middle line of equation (3.62) contains the terms that are diagonal in the quasi-particle operators, and the last line contains the off-diagonals. The trick is to choose θ_k such that the off-diagonals vanish:

$$ng(u_k^2 + v_k^2) - 2(\epsilon_k + ng) u_k v_k = 0, \tag{3.63}$$

or

$$\tanh 2\theta_k = \frac{ng}{\epsilon_k + ng}. \tag{3.64}$$

The right hand side diverges for some n if $g < 0$, so we must limit the discussion to positive scattering lengths, meaning repulsive effective potentials.

Solving for u_k, v_k:

$$v_k^2 = u_k^2 - 1 = \frac{1}{2} \left(\frac{\epsilon_k + ng}{E_k} - 1 \right), \tag{3.65}$$

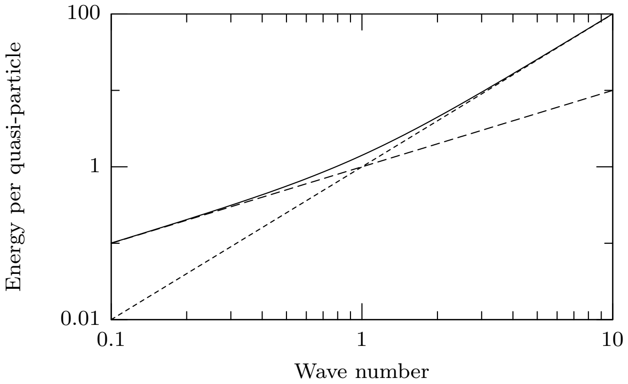


Figure 3.1: The quasi-particle spectrum (solid line), compared to a linear (dashed) and a quadratic (stippled) spectrum. The wave number is in units of $\sqrt{4ngm}/\hbar$.

with energy per quasi-particle[2]

$$E_k = \sqrt{\epsilon_k^2 + 2ng\epsilon_k}, \tag{3.66}$$

giving us the final diagonal quasi-particle Hamiltonian

$$H = \frac{1}{2}gn^2V - \frac{1}{2}\sum_{\mathbf{k}\neq 0}(\epsilon_k + ng - E_k) + \frac{1}{2}\sum_{\mathbf{k}\neq 0} E_k(\alpha_{\mathbf{k}}^\dagger\alpha_{\mathbf{k}} + \alpha_{-\mathbf{k}}^\dagger\alpha_{-\mathbf{k}}). \tag{3.67}$$

The ground state of the Hamiltonian of equation (3.67) is the quasi-particle vacuum, defined as

$$\alpha_{\mathbf{k}}|\Phi\rangle = 0 \tag{3.68}$$

for all $\mathbf{k}$ (including $\mathbf{k} = 0$).

The quasi-particle spectrum of equation (3.66) is shown in figure 3.1.

We are finally ready to estimate the ground state energy of the Bose gas.

3.2.6 Ground state energy

The ground state energy is, from the action of the quasi-particle Hamiltonian, equation (3.67), upon the ground state (3.68),

$$E_{\mathrm{G}} = \langle\Phi|H|\Phi\rangle = \frac{1}{2}gn^2V - \frac{1}{2}\sum_{\mathbf{k}\neq 0}(\epsilon_k + ng - E_k). \tag{3.69}$$

The sum is again divergent, but we handle that by adding and subtracting $n^2V/2$ times the divergence in equation (3.42), after letting the sums go to integrals.

---

[2] Notice in equations (3.62) and (3.67) that $\alpha_{\mathbf{k}}^\dagger\alpha_{\mathbf{k}}$ is the quasi-particle number operator.

The first term of equation (3.69) together with the divergent sum can now be expressed in terms of the scattering length, instead of the pseudo-potential:

$$E_{\mathrm{G}} = \frac{1}{2}n^2 V \frac{4\pi\hbar^2 a}{m} + \frac{1}{2}\int \frac{\mathrm{d}^3 k}{(2\pi)^3}\left(E_k - \epsilon_k - ng + \frac{mg^2n^2}{\hbar^2 k^2}\right), \tag{3.70}$$

and the last integral converges, as can be seen by the substitution of

$$y = \sqrt{\frac{\epsilon_k}{ng}} = \frac{\hbar k}{\sqrt{2mng}} \tag{3.71}$$

into the integral of equation (3.70) and accounting for the density of states,

$$\begin{aligned}\frac{1}{2}\int \frac{\mathrm{d}^3 k}{(2\pi)^3}\cdots &= \frac{1}{2}\int \frac{V}{2\pi^2}\frac{k^2\mathrm{d}k}{(2\pi)^3}\cdots \\ &= \frac{1}{2}n^2 V\left(\frac{nm^3}{32\pi^{10}\hbar^6}\right)^{1/2} g^{5/2}\int_0^\infty y^2\mathrm{d}y\left[\sqrt{y^4+2y^2} - y^2 - 1 + (2y)^{-2}\right].\end{aligned} \tag{3.72}$$

We handle the remaining interaction term $g^{5/2}$ by using only the leading term of equation (3.42), since the power of expansion in g is sufficient. This gives a closed form expression for the ground state energy per particle,

$$\frac{E_{\mathrm{G}}}{N} \approx \frac{2\pi\hbar^2 an}{m}\left[1 + \frac{128}{15}\left(\frac{na^3}{\pi}\right)^{1/2}\right]. \tag{3.73}$$

3.2.7 On the validity of the hard-sphere approximation

The validity of the mean field, hard sphere approximation is determined by the gas parameter na^3. Bogoliubov (1947) found that the ground state energy of a dilute gas could be expanded in powers of the square root of the gas parameter, similarly to what we have done here, and he derived the first term in this expansion. Lee et al. (1957) derived the next term, and Wu (1959) derived the final known term in the expansion, coincidentally showing that it is not a pure power series expansion but also involves logarithmic terms:

$$\frac{E_{\mathrm{G}}}{N} = \frac{4\pi\hbar^2 n^2 a}{m}\left[1 + \frac{128}{15\sqrt{\pi}}(na^3)^{1/2} + 8(\frac{4\pi}{3} - \sqrt{3})(na^3)\log(na^3) + \mathcal{O}(na^3)\right]. \tag{3.74}$$

It has been shown that the known terms of equation (3.74) are valid for any repulsive, spherical symmetrical potential; and that the unknown terms $\mathcal{O}(na^3)$ are *not* independent of the shape of the potential (Hugenholtz and Pines, 1959). Furthermore, numerical experiments show that the logarithmic term actually gives less accurate results for the solution for dense gases ($na^3 \sim 10^{-3}$), while

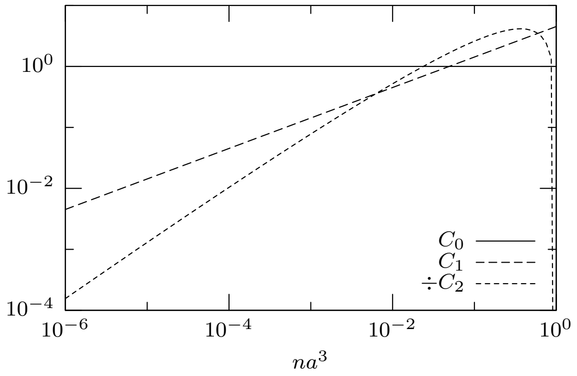


Figure 3.2: Relative size of corrections to the energy expansion. The leading term $C_0 = 1$ (solid line) is not strictly a correction, but it is included for reference. The MGP correction (dashed) becomes important at $na^3 \sim 10^{-4}$. The logarithmic correction C_2 (stippled, $-C_2$ shown) cancels C_1 at $na^3 \sim 10^{-2}$ turning the total correction negative.

the correction for less dense gases is negligible (Giorgini et al., 1999). This, and a glance at figure 3.2, suggests that our strategy of disregarding the exact shape of the potential (beyond symmetry) is effective at densities up to perhaps $na^3 \sim 10^{-2}$. At densities below $na^3 \sim 10^{-4}$ only the first term in the expansion, equation (3.73), contributes significantly.

More recently, it has been shown that the first (Bogoliubov) term is also a lower bound[3] for *any* non-negative spherical two-body potential (Lieb et al., 2000). This may explain in part why the logarithmic term is detrimental at medium-high densities, since its inclusion makes the energy go below this lower bound and is thus in this range less accurate than the uncorrected energy.

3.3 Trapped Bose gases

The previous sections treated a homogenous gas. In experimental settings, confining the gas is necessary. Usually the gas is trapped magnetically by an external quadratic magnetic field, i.e., a harmonic oscillator-like potential.

3.3.1 The Gross-Pitaevskii equation

The many-body Hamiltonian is

$$\hat{H} = \int \mathrm{d}^3 r \left[\hat{\psi}^\dagger (T + V_{\mathrm{trap}}) \hat{\psi} + \frac{V_{\mathrm{int}}}{2} \hat{\psi}^\dagger \hat{\psi}^\dagger \hat{\psi} \hat{\psi} \right] \tag{3.75}$$

---

[3] Modulo an error term, proportional to $(na^3/N)^\epsilon$, where ϵ depends on the long-range behaviour of the potential; there are also certain other restrictions.

where $\hat{\psi}$, $\hat{\psi}^\dagger$ are bosonic field operators obeying the usual commutation relations. The operator is a linear combination of all single-particle operators and corresponding wave functions,

$$\hat{\psi}(\mathbf{r}) = \sum_j \chi_j(\mathbf{r}) a_j. \tag{3.76}$$

In the framework of the correlated basis function (CBF) approach (Blaizot and Ripka, 1986), we assume that this can be separated into the uncorrelated ground state and the (small) correlation operator.

$$\hat{\psi}(\mathbf{r}) \approx \Psi(\mathbf{r}) + \hat{\phi}(\mathbf{r}). \tag{3.77}$$

Note that Ψ is a wave function, not an operator.

We then use the local density approximation (LDA) in combination with our previous results for the ground state energy of the homogenous gas, equation (3.73), to approximate this uncorrelated wave function. In the local density approximation, we take the local energy of the inhomogenous gas to be the same as the energy of a homogenous gas at the same density.

This approximation is valid if the density variations are slow compared to the range of the correlations, as is the case here.

By ignoring the correlation we assert that the density[4] is given by

$$n(\mathbf{r}) = |\Psi(\mathbf{r})|^2, \tag{3.78}$$

normalised to N,

$$N = \int \mathrm{d}^3 r \, |\Psi|^2, \tag{3.79}$$

and we use this to construct the energy functional

$$E[\Psi] = \int \mathrm{d}^3 r \left[\frac{\hbar^2}{2m} |\nabla \Psi|^2 + V_{\text{trap}}(\mathbf{r}) |\Psi|^2 + \frac{2\pi\hbar^2 a \, |\Psi|^2}{m} \left(1 + \frac{128 \, |\Psi| \, a^{3/2}}{15\pi^{1/2}} \right) |\Psi|^2 \right]. \tag{3.80}$$

We find the variational minimum of the energy, subject to the constraint that the particle number is conserved, using the traditional Lagrangian multiplier μ (which we will later identify as the chemical potential):

$$\frac{\delta}{\delta \Psi^*} \left(E[\Psi] - \mu N[\Psi] \right) = 0, \tag{3.81}$$

---

[4] In equation (3.78), $n(\mathbf{r})$ is really a distribution since it depends on the infinitesimal volume. This shall not pose any problems for us, as long as the normalisation (equation (3.79)) holds.

for arbitrary[5] variations in Ψ^*.

$$\frac{\delta}{\delta\Psi^*}\int \mathrm{d}^3r\left[A(\Psi\nabla\Psi^* + \Psi^*\nabla\Psi)^2 + (V_{\text{trap}} - \mu)\Psi^*\Psi + B(\Psi^*\Psi)^2 + C(\Psi^*\Psi)^{5/2}\right]$$
$$= \int \mathrm{d}^3r\left[A\nabla^2\Psi + (V_{\text{trap}} - \mu)\Psi + 2B\Psi^*\Psi^2 + \frac{5}{2}C(\Psi^*\Psi)^{3/2}\Psi\right] = 0 \quad (3.82)$$

using A, B, C for the coefficients in equation (3.80). This is an Euler-Lagrange equation, and the interior vanishes everywhere:

$$\left[\frac{\hbar^2}{2m}\nabla^2 + V_{\text{trap}} + \frac{4\pi\hbar^2 a}{m}|\Psi|^2\left(1 + \frac{32a^{3/2}}{3\pi^{1/2}}|\Psi|\right)\right]\Psi = \mu\Psi. \quad (3.83)$$

Equation (3.83) is known as the Gross-Pitaevskii (GP) equation, or the Modified GP when the higher-order term proportional to $|\Psi|^3\Psi$ is included.

3.3.2 Properties of the trapped Bose gas

To ease the forthcoming discussion, we assume that the external potential is in fact harmonic and axisymmetric, of the form

$$V_{\text{trap}}(\mathbf{r}) = V_{\text{HO}}(r_\perp, z) = \frac{m}{2}(\omega_\perp^2 r_\perp^2 + \omega_z^2 r_z^2) = \frac{m\omega_\perp^2}{2}(r_\perp^2 + \lambda^2 z^2), \quad (3.84)$$

where we have introduced the anisotropy parameter

$$\lambda \equiv \frac{\omega_z}{\omega_\perp}. \quad (3.85)$$

The derivations do not rely on the exact form of the external potential. The results should thus be valid also for other potentials (in particular, deformed ones), although the qualitative statements may not be.

Consider first the ideal non-interacting gas. A particle of mass m in this potential has a Gaussian ground state wave function

$$\psi(\mathbf{r}) \propto \exp\left[-\frac{1}{2}\left(\frac{r_\perp^2}{d_\perp^2} + \frac{z^2}{d_z^2}\right)\right], \quad (3.86)$$

where we have defined

$$d_\perp = \sqrt{\frac{\hbar}{m\omega_\perp}}, \quad d_z = \frac{d_\perp}{\sqrt{\lambda}}. \quad (3.87)$$

We see that the anisotropy parameter decides the shape of the gas, with $\lambda \gg 1$ squeezing the gas to a pancake like shape, and $\lambda \ll 1$ stretching it to a cigar like shape.

---

[5] We consider Ψ and Ψ^* independent variables. Variation of Ψ gives the same resulting equations, but for Ψ^* instead of Ψ.

3.3.3 Critical temperature

Define $d_0 = (d_\perp^2 d_z)^{1/3}$ and $\omega_0 = (\omega_\perp^2 \omega_z)^{1/3}$ as the geometrical agerages of the wave function dimension and trap strength, and consider the classical limit $k_\mathrm{B}T \gg \hbar\omega_0$. In this limit, the density profile approaches the Maxwell-Boltzmann form

$$n(\mathbf{r}) \propto e^{-V_\mathrm{trap}/k_\mathrm{B}T}. \tag{3.88}$$

For simplicity, we consider the case of an isotropic trap with $\lambda = 1$. The classical thermal radius is then

$$R_T^2 = \frac{2k_\mathrm{B}T}{m\omega_0^2} = d_0^2 \frac{2k_\mathrm{B}T}{\hbar\omega_0} \gg d_0^2, \tag{3.89}$$

yielding a Gaussian density profile, but with much larger radius and correspondingly lower density, $n \sim N/R_T^3$, than in the non-interacting case.

The critical temperature can be approximated by inserting this density into equation (3.23) using the thermal wavelength in equation (3.22), and we find it to be

$$k_\mathrm{B}T_\mathrm{C} \sim N^{1/3}\hbar\omega_0, \tag{3.90}$$

where N is typically large. This means that the BEC transition for a trapped gas extends well into the classical regime ($k_\mathrm{B}T_\mathrm{C} \gg \hbar\omega_0$). Condensation in a trapped gas is signaled by a spike in the particle density (as well as a much sharper spike in the momentum distribution).

3.3.4 Healing length

Looking back at the quasi-particle Hamiltonian in section 3.2.5, in particular equation (3.66), we see that the quasi-particle spectrum is described by

$$E_k = \sqrt{\epsilon_k^2 + 2ng\epsilon_k} = \begin{cases} \left(\frac{ng}{m}\right)^{1/2}\hbar k & \text{for } k \to 0 \\ ng + \frac{\hbar^2 k^2}{2m} & \text{for } k \to \infty \end{cases}. \tag{3.91}$$

The spectrum is linear in k (soundlike spectrum) for long wavelengths, and for short wavelengths it is quadratic in k (shifted free-particle spectrum). The cross-over between the two regions is when the two contributions are of the same magnitude,

$$\epsilon_k = 2ng \Rightarrow k = \frac{\sqrt{4ngm}}{\hbar}. \tag{3.92}$$

This suggests a new length scale, the length where potential and kinetic energy are balanced, such that $\xi k = 1$.

This length is called the *healing length*.[6] Using the first order approximation to g in equation (3.42), we find

$$\xi = (8\pi n a)^{-1/2}. \tag{3.93}$$

This is the length scale over which a local disturbance in the gas is smoothed back to the bulk density.

3.3.5 The Thomas-Fermi approximation

In the strong interaction limit, or for sufficiently large condensates, the ratio of kinetic to interaction energy becomes small, and the kinetic term in the Gross-Pitaevskii equation may be neglected. The resulting equation

$$[V(\mathbf{x}) + C|\Psi|^2]\Psi = \mu\Psi \tag{3.94}$$

is trivially solved by dividing out Ψ, whence we get the density distribution

$$|\Psi|^2 = \begin{cases} \frac{\mu - V(\mathbf{x})}{C} & \text{if } \mu - V > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.95}$$

The extension of the cloud, given the trap potential in equation (3.84), is then

$$R_i^2 = \frac{2\mu}{m\omega_i^2}, \quad i = \perp, z. \tag{3.96}$$

We can then use the normalisation condition, equation (3.79), to establish the eigenvalue of the Thomas-Fermi ground state,

$$\mu = \frac{15^{2/5}}{2} \left(\frac{Na}{d_0}\right)^{2/5} \hbar\omega_0. \tag{3.97}$$

Surface structure

Even for large N, the Thomas-Fermi approximation is clearly not applicable at the surface of the cloud, near R_i, where the wave function (and thus the interaction energy) vanishes, while the variation in space (and thus the kinetic energy) is high.

To find the thickness of this surface where the Thomas-Fermi approximation breaks down, we expand the wave function around a point on the surface of the

---

[6] The terms *coherence length* or *correlation length* are also used in the literature.

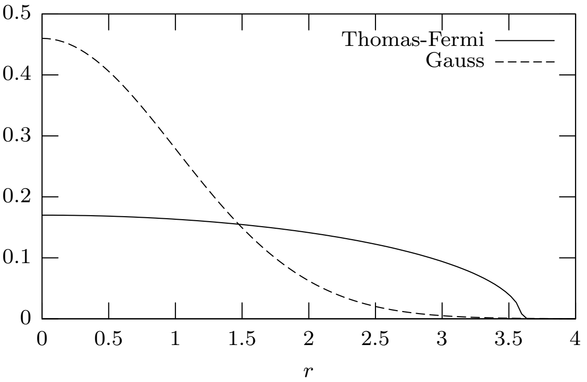


Figure 3.3: Illustration of the Gaussian and the Thomas-Fermi ground states. These are the no-interaction and strong-interaction limits, and we expect the weak-interaction ground state to lie somewhere in between.

---

cloud in the direction normal to the surface. This “radial” component of the wave function goes as

$$\psi(x) \sim \begin{cases} (x_0 - x)^{1/2}, & x \leq 0 \\ 0, & x > 0. \end{cases} \tag{3.98}$$

The kinetic energy is dominated by the derivative in the x direction,

$$\frac{\hbar^2 |\mathrm{d}\psi/\mathrm{d}x|^2}{2m|\psi|^2} \sim \frac{\hbar^2}{2m(x_0 - x)^2}, \tag{3.99}$$

for $x < 0$, while the Thomas-Fermi energy is dominated by the trap energy $V(\mathbf{x})$. We saw in equation (3.95) that $V = \mu$ at x_0, so we approximate the trap energy by

$$\mu - V(\mathbf{x}) \simeq F(x_0 - x), \tag{3.100}$$

where F is the magnitude of the trapping force acting on a particle on the surface. Equating these energies, we get a surface thickness

$$\delta = \left(\frac{\hbar^2}{2mF}\right)^{1/3}, \tag{3.101}$$

inside which the kinetic energy is dominant. For an isotropic harmonic oscillator potential, the surface thickness is

$$\delta = \left(\frac{a_{\mathrm{osc}}^4}{2R}\right)^{1/3}. \tag{3.102}$$

Although many other interesting results in the large N regime can be derived analytically in the Thomas-Fermi approximation and contrasted with the non-interacting limit, we will not dwell on it further in this report. The interested reader may consult for example (Pethick and Smith, 2001) for further analysis.

3.3.6 Scaling

For the subsequent numerical treatment of the Gross-Pitaevskii equation, it is convenient to reduce the number of parameters in the equation, leaving only those that affect the qualitative behaviour.

A natural unit length scale of the harmonic oscillator potential is $d_\perp$ as defined in equation (3.87),

$$d_\perp = \sqrt{\frac{\hbar}{m\omega_\perp}}.$$

We change integration variable in the energy functional, equation (3.80), to $r \to r/d_\perp$. It is also convenient to normalise the wave function to unity, to expose the parameter N.

We introduce the rescaled wave function ψ, defined as

$$\psi(\tilde{\mathbf{r}}) = \sqrt{\frac{d_\perp^3}{N}}\Psi(d_\perp\tilde{\mathbf{r}}). \tag{3.103}$$

We find the energy functional in the new variable $\tilde{\mathbf{r}}$ by making the following substitutions in equation (3.80). These substitutions all follow directly from equation (3.103).

$$\mathrm{d}^3r = d_\perp^3\mathrm{d}^3\tilde{r}, \tag{3.104}$$

$$\Psi = \left(\frac{N}{d_\perp^3}\right)^{1/2}\psi, \tag{3.105}$$

$$\nabla_r\Psi = \left(\frac{N}{d_\perp^5}\right)^{1/2}\nabla_{\tilde{r}}\psi. \tag{3.106}$$

The resulting energy per particle, $\tilde{E} = E/N\hbar\omega_\perp$, is

$$\tilde{E}[\psi] = \int \mathrm{d}^3r\left[\frac{1}{2}|\nabla\psi|^2 + \frac{1}{2}(x^2+y^2+\lambda^2z^2)|\psi|^2 + C\left(\frac{1}{2}+\frac{2D}{5}|\psi|\right)|\psi|^4\right], \tag{3.107}$$

where we have dropped the tilde on $\tilde{r}$. We identify the different contributions to the energy in the right hand side of equation (3.107) as the kinetic energy E_K, the potential energy E_HO, the second order mean-field interaction energy E_2, and the third order mean-field energy E_3. The dimensionless interaction parameters are

$$C = \frac{4\pi Na}{d_\perp}, \quad D = \frac{32}{3}\left(\frac{Na^3}{\pi d_\perp^3}\right)^{1/2}, \tag{3.108}$$

and the anisotropy parameter λ describes the deformation of the trap.

The corresponding Euler-Lagrange equation, which becomes the dimensionless modified Gross-Pitaevskii equation, is formally independent of the number of particles, N,

$$\left(-\frac{1}{2}\nabla^2 + \frac{1}{2}(x^2 + y^2 + \lambda^2 z^2) + C(1 + D|\psi|)|\psi|^2\right)\psi = \mu\psi, \tag{3.109}$$

but it does of course enter through the interaction parameters. With $D = 0$, equation (3.109) reduces to the ordinary Gross-Pitaevskii equation.

In the remainder of this thesis, we will use the dimensionless form exclusively.

3.4 Dynamics of the trapped Bose gas

Understanding the time-dependent behaviour of the condensate is important for understanding its physical properties and for designing experiments.

We begin by describing how the time-dependent GPE is derived and the hydrodynamical equations governing the same, and derive some fundamental properties of the dynamics.

The main reference for this section is (Pethick and Smith, 2001).

3.4.1 Time evolution of the condensate

The time-dependent Gross-Pitaevskii equation may be derived in the same way the time-dependent Schrödinger equation is derived from the time-independent one: by replacing the eigenvalue by $(i\partial/\partial t)$.

The validity of this procedure can be seen from the action principle,

$$\delta \int_{t_1}^{t_2} L \, \mathrm{d}t = 0, \tag{3.110}$$

where the Lagrangian is formed from the dimensionless energy functional, equation (3.107), which we again take to be the local energy density, and minimise in an analogous way. We find that it is indeed

$$i\frac{\partial}{\partial t}\Psi = \left(-\frac{\nabla^2}{2} + \frac{1}{2}(x^2 + y^2 + \lambda^2 z^2) + C(1 + D|\Psi|)|\Psi|^2\right)\Psi. \tag{3.111}$$

The time is in units of $\omega_\perp^{-1}$.

We can then derive an equivalent set of equations, similar to the hydrodynamic equations that govern the flow of perfect fluids.

3.4.2 Hydrodynamical formulation

Multiplying equation (3.111) by Ψ^* and subtracting its complex conjugate, we get an equation for the imaginary part,

$$\frac{\partial|\Psi|^2}{\partial t} + \nabla\left[\frac{i}{2}(\Psi\nabla\Psi^* - \Psi^*\nabla\Psi)\right] = 0. \tag{3.112}$$

We define the density $n = |\Psi|^2$, and write $\Psi = \sqrt{n}e^{i\phi}$ with the complex phase ϕ. It follows that

$$\nabla\phi = i\frac{\Psi\nabla\Psi^* - \Psi^*\nabla\Psi}{2|\Psi|^2}, \tag{3.113}$$

and equation (3.112) can be written as

$$\frac{\partial n}{\partial t} + \nabla[n\nabla\phi] = 0. \tag{3.114}$$

This is a continuity equation for the particle density n in a velocity field $\mathbf{v} = \nabla\phi$. We see that as long as ϕ has no singularities, the velocity field must be irrotational,

$$\nabla\times\mathbf{v} = \nabla\times(\nabla\phi) = 0. \tag{3.115}$$

The fluid may only rotate around a singularity in ϕ, which may occur for example at the core of a vortex where the local density vanishes.

Another consequence of the identification of the velocity with the gradient of the complex phase, is that the circulation about any point is quantised; the single-valuedness of Ψ requires that the change in ϕ around a closed curve is a multiple of 2π, and so also the circulation Γ:

$$\Gamma = \oint \mathbf{v}\cdot\mathrm{d}\mathbf{l} = \oint \nabla\phi\cdot\mathrm{d}\mathbf{l} = 2\pi l, \quad l = 0, \pm1, \pm2, \ldots. \tag{3.116}$$

If $l \neq 0$, the interior of the curve must contain at least one singularity of ϕ, and for the kinetic energy to remain finite, n must vanish at this point. This creates a quantised vortex (and, by extension, a quantised vortex line in three dimensions).

3.4.3 Vortex states

Given the irrotational nature of the condensate, how does it behave when angular momentum is added? We briefly list the qualitative answers to this question, and refer to the general reference (Pethick and Smith, 2001) for details and quantitative discussion.

The angular momentum is generally absorbed by the condensate in three ways: Vortex formation, surface modes, and centre-of-mass motion. For repulsive interactions, the lowest-energy state is mainly a superposition of vortices and surface

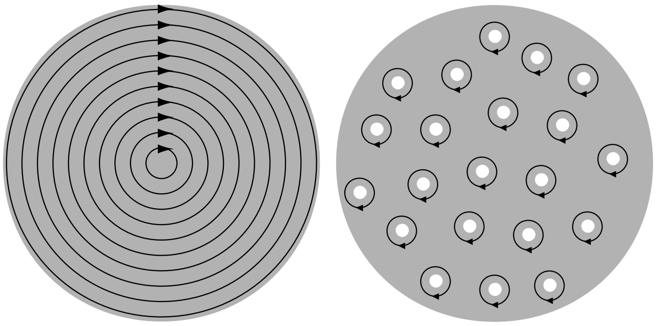


Figure 3.4: Rotation in regular and irrotational fluids. A regular fluid (a) can rotate freely, while in the irrotational fluid (b), any rotation must have a singularity at the core. For the Bose-Einstein condensate in a harmonic trap, it is energetically favourable to form several single-quantised vortices rather than a giant central vortex for a given rotation. Thus, a regular pattern of vortices is formed. Figure from (Ketterle, 2001).

waves; for attractive interactions, centre-of-mass motion is favoured. We concentrate on repulsive interactions in this thesis.

The surface wave energy goes as $\sqrt{l}$, while the single-quantised vortex energy goes as l, so the surface waves are initially favoured. But for high l the waves penetrate deeper than the surface thickness δ, and the spectrum instead goes as l^2, favouring vortices. The energy of multiply quantised vortices goes as l^n, and they are never energetically favoured in harmonic traps. A superposition of surface waves and single-quantised vortices is thus expected.

One way to add angular momentum to the condensate is to place it in a rotating anisotropic trap. A natural question is then: what is the equilibrium state, the lowest energy state, in such a trap? We transform to the frame rotating with the trapping potential.

Let $\mathbf{L}$ be the angular momentum vector and Ω be the angular velocity. The Hamiltonian is transformed to

$$H'(\mathbf{q}_\mathrm{S}, \mathbf{p}_\mathrm{S}) = H_0(\mathbf{q}_\mathrm{R}, \mathbf{p}_\mathrm{R}) - L \cdot \Omega, \tag{3.117}$$

where the subscript R denotes the canonical variables in the rotating frame, and S in the stationary frame. The energy is thus transformed as

$$E' = E - \mathbf{L} \cdot \mathbf{\Omega}. \tag{3.118}$$

A state will be favourable compared with the ground state if the angular velocity of the trap exceeds a critical value Ω_C, given by

$$\Omega_\mathrm{C} = \frac{E_L - E_0}{L}. \tag{3.119}$$

In the Thomas-Fermi regime, the critical angular velocity of a vortex state is

$$\Omega_{\mathrm{C}} \approx \frac{5}{2}\omega_0 \left(\frac{a_{\mathrm{osc}}}{R}\right)^2 \ln\left(0.671\frac{R}{\xi_0}\right), \tag{3.120}$$

while the critical angular velocity of a surface wave is higher,

$$\Omega_{\mathrm{C}} \sim \omega_0 \left(\frac{a_{\mathrm{osc}}}{R}\right)^{2/3}. \tag{3.121}$$

Chapter 4

Numerical Methods

The Gross-Pitaevskii equation of the previous chapter is a nonlinear partial differential eigenvalue problem. In general these cannot be solved analytically, and we must instead find numerical approximations. In this chapter we present the basic numerical tools of these calculations: The discretisation method of finite elements; linear algebra methods for solving systems of linear equations; methods for solving systems of nonlinear equations; how the eigenvalue problem can be handled by the nonlinear methods; and time propagation of the solution.

4.1 The finite element method

The finite element method (FEM) is a flexible method for approximating the solution of partial differential equations (PDEs) and variational problems. In quantum mechanics, the variational approach is commonly used to derive the PDEs; we did the same to arrive at equation (3.83). Thus it is natural to adopt this formulation also in the finite element method.

In the variational formulation we start with the energy functional, equation (3.80), and minimise the approximation, following what is known as the Rayleigh-Ritz variational approach. The alternative, which is to minimise a weak form of equation (3.83), is known as the Galerkin weighted residuals formulation. If the standard choice of weighting functions is used, the two approaches are equivalent (whenever a variational form exists).

The advantages of the Rayleigh-Ritz formulation are purely conceptual, since the two yield equivalent results. The apparently arbitrary choice of weighting functions becomes unnecessary, and a lot of work on stability and convergence of variational methods becomes directly available.

The contents of this chapter is found in more detail in the text book by Langtangen (2003), supported by analysis from Strang and Fix (1973). Ram-Mohan

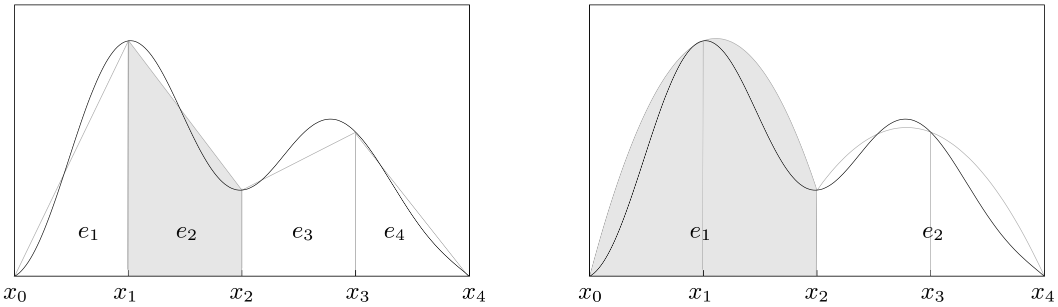


Figure 4.1: Approximation of integration by piecewise interpolation, using linear (left) and quadratic (right) polynomials. In both cases the domain is represented by five nodes $x_0, \ldots, x_4$, which creates four linear or two quadratic elements e_i. One element is shown shaded in each case.

(2002) covers the use of finite elements in quantum mechanics with several examples.

We should mention a potential point of confusion concerning the terms *elements* and *matrices*. A *matrix element* is one scalar value inside a matrix, while an *element matrix* is a matrix representing one finite element. An *element matrix element* would be a scalar value inside a matrix representing a finite element, but hopefully we have managed to avoid this particular construction.

The terms *linear elements* and *quadratic elements* are taken to mean elements defined for n-linear and n-quadratic basis functions, respectively, in n dimensions; in general, the type of an element refers to its basis functions.

4.1.1 The general approach

We split the full domain of the functional into smaller intervals, and on each interval we define an interpolating function (polynomial).

$$\int_\Omega f(\mathbf{x})\,\mathrm{d}\mathbf{x} \approx \sum_i \int_{\Omega_i} f_i(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{4.1}$$

The elements are these piecewise approximations to f. To connect to the results in chapter 3, let us write it as

$$f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) = \sum_i u_i N_i(\mathbf{x}), \tag{4.2}$$

where u_i are coefficients of the basis functions N_i, and N_i vanish outside the element (interval) they represent. We write the energy functional, equation (3.107), as

$$E[\psi] = \int \mathrm{d}^3 r \left[\frac{1}{2}|\nabla\psi|^2 + V(\mathbf{x})|\psi|^2 + g(|\psi|)|\psi|^2\right], \tag{4.3}$$

where $V(\mathbf{x}) = (x^2 + y^2 + \lambda^2 z^2)/2$ is the harmonic potential, and $g(\psi) = C(1/2 + 2D|\psi|/5)|\psi|^2$ is the nonlinear mean-field interaction. After inserting the approximation from equation (4.2) and again applying the Lagrange multiplier, equation (4.3) becomes

$$E[\psi] \approx \sum_{i,j} u_i^* u_j \int_\Omega \mathrm{d}\mathbf{x}\, \frac{1}{2}\nabla N_i \nabla N_j + (V(\mathbf{x}) + g(\psi))N_i N_j - \mu \int_\Omega \mathrm{d}\mathbf{x}\, N_i N_j, \tag{4.4}$$

The nonlinear term g presents a difficulty to us. We ignore this for now, and focus on just the linear case. In section 4.3, we look at the consequences of nonlinearity and how to find a self-consistent solution in that case.

We now define the matrix elements K_{ij} and M_{ij} to be the integrals in equation (4.4), with $g = 0$,

$$K_{ij} = \int_\Omega \mathrm{d}\mathbf{x}\, \frac{1}{2}\nabla N_i \nabla N_j + V(\mathbf{x}))N_i N_j, \quad M_{ij} = \int_\Omega \mathrm{d}, \mathbf{x}\, N_i N_j \tag{4.5}$$

Minimising equation (4.4) with respect to the coefficients u^*, we get the piecewise constraints

$$\frac{\delta E[\psi]}{\delta u_i^*} = 0 = \sum_j u_j (K_{ij} - \mu M_{ij}), \quad i = 1, \ldots, N. \tag{4.6}$$

We can assemble the factors as a vector and a matrix,

$$\mathbf{u} = [u_1\ u_2\ \ldots\ u_N]^{\mathrm{T}}, \tag{4.7}$$

$$\mathbf{K} = \begin{bmatrix} K_{11} & K_{12} & \\ K_{21} & \ddots & \\ & & K_{NN} \end{bmatrix}, \tag{4.8}$$

and similarly for $\mathbf{M}$. The constraints, equation (4.6), can then be written succinctly as

$$(\mathbf{K} - \mu\mathbf{M})\mathbf{u} = 0 \tag{4.9}$$

which is a generalised eigenvalue problem in linear algebra.

Similarity to the Galerkin formulation

In the Galerkin weighted residual formulation, we would instead start with equation (3.109) and require that the residuals vanish uniformly for a sequence of weighting functions W_j. We again set $g = 0$.

$$\int_\Omega \mathrm{d}\mathbf{x} \left(-\frac{1}{2}\nabla^2 + V - \mu\right)\left(\sum_i u_i N_i\right) W_j = 0. \tag{4.10}$$

Now, if we choose the weighting functions to be the basis functions, $W_j = N_j$, and shift one of the derivatives to the weighting functions by means of partial integration, we get

$$\sum_i u_i(K_{ij} - \mu M_{ij}) + \left[\frac{1}{2}(\nabla f)N_j\right]_{\delta\Omega} = 0. \tag{4.11}$$

The only difference between this equation and equation (4.6) is the presence of von Neumann boundary terms. We shall revisit boundary conditions in section 4.1.7.

4.1.2 Elementwise formulation

Although the previous section used a global numbering scheme for the coefficients and basis functions, this technique becomes cumbersome in higher-dimensional domains and with more complicated geometries. We therefore switch to a elementwise formulation, where an element is taken to be the domain of one piecewise polynomial. In one dimension, using linear interpolation, each element is thus the interval between two nodes.

We can then replace the global integrals, for example equation (4.5), with a sum of integrals over the elements,

$$\int_\Omega \mathrm{d}\mathbf{x}\, N_i N_j = \sum_e \int_{\Omega_e} \mathrm{d}\mathbf{x}\, N_i N_j, \tag{4.12}$$

or similarly for other integrands; but the integrand always contains $N_i N_j$ or their derivatives. For a given element, the integral over the element domain is nonzero only if the nodes i and j both belong to the element, because otherwise at least one of the piecewise polynomials vanish.

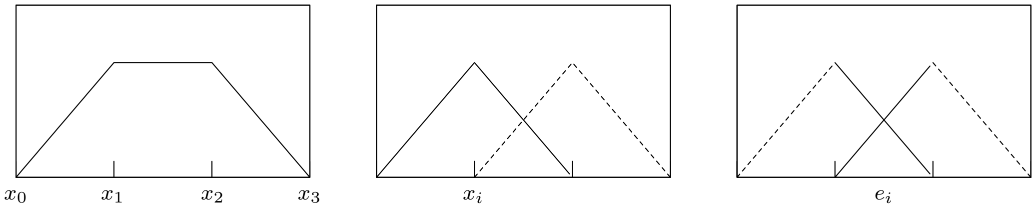
This motivates the introduction of local numbering of nodes within the element. From the local numbering, we can build a local matrix for each element where the contributions from each local basis function to each local node is computed. This local matrix is then of dimension equal to the number of nodes that each element spans, and it is dense in the sense that it contains few or no nonzeroes. Given a mapping between the local and the global numbering of nodes, we can build the global matrix from the local ones. This approach turns out to simplify calculations immensely.

We use the notation

$$u_i^{(e)} = u_{q(e,i)} \tag{4.13}$$

for local index i of element number e. Each element spans n nodes.[1] The mapping between the element-local and the global indices goes as

$$q(e,i) = (n-1)(e-1) + i. \tag{4.14}$$



(a) Global view (b) Per-node view (c) Per-element view

Figure 4.2: Different views of linear interpolation. In (a), the total interpolation is shown. The total is broken down to a per-node, or per-basis-function, view in (b). In (c), we shift to a per-element view where we consider all basis functions that contribute within a given element.

For convenience when defining the interpolating functions over the elements, and when performing the integrations, we rescale the interval to go from -1 to 1. On this interval, the basis functions can be defined uniquely, as for example in figure 4.4 (p. 48). We use $\tilde{N}_i$ to differentiate the element-local basis functions, defined on $(-1, 1)$ from the global basis functions N_i.

In one dimension, the mapping between global coordinates x and local coordinates ξ is then

$$x(\xi, e) = \frac{1-\xi}{2} x_1^{(e)} + \frac{1+\xi}{2} x_n^{(e)} = x_1^{(e)} + \frac{1+\xi}{2} h^{(e)} \tag{4.15}$$

where $h^{(e)} \equiv x_n^{(e)} - x_1^{(e)}$ is the length of the element,

We require also in the elementwise formulation, as was implicit in the global formulation, that the function is continuous at the element boundaries.[2] Thus,

$$u_n^{(e)} = u_1^{(e+1)} \tag{4.16}$$

We can now formulate the constraint, equation (4.6), elementwise,

$$\frac{\delta E[\psi]}{\delta u_i^{(e)*}} = 0 = \sum_j u_j^{(e)} (K_{ij}^{(e)} - \mu M_{ij}^{(e)}), \quad i = 1, \dots, n \quad e = 1, \dots, N \tag{4.17}$$

and the global matrices $\mathbf{K}$ and $\mathbf{M}$ can be built from the smaller element matrices $\mathbf{K}^{(e)}$ and $\mathbf{M}^{(e)}$ in a predictable way. This procedure is known as element assembly, and is illustrated in figure 4.3.

---

[1] For now, we only consider the case where all elements are of the same type.

[2] This is not strictly required, but it simplifies the formulation.

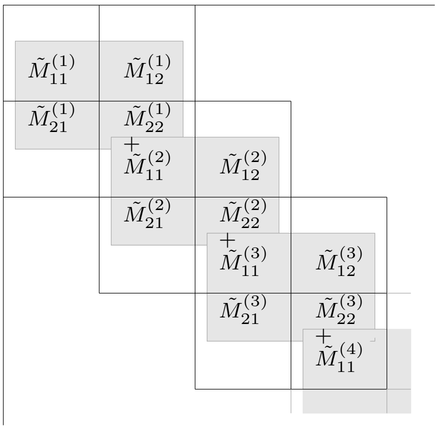


Figure 4.3: The global matrix M, assembled from element matrices $\tilde{M}^{(e)}$, using linear elements. The mixing of the diagonal is due to the continuity requirement in equation (4.16).

4.1.3 Example: The non-interacting Bose gas with linear elements

We want to build the generalised matrix eigenvalue problem

$$\mathbf{K}\mathbf{u} = \mu\mathbf{M}\mathbf{u}, \tag{4.18}$$

where $\mathbf{K}$ and $\mathbf{M}$ are defined in as in equation (4.5). In the absence of interactions, the Gross-Pitaevskii equation reduces to the harmonic oscillator equation. We calculate the matrix elements for the one-dimensional case.

The elementwise basis functions are

$$\tilde{N}_1(\xi) = (1-\xi)/2 \tag{4.19}$$

$$\tilde{N}_2(\xi) = (1+\xi)/2, \tag{4.20}$$

and we consider a uniform grid of N nodes covering the interval $(-R, R)$. The distance between nodes is thus $h = 2R/(N-1)$.

An element in the 2×2 matrix for element (e) is then

$$\tilde{K}^{(e)}_{ij} = \int_{-1}^{1} \mathrm{d}\xi \, \frac{1}{2h}\tilde{N}'_i\tilde{N}'_j + hx^2\tilde{N}_i\tilde{N}_j. \tag{4.21}$$

We have $x = x(\xi, e) = x_e + h(\xi+1)/2$; hence

$$x^2 = (h/2)^2\xi^2 + h(x_e + h/2)\xi + (x_e + h/2)^2, \tag{4.22}$$

and we calculate one element in the matrix,

$$\begin{aligned}\tilde{K}^{(e)}_{11} = \int_{-1}^{1} \mathrm{d}\xi \, \frac{1}{8h} + \frac{h}{4}\Big[(x_e + \frac{h}{2})^2 - x_e(2x_e + h)\xi \\ + (x_e^2 - hx_e - \frac{h^2}{2})\xi^2 + hx_e\xi^3 + \frac{h^2}{2}\xi^4\Big].\end{aligned}$$

All odd powers of ξ vanish, and the rest are found by substituting $\int_{-1}^{1} \mathrm{d}\xi\, \xi^k = 2/(k+1)$:

$$\tilde{K}_{11}^{(e)} = \frac{1}{4h} + h\left(\frac{2}{3}x_e^2 + \frac{1}{3}hx_e + \frac{1}{15}h^2\right) \tag{4.23}$$

The other elements of $\tilde{\mathbf{K}}^{(e)}$ are found in a similar way. We can summarise the element matrix as

$$\begin{aligned}\tilde{\mathbf{K}}^{(e)} = {} & \frac{h^3}{60}\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} + \frac{h^2(2x_e+h)}{6}\begin{bmatrix} -1 & 0 \\ & 0 & 1 \end{bmatrix} \\ & + \frac{1}{4h}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + \frac{h(2x_e+h)^2}{12}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\end{aligned} \tag{4.24}$$

While the calculations are not difficult, they quickly become tedious. In practice we usually just define the integrand of equation (4.21), and use Gaussian quadrature to approximate the integrals. Nevertheless, we observe that if the integrations can be performed analytically, they only have to be done for a single finite element, and the results are mapped to a potentially irregular and geometrically complicated grid by simple transformations.

After calculating the element matrices, we assemble the global matrix as exemplified in figure 4.3 for one dimension, and in 4.6 (p. 51) for two dimensions.

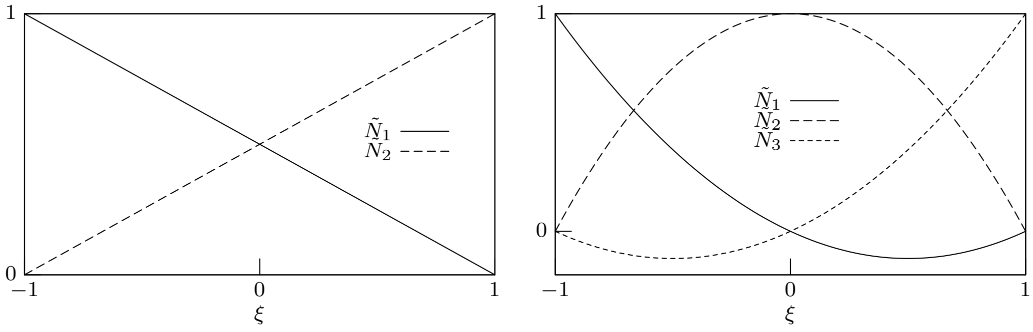
To finish this example, we look at how the global matrices are formed for a grid of 5 nodes at $x_i = -2, -1, 0, 1, 2$. We then have 4 elements, with $x_e = -2, -1, 0, 1$ and the element size is $h = 1$. The matrix $\mathbf{K}$ becomes, after element assembly,

$$\mathbf{K} = \frac{1}{60}\begin{bmatrix} 79 & 31 & & & \\ 31 & 139+19 & -9 & & \\ & -9 & 39+39 & -9 & \\ & & -9 & 19+139 & 31 \\ & & & 31 & 79 \end{bmatrix}, \tag{4.25}$$

and the mass matrix $\mathbf{M}$, which is trivial to compute, becomes

$$\mathbf{M} = \frac{1}{3}\begin{bmatrix} 2 & 1 & & & \\ 1 & 2+2 & 1 & & \\ & 1 & 2+2 & 1 & \\ & & 1 & 2+2 & 1 \\ & & & 1 & 2 \end{bmatrix}. \tag{4.26}$$

We can enforce homogenous boundary conditions by chopping off the first and last column and row. The lowest eigenvalue of the reduced matrices is $\lambda_0 = 0.81$; the exact value for this problem should be a bit above the non-bounded analytical result of 0.5.



(a) Linear elements (2 nodes)

(b) Quadratic elements (3 nodes)

Figure 4.4: Common element-local basis functions in 1 dimension. Note that the basis functions are defined such that their weight equals the value at the corresponding node: $\tilde{N}_i(\xi_j) = \delta_{ij}$ where ξ_j is the position of node j in element-local coordinates.

4.1.4 Different element types

Part of the beauty of the finite element method is that the interpolating polynomials, i.e., the element types, can be chosen to mimic known properties of the solution; smoothness, etc. Also, non-trivial geometries are possible; even infinite geometries are tractable as long as the asymptotic behaviour is known.

The simplest choice is linear elements, as depicted in figure 4.4(a). A constant interpolating function is not allowed, because we require the piecewise function to be continuous. Linear elements often give solutions that are similar[3] to those from finite difference methods. Quadratic elements, as in 4.4 (b), have the added complication of an internal node in the element. Quadratic elements reproduce linear elements as a special case, and are therefore at least as accurate as linear elements with the same number of elements.

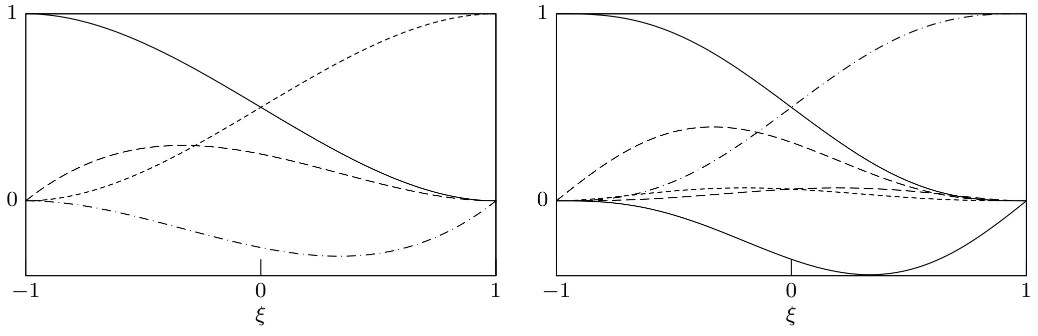
The elements discussed this far are continuous, but do not have continuous derivatives. Quantum mechanical wave functions, however, are required to be twice differentiable. This is not a major problem, since in the weak (weighted) formulation of finite elements we can transfer one of the derivatives to the weighting functions and avoid evaluating the derivatives at the discontinuities. The approximation can still be a good one.

Nevertheless, it is often better if the approximation mimics the properties of the physical solution. The finite element method is flexible enough to let us accomodate this by choosing other element types.

Let us look at a C_1-continuous polynomial with two nodes per element. For each

---

[3] Mass lumping — diagonalisation of the mass matrix — often makes the similarity explicit.



(a) C_1-continuous cubic Hermite elements (2 nodes)

(b) C_2-continuous quintic Hermite elements (2 nodes)

Figure 4.5: Hermite interpolating functions in 1 dimension with continuous derivatives, 2 or 3 degrees of freedom per node

basis function N_i in the C_0-continuous version, we require another basis function $\overline{N}_i$ to control the derivative.

$$N_i(\xi_j) = \delta_{ij} \tag{4.27}$$

$$\overline{N}_i(\xi_j) = 0 \tag{4.28}$$

$$N'_i(\xi_j) = 0 \tag{4.29}$$

$$\overline{N}'_i(\xi_j) = \delta_{ij} \tag{4.30}$$

We need a polynomial of sufficient degree, with four free parameters to satisfy these four requirements,

$$N_i(\xi) = a_i + b_i\xi + c_i\xi^2 + d_i\xi^3. \tag{4.31}$$

Combining equation (4.31) with equations (4.27) and (4.29), and setting $\xi = \pm 1$, gives a system of four equations in four unknowns for each basis function. The system is trivially solved to reveal the coefficients.

Note that in performing the element assembly, the double continuity requirement now demands an overlap of two matrix elements.

Figure 4.5 shows two of the simplest variants. Both are analogous to 4.4(a), in that they are defined for two nodes, but they are once or twice continuously differentiable. Analogues to the quadratic elements are also possible; the C_1 variant is a quintic, while the C_2 variant has 9 polynomial basis functions of degree 8.

Other commonly used element types are splines, which are especially useful for defining complex geometries, and elements that mimic known singularities in the solution (see for example Strang and Fix, 1973). Different elements may be used for different parts of the grid, thus known singularities can use one element type, while the rest of the domain uses another type.

4.1.5 Example: The non-interacting Bose gas with quadratic elements

The quadratic basis functions are defined on the interval $(-1,1)$ as

$$\tilde{N}_1 = \xi(\xi-1)/2, \tag{4.32}$$

$$\tilde{N}_2 = (1-\xi)(1+\xi), \tag{4.33}$$

$$\tilde{N}_3 = \xi(\xi+1)/2, \tag{4.34}$$

see figure 4.4(b) (p. 48).

We will not go through all the calculations of the previous example, just show what the final matrices look like, for comparison with equations (4.25) and (4.26). We do not create matrices by hand like this in the implementation, so these examples are only meant to give a feel for why they get the structure and properties that they get.

Again we use 5 nodes in the interval $(-2,2)$, but since each quadratic element spans three nodes, there are only two elements, thus $x_e = -2, 0$ and $h = 2$. Figure 4.1 (p. 42) gives a picture of the situation. Each element matrix is a 3×3 matrix.

We calculate the $\mathbf{K}$ matrix as

$$\mathbf{K} = \frac{1}{840}\begin{bmatrix} 1653 & 488 & -125 & & \\ 488 & 2608 & -408 & & \\ -125 & -408 & 309+309 & -408 & -125 \\ & & -408 & 2608 & 488 \\ & & -125 & 488 & 1653 \end{bmatrix}, \tag{4.35}$$

while the mass matrix $\mathbf{M}$ is

$$\mathbf{M} = \frac{1}{15}\begin{bmatrix} 4 & 2 & -1 & & \\ 2 & 16 & 2 & & \\ -1 & 2 & 4+4 & 2 & -1 \\ & & 2 & 16 & 2 \\ & & -1 & 2 & 4 \end{bmatrix}. \tag{4.36}$$

The lowest eigenvalue of the 3×3 matrix that results from enforcing homogenous boundary conditions at $x = \pm2$ is $\lambda_0 = 0.80$, a wee bit closer to the exact result than in the linear example. Still, with only three free nodes this does not mean much. In section 4.1.8, we shall investigate convergence in more general circumstances.

4.1.6 Finite elements in higher dimensions

We have concentrated on one-dimensional elements up to now, but it is in general no problem to move to higher dimensions. In two dimensions, common elements

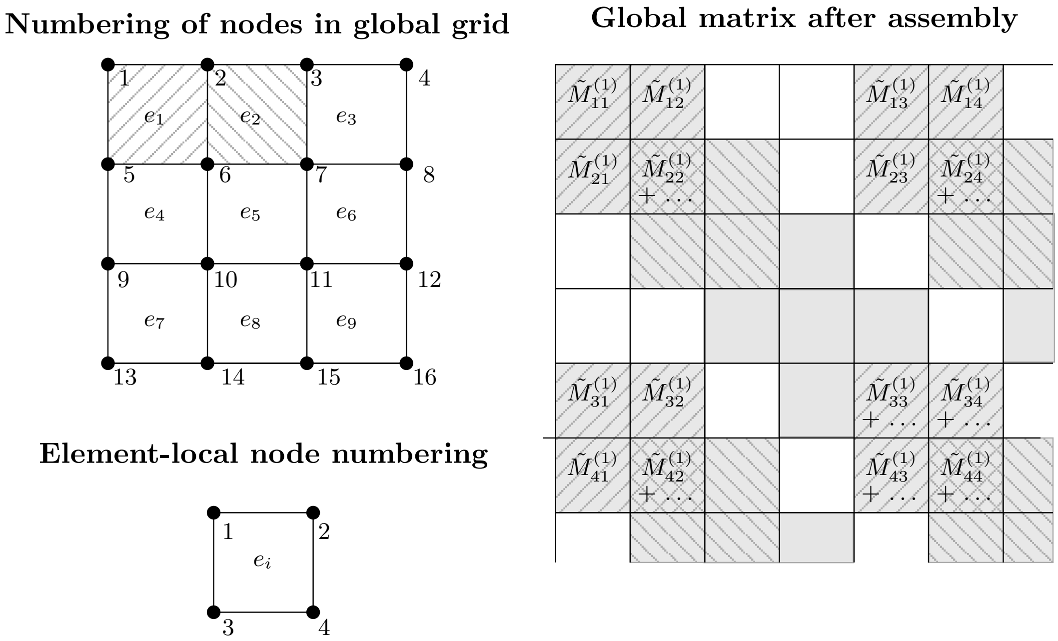


Figure 4.6: Example of the global matrix assembly in 2D using square linear elements. The elements of the local element matrices are mapped to non-consecutive locations in the global matrix, because neighbouring nodes in the y direction are not numbered consecutively. The element-local matrices are 4×4, and the global matrix is 17×17. The nonzero elements in the global matrix are shaded.

are bilinear or biquadratic squares or triangles; in three dimensions, tetrahedra and cubes. The major difference, apart from the obvious, i. e., the definition of the interpolating functions and so on, is that the element assembly process spreads out the element-local matrix when it is inserted in the global matrix, instead of copying it straight in, as in figure 4.3 (p. 46). This just reflects that two nodes can be connected without having consecutive indices, and it is seen to produce a block n-diagonal structure in figure 4.6.

4.1.7 Boundary conditions

In the Galerkin formulation, various boundary conditions enter the equations in a quite natural way. But in the current variational formulation, we are really interested in the whole of $\mathbb{R}^3$, without boundary conditions. We only require that the solution is normalisable, and has finite energy.

That means that the solution is continuous, not identically zero, and that it falls off sufficiently fast as $r \to \infty$. If we know the asymptotic form of the function, it is possible to employ special infinite boundary elements with custom integration rules (Ram-Mohan, 2002, sec. 3.7); but in practice, we get good results by restricting the solution to a large enough box or sphere with vanishing essential (Dirichlet) boundary conditions.

4.1.8 Convergence properties

We are interested in how the attractive convergence properties of the finite element method for the steady state problem carry over to the eigenvalue problem. That is, we look first at the linear problem

$$Lu = f, \tag{4.37}$$

and then at the eigenvalue problem

$$Lu = \mu u. \tag{4.38}$$

We start by introducing the terminology of a typical analysis.

The *linear operator* of order m

$$L : \mathcal{H}^m_B \to \mathcal{H}^0,$$

is a mapping between the m times differentiable function space that satisfies the given boundary conditions and the solution space.

The *inner product* of two functions is

$$(u, v) = \int_\Omega u^* v \, \mathrm{d}\mathbf{x} = (v^*, u^*).$$

The *norm*, or $\mathcal{H}^0$-norm of a function is

$$\|u\| = \|u\|_0 = (u, u)^{1/2}.$$

The $\mathcal{H}^n$-norm is

$$\|u\|_n = (\|u\| + \sum_{k=1}^{n} \|u^{(n)}\|)^{1/2}.$$

The *energy inner product* of two functions is

$$a(u, v) = (Lu, v),$$

with the associated *energy norm*

$$\|u\|_a = a(u, u)^{1/2}.$$

We list the most important properties of the steady-state problem here, without discussion or proof. An overview of the methods and results is found in (Langtangen, 2003, sec. 2.10), and a detailed analysis in (Strang and Fix, 1973).

There exists a constant C such that

$$\|u\|_2 \leq C\|f\|_0, \tag{4.39}$$

and it follows that the solution u is unique.

If u^h is the finite element approximation to u on a grid with element size h, then the error in s-norm of the approximation is bounded by some constant C such that

$$\|u - u^h\|_s \leq C[h^{k-s} + h^{2(k-m)}]\|u\|_k. \tag{4.40}$$

for interpolating polynomials of order $k-1$.

The best approximation property; the discrete solution u^h is the best approximation to the continuous solution in energy norm,

$$\|u - u^h\|_a \leq \|u - v\|_a, \tag{4.41}$$

for any v in the discrete solution space.

Convergence of the elliptic eigenvalue problem

The eigenvalues of equation (4.38) are the stationary points of the Rayleigh quotient, given as[4]

$$R(v) = \frac{a(v,v)}{(v,v)}. \tag{4.42}$$

The lowest eigenvalue is a minimum of $R(v)$,

$$\mu_1 = \min_{v \in \mathcal{H}^1_E} R(v), \tag{4.43}$$

and the other eigenvalues are minima of the subspaces that are orthogonal to the lower eigenfunctions. But a more tractable approach is known as the variational, or min-max, theorem: If $R(v)$ is maximised over all l-dimensional subspaces S_l, then the minimum possible value for this maximum is the l^{th} eigenvalue μ_l,

$$\mu_l = \min_{S_l} \max_{v \in S_l} R(v). \tag{4.44}$$

We state this principle also without proof.

---

[4] The attentive reader will notice the similarity with the variational principle as commonly used in the quantum mechanical eigenvalue problem, for example in section C.1. For the Schrödinger equation, $R(v) = \langle v|H|v\rangle / \langle v|v\rangle$; that is, the expectation value of the energy of the wave function v.

The Rayleigh-Ritz principle of eigenvalue approximation

We consider the Rayleigh quotient in the finite-dimensional subspace S^h of $\mathcal{H}^1_E$, and look for μ^h, u^h such that $\mu^h = R(u^h)$ is a critical point. If $\mathbf{K}$ and $\mathbf{M}$ are the finite element matrices introduced in equation (4.5), then it is straightforward to verify that $\mathbf{u}^\dagger\mathbf{K}\mathbf{u} = a(u^h, u^h)$ and $\mathbf{u}^\dagger\mathbf{M}\mathbf{u} = (u^h, u^h)$. This is true also for nonlinear $\mathbf{K}$. Thus the Rayleigh quotient is

$$R(u^h) = \frac{\mathbf{u}^\dagger\mathbf{K}\mathbf{u}}{\mathbf{u}^\dagger\mathbf{M}\mathbf{u}}, \tag{4.45}$$

which has critical points at $\mathbf{K}\mathbf{u} - \mu\mathbf{M}\mathbf{u} = 0$. This is a roundabout way of arriving at the previous naive discretisation of $Lu = \mu u$. But this approach yields one new insight: Since we found μ^h by minimisation over a subspace of the full solution space $\mathcal{H}^1_E$, it must be an upper bound for the true eigenvalue,

$$\mu^h_l \geq \mu_l \quad \text{for all } l. \tag{4.46}$$

Error bounds

Let P be the Rayleigh-Ritz projection operator,

$$a(u - Pu, v^h) = 0 \quad \text{for all } v^h \in S^h, \tag{4.47}$$

so that Pu is the S^h-component of u; and thus that Pu is the best approximation (in energy norm) to u in S^h. (Note that Pu is *not* equal to u^h, they are answers to slightly different questions; but they are close.) Then the standard result for stationary problems, equation (4.40) holds:

$$\|u - Pu\|_s \leq C[h^{k-s} + h^{2(k-m)}]\|u\|_k, \tag{4.48}$$

where $k-1$ is the order of the polynomial interpolation and $2m$ is the order of the linear operator. Thus an approximation of the Schrödinger equation using n-linear elements has $k = 2$ and $m = 1$.

We now turn to the error in the eigenvalues. Let E_l be the subspace spanned by eigenfunctions $u_1, \ldots, u_l$. $S_l = PE_l$ is thus spanned by $Pu_1, \ldots, Pu_l$. Let e_l be the set of all unit vectors in E_l. We can then define

$$\sigma^h_l = \max_{u \in e_l} |2(u, u - Pu) - (u - Pu, u - Pu)|. \tag{4.49}$$

Then, if $\sigma^h_l < 1$, the approximated eigenvalues are bounded above by

$$\mu^h_l \leq \frac{\mu_l}{1 - \sigma^h_l}, \tag{4.50}$$

because

$$\mu^h_l \leq \max_{u \in e_l} \frac{a(Pu, Pu)}{(Pu, Pu)} \leq \max_{u \in e_l} \frac{a(u, u)}{(u, u) - \sigma^h_l} = \frac{\mu_l}{1 - \sigma^h_l}. \tag{4.51}$$

Now we must estimate σ_l^h. The first term of equation (4.49) is

$$\begin{aligned} 2|(u, u - Pu)| &= 2|\sum_{i=1}^{l} c_i \mu_i^{-1} a(u_i - Pu_i, u - Pu)| \\ &\leq 2K \|(I - P) \sum c_i \mu_i^{-1} u_i\|_m \|(I - P)u\|_m \\ &\leq 2Kc^2 h^{2(k-m)} \|\sum c_i \mu_i^{-1} u_i\|_k \|u\|_k \\ &\leq D h^{2(k-m)} \|\sum c_i \mu_i^{k/2m-1} u_i\|_0 \|\sum c_i \mu_i^{k/2m} u_i\|_0 \\ &\leq D h^{2(k-m)} \mu_l^{k/m-1} \end{aligned} \tag{4.52}$$

where we have used the identity

$$(u, u - Pu) = \sum_{i=1}^{l} c_i \mu_i^{-1} a(u_i - Pu_i, u - Pu) \tag{4.53}$$

and the inequality

$$\|v\|_k \leq c \|L^{k/2m} v\|_0, \tag{4.54}$$

which holds for all smooth elliptic L.

The second term is higher order in h, so for sufficiently small h we subsume it in the constant D'. The eigenvalue is then bounded above by

$$\mu_l^h \leq \frac{\mu_l}{1 - D' h^{2(k-m)} \mu^{k/m-1}} \leq \mu_l + 2D' h^{2(k-m)} \mu_l^{k/m}, \tag{4.55}$$

or, including the lower bound found previously,

$$0 \leq \mu_l^h - \mu_l \leq c h^{2(k-m)} \mu_l^{k/m}. \tag{4.56}$$

The bound on the eigenfunctions can be derived in a similar fashion, but the calculations are too lengthy to include here. The argument is given in detail in (Strang and Fix, 1973, pp. 233–236). The relevant bounds are

$$\|u_l - u_l^h\|_0 \leq c' [h^k + h^{2(k-m)}] \mu_l^{k/2m}, \tag{4.57}$$

$$a(u_l - u_l^h, u_l - u_l^h) \leq c'' h^{2(k-m)} \mu_l^{k/m}. \tag{4.58}$$

Results for our model problem

We now have a quick look at our model problem, the Gross-Pitaevskii equation. In one dimension, we get the problem $Lu = \mu u$ with

$$L = H = -\frac{1}{2} \frac{\mathrm{d}^2}{\mathrm{d}x^2} + V(x) + g(|u|). \tag{4.59}$$

This is an example of an elliptic eigenvalue problem, known as a Sturm-Liouville problem. It is known to have real, well ordered eigenvalues and orthogonal eigenfunctions (Berger, 1965), and at least in the linear case, the n^{th} eigenfunction has exactly $n-1$ zero crossings.

We first consider the simplified case of $g = 0$ and constant V, with linear elements. The discretisation, from equation (4.5), yields a stiffness matrix consisting of rows

$$\mathbf{K}_j = \frac{1}{2h}\,[\,\dots\ \ -1\ \ 2\ \ -1\ \ \dots\,] + \frac{Vh}{6}\,[\,\dots\ \ 1\ \ 4\ \ 1\ \ \dots\,], \tag{4.60}$$

and a mass matrix of rows

$$\mathbf{M}_j = \frac{h}{6}\,[\,\dots\ \ 1\ \ 4\ \ 1\ \ \dots\,]. \tag{4.61}$$

We disregard the boundary conditions and look only at the interior equations. They are

$$3(-u_{i-1} + 2u_i - u_{i+1}) + qh^2(u_{i-1} + 4u_i + u_{i+1}) = \mu_l^h h^2(u_{i-1} + 4u_i + u_{i+1}). \tag{4.62}$$

It turns out that the analytical solution, $u(x) = \sin((l-1/2)x)$, also solves the discrete system exactly. We can then replace u_{i-1} and u_{i+1} in equation (4.62), using

$$\frac{u_{i-1} + u_{i+1}}{u_i} = \frac{u(x+h) + u(x-h)}{u(x)} = 2\cos((l-1/2)h). \tag{4.63}$$

The approximated eigenvalue can now be calculated directly, and we find it to be, as expected from the general results, to be

$$\mu_l^h = \mu_l + \mathcal{O}(h^2 l^4). \tag{4.64}$$

An interesting result is that lumping the mass matrix — i.e., moving off-diagonal matrix elements to the diagonal — reduces μ_l^h to a *lower* bound,

$$\mu_l^h = \mu_l - \mathcal{O}(h^2 l^4). \tag{4.65}$$

It is perhaps unlikely that this property carries over to the case of nonlinear equations, but numerical experiments with harmonic potential V at least hints that it may be valid for the harmonic oscillator case.

Finally, we derive some really coarse bounds for the full Gross-Pitaevskii equation. From the min-max theorem it follows that

$$l - \frac{1}{2} + \min_{|u|} g(|u|) \le \mu_l \le l - \frac{1}{2} + \max_{|u|} g(|u|), \tag{4.66}$$

since the linear part of the Gross-Pitaevskii equation has eigenvalues $l - \frac{1}{2}$, $l = 1, 2, \dots$, and since $|u|$ must be bounded for the energy to be finite. If $0 \le g(|u|) \le c$, then

$$l - \frac{1}{2} \le \mu_l \le l - \frac{1}{2} + c. \tag{4.67}$$

We can take c to be approximately C times the central density of the ground state. We only look at the one-dimensional results here. In the noninteracting regime, $c \approx \pi^{-1/4}C \approx 0.75C$, and in the Thomas-Fermi regime, $c \approx (9C^2/32)^{1/3} \approx 0.66C^{2/3}$. We expect the real central density to lie between these extremes.

Thus, the nonlinear eigenvalues are bounded below by the linear eigenvalues, and have the same asymptotic behaviour.

Spurious eigenvalues

We have seen in equation (4.56) that the higher discrete eigenvalues get progressively less accurate. Depending on the choice of interpolating functions, the discrete system may also have completely spurious eigenvalues. This happens when the matrices have more eigenvalues than the interpolation can represent. In the model problem, the n^{th} eigenfunction has $n-1$ zero crossing, and the discrete system of N nodes has N eigenvalues. The highest eigenvalue of the discrete system should thus have $N-1$ zero crossings to mimic the behaviour of the continuous equations.

The accuracy of these high eigenvalues is bad, but we are not bothered by that as long as they do not interfere with the solution of the lowest eigenvalues. The problem appears when the interpolating function cannot represent the full range of eigenfunctions, and spurious eigenvalues appear. This may happen if we use Hermite elements, which can only represent a function with up to $N/2$ zero crossings. In that case, at least half of the eigenvalues are spurious and we can only hope that they do not interfere with the desired eigenvalues.

A related phenomenon occurs when the eigenfunctions are complex and we solve a pair of coupled equations for the real and imaginary parts. Then the spurious eigenvalues turn up as degenerate eigenvalues, since each eigenvalue can be expressed as a pair of eigenvectors that are orthogonal in complex space. This will in general not be a problem.

4.2 Linear algebra

4.2.1 Linear solvers for large sparse systems

The topic of methods for solving linear equations is an enormous one. We can only try do cover a small corner, and none of it comprehensively. Thus this section aims to explain the methods that we are going to use in the numerical experiments, along with their roots, and also a short selection of popular or promising methods. More details on implementation of direct methods can be

Table 4.1: Properties of the matrices used in the Gross-Pitaevskii equation. The linear systems that are produced are linear combinations of these, and no single solver is optimal for all cases.

Matrix	Coupled real equations	Complex equations
$\mathbf{M}$	Symmetric, positive definite	Real symmetric, positive definite
$\mathbf{K}$	Nonsymmetric, positive definite	Hermitian, positive definite
$i\mathbf{K}$	Nonsymmetric, indefinite	Anti-hermitian, indefinite
$\delta\mathbf{K}, i\delta\mathbf{K}$	Nonsymmetric, indefinite	—

found in (Press et al., 1992), and of iterative methods in (Barrett et al., 1994); both are available on-line with thorough references. An overview of iterative, multigrid and domain decomposition methods is also found in (Langtangen, 2003, app. C).

Throughout this section we shall consider the solution of a system

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{4.68}$$

where the coefficient matrix $\mathbf{A}$ and the vector $\mathbf{b}$ are known and constant. When we have an approximation $\mathbf{x}$ to the true solution, we can also define the residual as a measure of the error in the approximation,

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}. \tag{4.69}$$

The residual obviously vanishes for any $\mathbf{x}$ that satisfies equation (4.68).

Direct solvers

Direct methods solve the system of equations explicitly, in a predictable number of steps. In practice, the only direct method is Gaussian elimination and variations of it. Gaussian elimination is exact and deterministic, although it may be numerically unstable for near-singular matrices — a problem which is only partly alleviated by pivoting. It is also in general very inefficient for sparsely structured matrices.

There is one case, however, where Gaussian elimination is a competitive method: When the matrix is banded with a small bandwidth. In the finite element method, this structure is produced in the one-dimensional case. The complexity is then optimal, $\mathcal{O}(N)$ (Press et al., 1992, p. 50).

We assume that the reader is familiar with Gaussian elimination, and we will not expound further on it. The reference text (Press et al., 1992) can be consulted for analysis.

Classical iterative solvers

For general sparse systems, we must turn to other methods. Iterative methods produce successive, hopefully improving, approximations at each step of the method.

Classical iterative solvers are characterised by the fact that one iteration of the method can be written on the form

$$\mathbf{x}^k = \mathbf{G}\mathbf{x}^{k-1} + c, \tag{4.70}$$

for some $\mathbf{G}$ and $\mathbf{c}$ derived from $\mathbf{A}$ and $\mathbf{b}$ of equation (4.68). Since G and c are constant, these methods are also known as stationary methods. Note that the actual implementation is usually not done in this way, but instead in an explicit loop. But it eases discussion and analysis, and in these simple cases it is rather straightforward to go from the matrix formulation to an efficient implementation.

We look at the iteration methods that are produced by splitting the coefficient matrix into two parts, $\mathbf{A} = \mathbf{V} - \mathbf{W}$,

$$\mathbf{V}\mathbf{x} = \mathbf{W}\mathbf{x} + \mathbf{b}, \tag{4.71}$$

justifying the iteration

$$\mathbf{V}\mathbf{x}^k = \mathbf{W}\mathbf{x}^{k-1} + \mathbf{b}, \tag{4.72}$$

which is on the same form as equation (4.70), with $\mathbf{G} = \mathbf{V}^{-1}\mathbf{W}$ and $\mathbf{c} = \mathbf{V}^{-1}\mathbf{b}$.

By induction we find that the error after k iterations is

$$\mathbf{x}^k - \mathbf{x} = \mathbf{G}^k(\mathbf{x}^0 - \mathbf{x}), \tag{4.73}$$

thus the convergence depends on the spectral radius, defined as the modulus of the highest eigenvalue, of the iterator $\mathbf{G}$.

The different classical iterative methods are found by splitting $\mathbf{A}$ in different ways. Let us write $\mathbf{A}$ as a sum of the lower-diagonal $\mathbf{L}$, the diagonal $\mathbf{D}$ and the upper-diagonal $\mathbf{U}$:

$$\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}. \tag{4.74}$$

Then the *Jacobi method* is generated by $\mathbf{V} = \mathbf{D}$, $\mathbf{W} = -\mathbf{L} - \mathbf{U}$:

$$\mathbf{D}\mathbf{x}^k = -(\mathbf{L} + \mathbf{U})\mathbf{x}^{k-1} + \mathbf{b}, \tag{4.75}$$

which is simple to compute, but converges slowly.

The *Gauss-Seidel method* is

$$(\mathbf{D} + \mathbf{L})\mathbf{x}^k = -\mathbf{U}\mathbf{x}^{k-1} + \mathbf{b}, \tag{4.76}$$

where the left-hand side is not diagonal, but still solvable by an explicit method. In contrast with the Jacobi method, where the order in which the unknowns are solved does not matter, the Gauss-Seidel method must be solved sequentially, making it less usable in parallel computations.

For elliptic partial differential equations discretised on a grid with mesh width h, the spectral radius of $\mathbf{G}$ for both these methods is $1-\mathcal{O}(h^2)$ (Barrett et al., 1994, p. 9), thus making convergence slower for larger grids. The expected number of iterations to get below some error threshold is thus $\mathcal{O}(h^{-2})$. The Gauss-Seidel method converges faster than the Jacobi method, but only by a constant (with respect to h) factor.

The *SOR method* and its symmetric variant, the *SSOR method*, are variations of the Gauss-Seidel method, where each step is amplified by a factor ω between 0 and 2. It is remarkable that in some cases an optimal choice of ω can reduce the spectral radius to $1 - \mathcal{O}(h)$ (Langtangen, 2003, p. 760), but there is no general prescription for finding the optimal value for ω.

Although these methods are not used as standalone solvers because of their slow convergence, they serve well as preconditioners for other solvers; see section 4.2.1. Their strong high-frequency damping behaviour makes them especially attractive for multigrid methods, as smoothers.

Conjugate gradient-like iterative methods

Non-stationary methods, like the conjugate gradient family of methods, differ from their classical counterparts in that the computations in each iteration are not constant, but change in response to previous iterations. A typical use of this extra information is to make the corrections orthogonal in some sense; this typically requires computing inner products in addition to the matrix-vector products of the classical iterative methods.

We start again with a system $\mathbf{Ax} = \mathbf{b}$ of N unknowns. If we propose a solution as

$$\hat{\mathbf{x}} = \mathbf{x}^0 + \sum_{i=1}^{N} \alpha_i \mathbf{q}^i, \tag{4.77}$$

for some initial vector $\mathbf{x}^0$ and N correction vectors $\mathbf{q}^i$ with weights α_i, then it is clear that as long as the correction vectors are not linearly dependent there exists a combination of α_i for which $\hat{\mathbf{x}} = \mathbf{x}$. This combination can in principle be found by the minimisation of the associated residual,

$$\hat{\mathbf{r}} = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}} = \mathbf{r}_0 - \sum_{i=1}^{N} \alpha_i \mathbf{A}\mathbf{q}^i, \tag{4.78}$$

in some norm, at which $\hat{\mathbf{r}} = \mathbf{r} = 0$.

The method of conjugate gradients offers a way to find the correction vectors and weights iteratively. It was first proposed as a direct solver which finds the exact solution in N steps, but it was only when it was discovered that it also has the property of strongly reducing the residual in far fewer than N steps that it gained popularity.

The conjugate gradient method. We consider only the non-preconditioned variant of the method. The iteration step is

$$\mathbf{x}^k = \mathbf{x}^{k-1} + \alpha_k \mathbf{q}^k, \tag{4.79}$$

which at the N^{th} step equals equation (4.77). The correction vector $\mathbf{q}^k$ is chosen from the Krylov subspace

$$V_k = \operatorname{span}\{\mathbf{r}^0, \mathbf{A}\mathbf{r}^0, \mathbf{A}^2\mathbf{r}^0, \ldots, \mathbf{A}^{k-1}\mathbf{r}^0\}, \tag{4.80}$$

such that it is $\mathbf{A}$-orthogonal to all previous $\mathbf{q}^i$.

We require $\mathbf{A}$ to be positive definite and symmetric.[5] Positive definiteness means that $(\mathbf{x}, \mathbf{y})_\mathbf{A} = \mathbf{x}^\mathrm{T}\mathbf{A}\mathbf{y}$ is an inner product, and hence $|\mathbf{x}|_\mathbf{A} = (\mathbf{x}, \mathbf{x})_\mathbf{A}$ is a norm; thus the minimisation property can be fulfilled. Symmetry allows construction of an orthogonal basis for the Krylov space, and thus of the sequence of correction vectors, with only minimal stored information. If we orthogonalise $\mathbf{q}^k$ with respect to $\mathbf{A}\mathbf{q}^{k-1}$, and $\mathbf{r}^k$ with respect to $\mathbf{r}^{k-1}$, then orthogonality to all earlier vectors follows automatically (Barrett et al., 1994).

We list the iterative steps, without further explanation. For the derivation, the references should be consulted. We start with an initial guess $\mathbf{x}^0$ and $\beta_0 = 0$. Then $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0$. For $k = 1, 2, \ldots$,

$$\mathbf{q}^k = \mathbf{r}^{k-1} + \beta_{k-1}\mathbf{q}^{k-1}, \tag{4.81}$$

$$\alpha_k = \frac{|\mathbf{r}^{k-1}|}{|\mathbf{q}^k|_\mathbf{A}}, \qquad \text{Minimises } (\mathbf{r}, \mathbf{A}^{-1}\mathbf{r}), \tag{4.82}$$

$$\mathbf{x}^k = \mathbf{x}^{k-1} + \alpha_k\mathbf{q}^k, \tag{4.83}$$

$$\mathbf{r}^k = \mathbf{r}^{k-1} - \alpha_k\mathbf{A}\mathbf{q}^k, \tag{4.84}$$

$$\beta_k = \frac{|\mathbf{r}^k|}{|\mathbf{r}^{k-1}|}, \qquad \text{Ensures orthogonality of } \mathbf{q}^k \text{ and } \mathbf{A}\mathbf{q}^{k-1}. \tag{4.85}$$

We count three vector additions, one matrix-vector product, and two extra inner products per iteration. This is more work than the classical methods, by about two additions and two inner products, but the extra work per iteration is more than made up for by the improved convergence.

Convergence properties. It can be shown that an error bound for the conjugate gradient method for symmetric positive definite matrices is given in terms of the spectral condition number $\kappa_2 = \lambda_{\max}/\lambda_{\min}$, as

$$|\mathbf{x}^k - \mathbf{x}|_\mathbf{A} \le 2\gamma^k|\mathbf{x}^0 - \mathbf{x}|_\mathbf{A}, \tag{4.86}$$

---

[5] Or, if the linear system is complex, it must be Hermitian.

for $\gamma = (\sqrt{\kappa_2} - 1)/(\sqrt{\kappa_2} + 1)$. Hence the error reduction in energy norm per iteration is γ, and the number of iterations to reduce the error by a given factor is proportional to $\sqrt{\kappa_2}$. For elliptic second order partial differential equations, the spectral condition number of $\mathbf{A}$ is of the same order as its spectral radius, (Barrett et al., 1994, p. 16)

$$\kappa_2 = \mathcal{O}(h^{-2}), \tag{4.87}$$

thus the number of iteration to reduce the error to an acceptable level is $\mathcal{O}(h^{-1})$. This is the same as the SOR method of the previous section, but without the difficulty of finding the optimal value for ω. For suboptimal ω, it is one order better.

Conjugate gradients on the normal equations. We can construct a symmetric positive definite matrix from any non-singular square matrix by squaring it. That is, the system

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b} \tag{4.88}$$

can be solved by the above method for the same $\mathbf{x}$. But the process also squares the spectral condition number, and hence the required number of iterations. We must therefore look for alternative methods to solve nonsymmetric or indefinite systems. Table 4.1 (p. 58) shows the different matrix types that we encounter when solving the Gross-Pitaevskii equation.

If we allow the coefficient matrix to be indefinite, two problems arise. Firstly, the denominator of equations (4.82) and (4.85) may be 0, leading to a breakdown of the process; secondly, $\mathbf{A}$ does not define a norm, thus the minimisation in equation (4.82) is not well-defined. If we allow the matrix to be nonsymmetric, the orthogonal basis cannot be formed from just the previous residual, as in equation (4.81). The remaining iterative methods deal with these problems.

The generalised minimal residual method. The GMRES method forms the orthogonal vectors $\mathbf{q}^k$ explicitly by Gram-Schmidt orthogonalisation, instead of a simple recurrence relation. The magnitude of the corrections, the α_k above, are not calculated until the end, when they are found by a global minimisation of the residual in the 2-norm.

These tweaks deal with the troublesome parts of the conjugate gradient method when applied to nonsymmetric indefinite systems. The price we pay is that all computed correction vectors must be retained. This makes it impractical, and restarted versions are used. The choice of when to restart then becomes important, as too frequent restarts may lead to slow convergence or lack of such, and too infrequent restarts may cause excessive memory consumption and work.

We do not list the iterative steps for the GMRES method, but use the operation count from (Barrett et al., 1994) for comparison with the other methods. Each iteration does $k+1$ inner products and vector additions in the k^{th} iteration, and one matrix-vector product. In the first iteration this is about the same number of operations as the conjugate gradient method, but it quickly becomes more expensive.

The bi-conjugate gradients method. BiCG sidesteps the problem of creating an orthogonal basis for nonsymmetric systems. Instead an additional sequence of vectors is created, and orthogonality is enforced between this sequence and the correction vectors. The sequences $\mathbf{q}^k$ and $\mathbf{r}^k$ are formed as in the conjugate gradient method, but their counterparts are instead formed by multiplication by $\mathbf{A}^{\mathrm{T}}$. The two sequences are mutually orthogonal in the same sense,

$$(\tilde{\mathbf{r}}^i, \mathbf{r}^j) = (\tilde{\mathbf{q}}^i, \mathbf{q}^j)_{\mathbf{A}} = 0 \quad \text{if } i \neq j. \tag{4.89}$$

This eliminates the increasing storage and computation of the GMRES method, but does not provide a true minimisation of the residual. This leads irregular convergence behaviour. Few theoretical results on the convergence are known.

An iteration of BiGC is a bit more expensive than a conjugate gradient iteration, by three vector additions and one transposed matrix product. Depending on the storage used for the matrix, a transposed matrix multiplication may be less efficient than a normal one.

Bi-conjugate gradients stabilised. Looking at the updates to the residual vector, equation (4.84), we see that it can be written instead as

$$\mathbf{r}^k = P_k(\mathbf{A})\,\mathbf{r}^0 \tag{4.90}$$

with P_k being a k^{th} degree polynomial in $\mathbf{A}$, and the same polynomial satisfies the bi-orthogonal sequence, $\tilde{\mathbf{r}}^k = P_k(\mathbf{A}^{\mathrm{T}})\,\tilde{\mathbf{r}}^0$. A method known as conjugate gradients squared (CGS) applies the operator P_k twice, leading to a convergence which is often faster than BiCG but highly irregular. The BiCG-stab method instead applies (in addition to P_k) another polynomial of the same degree that is a steepest descent update, minimising the local residual. This local minimisation, which is similar to repeated application of GMRES, makes the convergence smoother.

The operation count of BiCG-stab is two additional inner products and one additional vector addition compared to BiCG, but no transposed matrix-vector product is needed. Instead two ordinary matrix-vector products are formed.

Preconditioning

The convergence of the iterative methods depend on the spectral properties of the coefficient matrix. By multiplicating the system with a preconditioner, we can make these spectral properties more favourable. If $\mathbf{M}$ is a matrix that approximates $\mathbf{A}$, we can transform the system into

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}, \tag{4.91}$$

with the same solution $\mathbf{x}$. If calculating $\mathbf{M}^{-1}$ is significantly faster than solving the unpreconditioned system, this procedure may be effective.

Common preconditioners include Jacobi preconditioning, which is just the diagonal of the coefficient matrix, and is trivially parallelisable, and SSOR preconditioning which is less so. Various incomplete factorisation methods are also popular, such as the ILU(0) factorisation which discards any fill-ins during the factorisation.

Other solvers

While iterative methods are popular and in general quite effective, we should mention some other methods with attractive properties. Because they are not used in this thesis, we do not describe them in detail, but refer instead to the references.

Domain decomposition methods do their work by dividing the physical domain of the problem into smaller subdomains. The problem is then solved independently or nearly so on each subdomain. Domain decomposition methods can be very effective on sequential computers, but their main attractiveness lies in the ease of parallelisation. They are covered briefly in the main references, while (Cai, 2003) is a more detailed treatment in a finite element context.

Multigrid methods take advantage of the strong reduction of high-frequency errors shown by the classical iterative methods by transferring the solution between grids of different scales. A low-frequency error component on a fine grid becomes high-frequency on a coarse grid. Multigrid methods can achieve optimal $\mathcal{O}(N)$ complexity. A high-level description of multigrid methods is found in the main references, and a more in-depth treatment is found in (Mardal et al., 2003), again in the context of the finite element method.

These methods are frequently used as preconditioners for iterative methods, or vice versa, as their strong points tend to complement each other.

4.2.2 Eigensolvers

The linear eigenvalue problem can be regarded as a special case of the nonlinear problem,

$$\mathbf{B}(\mathbf{u})\mathbf{u} = 0, \tag{4.92}$$

where $\mathbf{B}(\mathbf{u}) = \mathbf{K} - R(\mathbf{u})\mathbf{M} = \mathbf{K} - (\mathbf{u}^{\mathrm{T}}\mathbf{M}\mathbf{u})^{-1}(\mathbf{u}^{\mathrm{T}}\mathbf{K}\mathbf{u})\mathbf{M}$. As such, they can be treated by the same methods.[6] We shall discuss these methods in section 4.3.

---

[6] Albeit sometimes under different names; the nonlinear method of *Picard iterations* is for example better known as the *inverse power method* when used in linear eigenvalue problems.

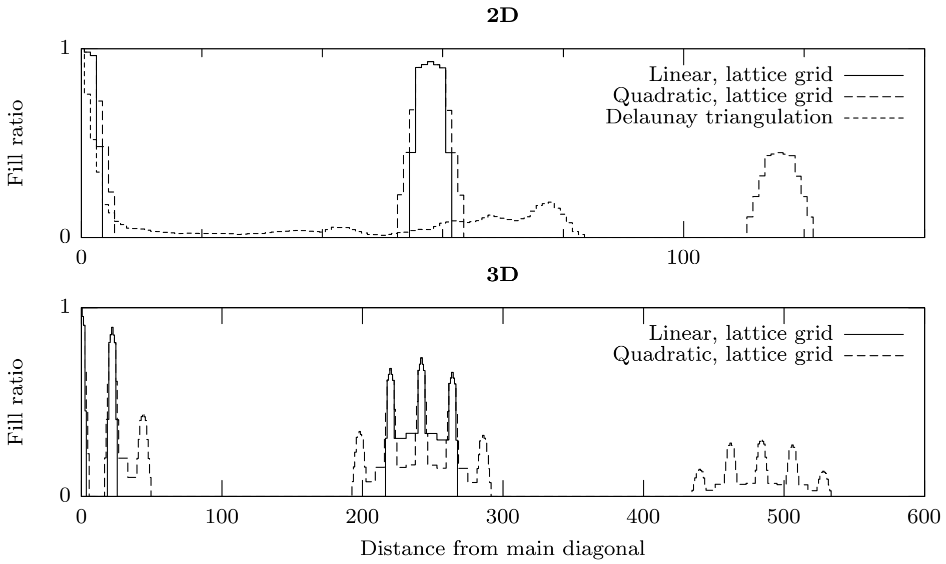


Figure 4.7: Ratio of nonzero matrix elements for a few different grid choices, in 2D (top) and 3D (bottom). The fill ratio describes the density of nonzero matrix elements along the diagonal at a given distance from the main diagonal. The grids have two degrees of freedom per node, and are chosen to have about the same resulting matrix dimensions: 28 × 28 nodes for the 2D lattices, 1565 triangles for the 2D Delaunay triangulation, and 10 × 10 × 6 nodes for the 3D lattices.

For the linear problem, block methods can be used to find several eigenvalue/eigenvector pairs efficiently. Examples of these are the inverse block power method, or the Arnoldi and Lanczos Krylov space methods. These methods are not central in this thesis, and we will not cover them in any detail.

4.2.3 Matrix representations and properties

The representation we use for matrices is an important consideration, because it is a major factor both for the speed of solving the linear system and for the memory requirements. The matrices produced by the finite element method consist mostly of zeroes; in particular, a matrix element A_{ij} is nonzero only if the nodes i and j belong to the same finite element. Figure 4.7 shows how this is reflected in the matrix structure.

Dense matrices are the most straightforward representation, in that all matrix elements are stored whether they are zero or not. It is simple to handle for a direct linear solver, which can produce new nonzero elements. They are also unusable except in trivial one-dimensional problems.

Banded matrices store a band of diagonals around the central diagonal. The one-sided width of this band is referred to as its bandwidth. If the nonzero values are clustered around the diagonal — as it is in one-dimensional problems,

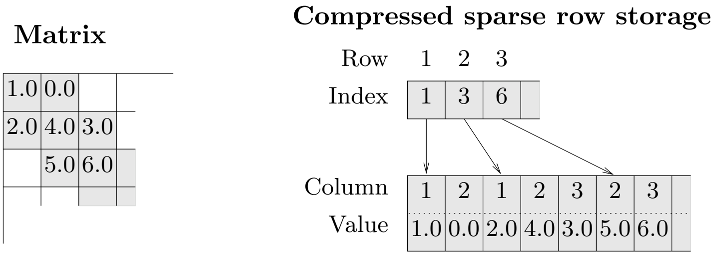


Figure 4.8: Example of a general sparse matrix representation optimised for row-by-row access. The matrix-vector product can be calculated optimally with this storage scheme. The overhead is one index per row, plus one index per stored matrix element.

because spatially neighbouring nodes are also adjacent in the matrix — this is an efficient storage format. In higher dimensions, the bandwidth becomes too large, about $N^{1/2}$ for an $N \times N$ matrix in two dimensions and $N^{2/3}$ in three dimensions.

Sparse diagonal matrices are similar to banded matrices, but can efficiently represent a band with empty diagonals; that is, the diagonals do not have to lie adjacent to each other. In some special (although not uncommon in practice) cases, this works well: With a lattice grid used for space discretisation, linear finite elements and only one degree of freedom per node, the nonzero matrix elements lie on a small number of dense diagonals. If higher order elements are used, however, the diagonals have a lot of zeroes in them; and if the grid is irregular then all diagonals inside the bandwidth must be stored.

General sparse matrices associate an index pair with every stored value, thus not imposing any structure on the matrix. There are a few drawbacks: The storage of the indices requires almost as much space as the storage of the data, and a lookup in the matrix requires searching for the index. Common implementations of general sparse matrices, such as the compressed sparse row (CSR) format illustrated in figure 4.8, focus on making sequential row-by-row access fast.

A hybrid of the general sparse format and the banded or sparse diagonal format may be used to exploit partial structure of the matrix to reduce the space overhead. In particular, the main diagonal is always completely filled, and some of the lesser diagonals may be worthwhile to store separately as well.

We take the time here to mention a characteristic of modern processors: A memory access is so much more expensive than a floating-point calculation that the computational cost of a calculation is well approximated by how much memory it needs to access.[7] For example, a complex matrix-vector product requires four times as many floating-point operations as a real matrix-vector product; but it takes only about twice as long, proportional to the memory requirements. That is why we should care about memory overhead even before the problems get too

---

[7] This is true as long as the memory access pattern is mainly sequential.

Table 4.2: Estimated memory requirements for the various matrix representations, in units of the size of one floating point number. We consider a grid with a total of N nodes in d dimensions, using elements of order $p = 1$ for linear, $p = 2$ for quadratic elements. We assume one degree of freedom per node, and, for general sparse matrices, that one index is half the size of one matrix element.

	Lattice grid	Irregular grid
Dense	N^2	Unchanged
Banded	$N(2pN^{\frac{d-1}{d}} + 1)$	Somewhat higher, see figure 4.7
Sparse diagonal	$N(2p+1)^d$	Same as banded
General sparse	$\sim \frac{3}{2}N(p+2)^d$	Unchanged

large to fit in core. Table 4.2 shows estimates of memory requirements for the different matrix representations.

Since we aim to solve two- and three-dimensional systems on possibly irregular grids, we will use the general sparse storage scheme exclusively in the implementation.

4.2.4 Continued example: The non-interacting Bose gas

The discretised system for the Gross-Pitaevskii equation is, we recall,

$$\mathbf{K}\mathbf{u} = \mu\mathbf{M}\mathbf{u},$$

where $\mathbf{u}$ and μ are unknown. $\mathbf{K}$ and $\mathbf{M}$ are defined as in equation (4.5). In section 4.1.3, we looked at how this system was discretised by the finite element method. We now look at a few properties of the discretised system.

Scaling

The norm of the wave function is 1. In the FEM framework, the norm is

$$\int |\psi|^2 \,\mathrm{d}\mathbf{x} = \int \sum_i u_i^* N_i \sum_j u_j N_j \,\mathrm{d}\mathbf{x} = \sum_{ij} u_i^* M_{ij} u_j = \mathbf{u}^\dagger \mathbf{M}\mathbf{u}, \tag{4.93}$$

hence the normalisation requirement translates as

$$\mathbf{u}^\dagger \mathbf{M}\mathbf{u} = 1. \tag{4.94}$$

Hamiltonian

We can then find the chemical potential (or the eigenvalue, if the state is an eigenstate) by calculating the expectation value of the Hamiltonian,

$$\mu = \langle H \rangle = \int \psi^* H \psi dx = \mathbf{u}^\dagger \mathbf{K} \mathbf{u}, \tag{4.95}$$

in a similar way as when we found in equation (4.93) that $\langle I \rangle = \mathbf{u}^\dagger \mathbf{M} \mathbf{u}$.

These are examples of a general rule, namely that we can substitute the quantum mechanical operators, acting on states, with matrices acting on vectors, by the simple substitutions

$$I \to \mathbf{M}, \quad H \to \mathbf{K}, \quad |\psi\rangle \to \mathbf{u}, \quad \langle \psi | = \mathbf{u}^\dagger. \tag{4.96}$$

Such matrices could also be defined for other observables of the system, such as the coordinate operator

$$X_{ij} = \int x N_i N_j \, \mathrm{d}\mathbf{x}, \tag{4.97}$$

the momentum operator

$$(P_x)_{ij} = -i \int N_i \frac{\partial N_j}{\partial x} \, \mathrm{d}\mathbf{x}, \tag{4.98}$$

the angular momentum operator

$$(L_z)_{ij} = -i \int \left(x N_i \frac{\partial N_j}{\partial y} - y N_i \frac{\partial N_j}{\partial x} \right) \mathrm{d}\mathbf{x}, \tag{4.99}$$

and so forth. But unless the matrix is required for solving the system, or it is used many times and thus is stored for efficiency reasons, it is normally easier to perform the integration explicitly.

4.3 Nonlinear equations

The nonlinear eigenvalue problem can be written as

$$\mathbf{F}(\mathbf{u}) = (\mathbf{K}(\mathbf{u}) - \mu \mathbf{M})\mathbf{u} = 0, \tag{4.100}$$

where μ can be written as $R(\mathbf{u})$ using the Rayleigh quotient, thus transforming it into a steady-state problem. In general, we cannot solve this equation directly; instead, we make a linear approximation that we solve self-consistently. This means that we solve the linearised system repeatedly until we get a solution that is consistent, i.e., it also solves the nonlinear system. We will describe various methods for finding this self-consistent solution. We focus on the discussion of the eigenvalue problem rather than on general nonlinear equations.

4.3.1 The model problem: The interacting Bose gas

In the nonlinear case, we must amend our original definition of the matrix $\mathbf{K}$. We defined it for the linear case in equation (4.5) as the integrals of the energy functional, equation (4.4). But in the nonlinear case we must instead define it so that equation (4.6) holds, since that is our basis for building the system of equations, $\mathbf{Ku} = \mu\mathbf{Mu}$.

Thus, we must instead use the corresponding Euler-Lagrange equation, exactly analogous to what we did in section 3.3.6; or we utilise the Galerkin formulation on the modified Gross-Pitaevskii equation (3.109). Either way, we get

$$K_{ij}(\psi) = \int_\Omega \mathrm{d}\mathbf{x}\, \frac{1}{2}\nabla N_i \nabla N_j + (V(\mathbf{x}) + C(1 + D|\psi|)|\psi|^2)N_i N_j, \tag{4.101}$$

with $V(\mathbf{x}) = (x^2 + y^2 + \lambda^2 z^2)/2$. There is no change for M_{ij}, which is linear.

4.3.2 Successive substitutions

The method of successive substitutions in the eigenvalue context is to repeatedly solve the linearised eigenvalue problem. After a solution is found, $\mathbf{K}$ is updated using the new solution and the process is repeated. The system to solve is

$$\mathbf{K}(\mathbf{u}^k)\mathbf{u}^{k+1} = \mu^{k+1}\mathbf{M}\mathbf{u}^{k+1}, \tag{4.102}$$

and it is solved exactly using a dedicated eigenvalue solver such as `arpack` (see bibliography).

The form of the linear system in each iteration is, dropping all indices to clarify which quantities are known and which are not,

$$\mathbf{Ku} = \mu\mathbf{Mu}. \tag{4.103}$$

Although the linear solution method in arpack, implicitly restarted Arnoldi iterations, is quite efficient, we end up doing a lot of work solving the linear eigenvalue problem exactly, without making much progress toward the nonlinear solution. Neither is this method stable for highly nonlinear problems in our numerical experiments.

4.3.3 Inverse power method

The inverse power method is applicable to nonlinear problems, and is quite similar to the method of successive substitutions. We repeatedly solve the linearised system

$$\mathbf{K}(\mathbf{u}^k)\mathbf{u}^{k+1} = \mu^k\mathbf{M}\mathbf{u}^k, \tag{4.104}$$

which is of the form (compare equation (4.103))

$$\mathbf{K}\mathbf{u} = \mathbf{b}, \tag{4.105}$$

since the whole right hand side is constant in each iteration. Notice that μ^k is just a scaling factor for $\mathbf{u}^{k+1}$, and can be dropped since we must scale the result regardless.

The inverse power method converges to the lowest eigenvalue in the linear case. To see why, we expand $\mathbf{u}^k$ in the eigenvector basis $\{\mathbf{u}_l\}$, with positive eigenvalues $\mu_l \leq \mu_{l+1}$,

$$\mathbf{u}^k = \sum_{l=1}^{N} c_l^k \mathbf{u}_l, \tag{4.106}$$

and look at one iteration of the method, ignoring the $\mathbf{u}$-dependence of $\mathbf{K}$:

$$\mathbf{u}^{k+1} = \mathbf{K}^{-1}\mathbf{M}\sum c_l^k \mathbf{u}_l = \sum c_l^k \mu_l^{-1}\mathbf{u}_l. \tag{4.107}$$

Each eigenvector component is divided by its eigenvalue. Thus, in each iteration the lowest eigenvalue is amplified by a factor $\frac{\mu_2-\mu_1}{\mu_1}$.

One way to improve convergence is to increase this factor. We can shift the eigenvalues without changing the eigenvectors by subtracting $a\mathbf{M}\mathbf{u}$ on the left hand side, changing equation (4.107) to

$$\mathbf{u}^{k+1} = \sum c_l^k (\mu_l - a)^{-1}\mathbf{u}_l, \tag{4.108}$$

thus making the amplification factor $\frac{\mu_2-\mu_1}{\mu_1-a}$ arbitrarily high as $a \to \mu_1$. Using the Rayleigh quotient $R(\mathbf{u}^k) = \mu^k$ as shift gives cubic convergence (Strang and Fix, 1973, p. 273), which we have confirmed in simple linear problems.

In nonlinear eigenvalue problems, the analysis above implies that in each iteration the method moves towards the lowest eigenvalue of the current linearised system. As long as the nonlinear eigenvalues are well separated, this works out well, and in our experience this is a quite reliable method. The shifted variant, however, does not converge reliably for nonlinear problems. We have not investigated whether this is because the shift makes the particular problems ill-conditioned, or if it is a weakness inherent to nonlinear problems.

4.3.4 Newton's method

We again write the problem as $\mathbf{F} = 0$ for $\mathbf{F}(\mathbf{u}) = (\mathbf{K} - \mu\mathbf{M})\mathbf{u}$. Then, for an initial guess $\mathbf{u}$ sufficiently close to the solution $\mathbf{u} + \delta\mathbf{u}$, the Taylor series of $\mathbf{F}$ is

$$\mathbf{F}(\mathbf{u} + \delta\mathbf{u}) = \mathbf{F}(\mathbf{u}) + \nabla\mathbf{F}(\mathbf{u})\delta\mathbf{u} + \mathcal{O}(\delta\mathbf{u}^2) = 0, \tag{4.109}$$

or, to first order,

$$\nabla\mathbf{F}(\mathbf{u})\delta\mathbf{u} = -\mathbf{F}(\mathbf{u}). \tag{4.110}$$

Here, $\nabla\mathbf{F} = \mathbf{J}$ is the Jacobian matrix. In general, if $\mathbf{F} = \mathbf{A}\mathbf{u} - \mathbf{b}$ ($\mathbf{b} = 0$ in our case), the Jacobian is

$$\mathbf{J} = \mathbf{A} + \delta\mathbf{A}, \tag{4.111}$$

for some $\delta\mathbf{A}$ that is nonzero when $\mathbf{A}$ depends on $\mathbf{u}$. The residual after k iterations is then

$$\mathbf{A}^k\mathbf{u}^k - \mathbf{b} = \mathcal{O}(\delta\mathbf{u}^2), \tag{4.112}$$

making the convergence quadratic.

Newton's method on eigenvalue problems

Calculating $\mathbf{J}$ is not really difficult, but we must be careful with the indices. One element of the Jacobian is

$$\begin{aligned} J_{ij} = \frac{\partial F_i}{\partial u_j} &= \sum_k \frac{\partial (K_{ik} - \mu M_{ik}) u_k}{\partial u_j} \\ &= K_{ij} - \mu M_{ij} + \underbrace{\sum_k u_k \frac{\partial K_{ik}}{\partial u_j}}_{\delta K_{ij}} - (\mathbf{M}\mathbf{u})_i \frac{\partial \mu}{\partial u_j}, \end{aligned} \tag{4.113}$$

where we use the notation $(\mathbf{M}\mathbf{u})_i$ for the i^{th} element of the vector $\mathbf{M}\mathbf{u}$.

Using the definition of $\mathbf{K}$ in equation (4.101):

$$\frac{\partial K_{ik}}{\partial u_j} = \int C\left(1 + \frac{3}{2}D|\psi|\right)\frac{\partial |\psi|^2}{\partial u_j} N_i N_k \,\mathrm{d}\mathbf{x} \tag{4.114}$$

Before proceeding any further, we make a key observation: $|\psi|^2$ *is not complex differentiable.* Thus, in order to use Newton's method on this problem, we are forced to use real coefficients. We choose to split ψ into real and imaginary components, and define

$$\psi = \psi_{\mathrm{R}} + i\psi_{\mathrm{I}} = \sum_{\text{real indices } j} u_j N_j + i \sum_{\text{imag indices } j} -i u_j N_j. \tag{4.115}$$

This choice is consistent with the standard definition of $\psi = \sum_i u_i N_i$, at the price of making some of the basis functions imaginary. We will change this when we implement the method.

Now the differentiation is well defined, and we can write

$$\frac{\partial |\psi|^2}{\partial u_j} = \frac{\partial(\psi_{\mathrm{R}}^2 + \psi_{\mathrm{I}}^2)}{\partial u_j} = 2\left(\psi_{\mathrm{R}}\frac{\partial \psi_{\mathrm{R}}}{\partial u_j} + \psi_{\mathrm{I}}\frac{\partial \psi_{\mathrm{I}}}{\partial u_j}\right) = 2\chi_j N_j, \tag{4.116}$$

where

$$\chi_j = \begin{cases} \psi_{\mathrm{R}} & \text{if } j \text{ is a real index,} \\ -i\psi_{\mathrm{I}} & \text{if } j \text{ is an imaginary index.} \end{cases} \tag{4.117}$$

Multiplication by $\sum_k u_k$ takes care of the stray index k in equation (4.114), resulting in

$$\delta K_{ij} = \int C(2 + 3D|\psi|)\psi N_i N_j \chi_j \,\mathrm{d}\mathbf{x}, \tag{4.118}$$

but since ψ is complex, this is of formal validity only. Sorting out the real and imaginary elements in $\delta\mathbf{K}$, we finally get

$$\delta K_{ij} = \int C(2 + 3D|\psi|)\chi_i \chi_j N_i N_j \,\mathrm{d}\mathbf{x}. \tag{4.119}$$

A possible source of confusion should be cleared up here. In the implementation, all of the basis functions are real, and as such there is an implicit multiplication by i on the right hand side of equation (4.119) when either of the indices are imaginary. The effect is that $-i$ disappears from equation (4.117), and all δK_{ij} are then real and positive.

Differentiating the eigenvalue is a bit more tedious, but in the end we get

$$\begin{aligned} \frac{\partial \mu}{\partial u_j} &= \frac{\partial(\mathbf{u}^{\mathrm{T}}\mathbf{K}\mathbf{u})}{\partial u_j} = \sum_{ik} \frac{\partial(u_i K_{ik} u_k)}{\partial u_j} \\ &= (\mathbf{K}\mathbf{u} + \mathbf{K}^{\mathrm{T}}\mathbf{u} + \delta\mathbf{K}^{\mathrm{T}}\mathbf{u})_j. \end{aligned} \tag{4.120}$$

Since $\mathbf{K}$ and $\delta\mathbf{K}$ are symmetric, equation (4.113) can be written as

$$J_{ij} = K_{ij} - \mu M_{ij} + \delta K_{ij} - [\mathbf{M}\mathbf{u}]_i[2\mathbf{K}\mathbf{u} + \delta\mathbf{K}\mathbf{u}]_j. \tag{4.121}$$

Unfortunately, the last term in equation (4.121) renders it useless for our purposes, since it is nonzero for nearly every index pair i, j. We must have a sparse Jacobian to be able to solve problems of any realistic size.

Implicit eigenvalue method

The simplest way to keep the Jacobian sparse is simply to ignore the $\mathbf{u}$-dependency of μ. The Jacobian is then approximated by

$$\mathbf{J} = \mathbf{K} - \mu\mathbf{M} + \delta\mathbf{K}. \tag{4.122}$$

We define $\Delta\mu^k = \mu^k - \mu^{k-1}$ as the change in eigenvalue. We use Δ here for the actual change, reserving δ for the calculated first order correction. The residual after k iterations is

$$\mathbf{F}^k = -\Delta\mu^k \mathbf{M}\mathbf{u}^k + \mathcal{O}(\delta\mathbf{u}^2), \tag{4.123}$$

which is linear in the corrections. Thus, when we ignore the change in μ, we no longer get quadratic convergence. But as long as μ is fairly stable — and we know that the Rayleigh quotient has a stationary point at the eigenstate, so it does not change much in the vicinity — we can still achieve good linear convergence.

Explicit eigenvalue method

If we use the approximation above, but add the normalisation requirement $\mathbf{u}^{\mathrm{T}}\mathbf{M}\mathbf{u} = 1$ to the linear system, we get

$$\hat{\mathbf{F}} = \begin{bmatrix} \mathbf{F} \\ \mathbf{u}^{\mathrm{T}}\mathbf{M}\mathbf{u} - 1 \end{bmatrix} = 0, \tag{4.124}$$

with associated Jacobian

$$\hat{\mathbf{J}} = \begin{bmatrix} \mathbf{J} & -\mathbf{M}\mathbf{u} \\ 2(\mathbf{M}\mathbf{u})^{\mathrm{T}} & 0 \end{bmatrix}. \tag{4.125}$$

We then set up and solve the linear system for the first order correction in $\mathbf{u}$ and μ,

$$\hat{\mathbf{J}} \begin{bmatrix} \delta\mathbf{u} \\ \delta\mu \end{bmatrix} = -\hat{\mathbf{F}}. \tag{4.126}$$

Solving this augmented system for the residual in iteration k, we see that we have recovered quadratic convergence as long as the corrections are sufficiently small,

$$\hat{\mathbf{F}}^k = \begin{bmatrix} -\delta\mu^k\mathbf{M}\delta\mathbf{u}^k + \mathcal{O}(\delta\mathbf{u}^2) \\ \mathcal{O}(\delta\mathbf{u}^2) \end{bmatrix}. \tag{4.127}$$

Notice also that the last row of equation (4.125) is only used for scaling the correction vector. If we scale $\mathbf{u}$ explicitly, the last element of $\hat{\mathbf{F}}$ is 0, and we can multiply the corresponding row in $\hat{\mathbf{J}}$ by an arbitrary number without changing the solution. Dividing this row by -2 makes the augmented Jacobian symmetric.

4.3.5 Imaginary time propagation

The time evolution of the Schrödinger equation[8] is given by

$$i\frac{\partial}{\partial t}\psi = H\psi, \tag{4.128}$$

---

[8] The results in this section are more general, but we expect the reader to be most familiar with them in this context.

with the formal solution

$$\psi(t) = e^{-iHt}\psi(0), \tag{4.129}$$

and thus one time step Δt changes an arbitrary state $\psi(t_0)$ by a factor $e^{-iH\Delta t}$. The exponential is defined by its power series,

$$e^{-iH\Delta t} = \sum_{k=0}^{\infty} \frac{(-iH\Delta t)^k}{k!} \tag{4.130}$$

If we let the wave function develop in imaginary time $\hat{t} = it$, with $\tau = \hat{t}/i$ real,

$$\psi(\tau_0 + \Delta\tau) = e^{-H\Delta\tau}\psi(\tau_0), \tag{4.131}$$

the lowest eigenvalue of H will soon dominate in $\psi(\tau)$, as the higher eigenvalues are suppressed by a constant factor each time step. This makes the convergence linear. Re-scaling is required to keep the norm.

We use the Taylor expansion $(1 + x)^{-1} = 1 - x + x^2 - \mathcal{O}(x^3)$ to find

$$(I + \theta\Delta\tau H)^{-1}(I - (1 - \theta)\Delta\tau H) = I - \Delta\tau H + \theta(\Delta\tau H)^2 - \mathcal{O}(\Delta\tau^3 H^3), \tag{4.132}$$

which is equal to the series in equation (4.130) to second order if $\theta = 1/2$, or first order otherwise. This is a reformulation of the well known theta rule, which we shall derive in a more formal way in section 4.4.

We can now substitute the discretisation $H \to \mathbf{M}^{-1}\mathbf{K}$, $\psi \to \mathbf{u}$, and equation (4.131) is then approximated by the linear system

$$(\mathbf{M} + \theta\Delta\tau\mathbf{K})\mathbf{u}^{k+1} = (\mathbf{M} - (1 - \theta)\Delta\tau\mathbf{K})\mathbf{u}^k \tag{4.133}$$

for one imaginary-time step from k. If $\mathbf{K}$ depends on $\mathbf{u}$, we will normally use the linearisation $\mathbf{K}(\mathbf{u}^k)$ on both sides of equation (4.133) so that each imaginary-time step is linear. It seems unlikely that solving equation (4.133) as a nonlinear equation improves convergence more than marginally.

This method converges to the lowest eigenvalue, but it has a stability requirement that can make convergence slow. The truncation of the series of equation (4.130) requires $\Delta\tau H \ll 1$ to be accurate, and $\Delta\tau H < 1$ to be stable. We only care about stability, as the accuracy will improve as the higher eigenvalues of H are suppressed. Thus, if the highest eigenvalue of H is μ_N, then the stability requirement is $\Delta\tau\mu_N < 1$.

We can try to estimate the highest eigenvalue for the discretised Gross-Pitaevskii equation. A grid of N elements in one space dimension can hold linear harmonic oscillator eigenvalues up to about $\mu \sim N$. In d dimensions, they combine to $\mu_N \sim Nd$. We can then use the upper bound for the nonlinear eigenvalues, as we did in one dimension in equation (4.67), to find a rough estimate of the stability bound for $\Delta\tau$,

$$\Delta\tau \lesssim \frac{1}{NdC(1+D)n_0}, \tag{4.134}$$

where n_0 is the central density of the ground state. The density can be computed analytically in the noninteracting and in the Thomas-Fermi regime,[9] but that seems redundant in such a rough estimate. It is always less than about 0.75, and proportional to $C^{-1/3}$ in the strongly interacting limit. Hence, reducing Cn_0 to $C^{2/3}$ is a useful rule of thumb even if its validity is uncertain in the mixed regime.

This stability requirement does not apply to evolution in real time, where for example the Crank-Nicholson form ($\theta = 1/2$) is unconditionally stable; there we are more concerned about the accuracy.

4.4 Time evolution

Our interest in the time evolution stems from the range of dynamical phenomena of Bose-Einstein condensates that have become available experimentally in the last ten years. Numerical time development of the Gross-Pitaevskii equation will allow us to reproduce some of these phenomena.

Nevertheless, time evolution is not a central part of this thesis and we will only look at a few of the available time integration methods. A serious study of this field should include at least symplectic integrators and operator-splitting methods; we only look at the simpler finite-difference time integrators.

4.4.1 Finite difference methods

A common approach is to use finite difference stepping in time. The common finite difference methods are based on the Taylor series,

$$f(t+\tau) = f(t) + \tau f'(t) + \frac{\tau^2}{2} f''(t) + \mathcal{O}(\tau^3). \tag{4.135}$$

Setting $\tau = \Delta t$, we get the forward time step, for a first order accurate expression for f',

$$f'(t) = \frac{f(t+\Delta t) - f(t)}{\Delta t} - \frac{\Delta t}{2} f''(t) + \mathcal{O}(\Delta t^2), \tag{4.136}$$

---

[9] We ignore that the Thomas-Fermi approximation is valid only for the regular, not the modified, Gross-Pitaevskii equation. The third order term (if positive) serves to increase repulsion, thus further lowering the central density.

and $\tau = -\Delta t$ is the backward time step,

$$f'(t) = \frac{f(t) - f(t - \Delta t)}{\Delta t} + \frac{\Delta t}{2} f''(t) + \mathcal{O}(\Delta t^2). \tag{4.137}$$

Finally, by adding equations (4.136) and (4.137), we get the central difference, accurate to second order in Δt.

$$f'(t) = \frac{f(t + \Delta t) - f(t - \Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2). \tag{4.138}$$

How does this apply to the present problem? The time evolution of the Schrödinger equation is given by the difference equation

$$i\frac{\partial}{\partial t}\psi = H\psi, \tag{4.139}$$

of which the Gross-Pitaevskii equation is an example, given in equation (3.111). We can use one of the above approximations to discretise the left-hand side; for the right-hand side, we use the finite element space discretisation, as described elsewhere.

Forward and backward time stepping

Applying equation (4.136) to equation (4.139), we get the forward difference formula, also known as forward Euler,

$$\psi^{k+1} = (1 - i\Delta t H^k)\psi^k, \tag{4.140}$$

where we have put a time index on H to clarify the time step at which it is discretised, if it varies in time. The backward version, or backward Euler, reads instead

$$(1 + i\Delta t H^k)\psi^k = \psi^{k-1}. \tag{4.141}$$

The main difference between the two is that the forward version, as written here, is explicit, while the backward version is implicit. This can make a large difference both for complexity of implementation and for performance. It is not true, however, when H is derived by finite elements, as it is here. Then, H in matrix form is $\mathbf{M}^{-1}\mathbf{K}$, so the seemingly explicit equation (4.140) instead becomes the implicit

$$\mathbf{M}\mathbf{u}^{k+1} = (\mathbf{M} - i\Delta t \mathbf{K}^k)\mathbf{u}^k. \tag{4.142}$$

Thus, both versions are in fact implicit in this case.

The Crank-Nicholson method

We get the Crank-Nicholson method by applying the central difference formula, equation (4.138), at the mid-point of two time steps,

$$\psi^k - \psi^{k-1} = -i\Delta t H^{k-1/2}\psi^{k-1/2}, \tag{4.143}$$

which is known as the midpoint rule. We can approximate the right-hand side by its average,

$$H\psi^{k-1/2} \approx \frac{H^k\psi^k + H^{k-1}\psi^{k-1}}{2}. \tag{4.144}$$

Combining these two equations, we get the Crank-Nicholson formula, also known as the trapezoidal rule,

$$(1 + i\frac{\Delta t}{2}H^k)\psi^k = (1 - i\frac{\Delta t}{2}H^{k-1})\psi^{k-1}. \tag{4.145}$$

We observe that equation (4.145) is just the arithmetic average of the forward and backward Euler methods from the previous section. This allows us to write all three on a common form,

$$(1 + i\theta\Delta t H^k)\psi^k = (1 - i(1-\theta)\Delta t H^{k-1})\psi^{k-1}, \tag{4.146}$$

where $\theta = 0, 1$ recovers the forward and backward methods respectively, while $\theta = 1/2$ recovers the Crank-Nicholson method. This form is known as the theta rule. It is derived in another way in section 4.3.5.

Stability of the Crank-Nicholson method

Stability of a numerical time integration scheme is concerned with whether any component of the solution is amplified, since a constant amplification in one time step turns into an exponential amplification over time. Thus, a scheme that exhibits amplification cannot be stable, since a small perturbation in an amplified component later will dominates the system.

In addition to this classical stability requirement, we would also like that the norm of the system is unchanged — in the present case this corresponds to conservation of the particle number — and that the energy is unchanged. These considerations stem from the physical properties of the system, and some time integration methods that fulfill the classical stability requirement do not fulfill the norm and energy preservation properties.

We consider the time operator, derived in the previous sections, that brings the system forward one time step, $\psi^{k+1} = U^k\psi^k$. It can be written in the combined form of equation (4.146) as

$$U^k = (1 + i\theta\Delta t H^{k+1})^{-1}(1 - i(1-\theta)\Delta t H^k). \tag{4.147}$$

If H is independent of time and has eigenvalues μ_n, then U has the same eigenvectors as H, with eigenvalues

$$\lambda_n = \frac{1 - i(1-\theta)\Delta t \mu_n}{1 + i\theta\Delta t \mu_n}, \tag{4.148}$$

and their magnitudes are given by

$$|\lambda_n|^2 = 1 + \alpha_n(1 - 2\theta), \tag{4.149}$$

for a positive number $\alpha_n = 1/((\Delta t\mu_n)^{-2} + \theta^2)$. Clearly, for $\theta < 1/2$ all eigenvalues are amplified, thus the method is unstable in the classical sense, and for $\theta \geq 1/2$ the method is unconditionally stable. But norm preservation is only achieved when $\theta = 1/2$.

We proceed to look at the case of varying H, following the analysis in (Kvaal, 2004).[10] To simplify the equations, we set $\theta = 1/2$ unconditionally,[11] and write quantities at time step $k+1$ as $\hat{X}$, quantities at time step k without such adornment. We define $\tau = \Delta t/2$. The time evolution operator is then

$$U = (\underbrace{1 + i\tau\hat{H}}_{A})^{-1}(\underbrace{1 - i\tau H}_{B}). \tag{4.150}$$

Conservation of the norm. The norm at time step $k+1$ is

$$\|\hat{\psi}\|^2 = \|U\psi\|^2 = (\psi, U^\dagger U\psi), \tag{4.151}$$

and if U is unitary, then it must be equal to the norm at step k, $\|\psi\|^2 = (\psi, \psi)$. Hence we look specifically at how much $U^\dagger U$ departs from the identity.

The transformation can be written as $U^\dagger U = B^\dagger (AA^\dagger)^{-1} B$, and we approximate the inverse through the power series expansion, $(1 - X)^{-1} = \sum_{j=0}^{\infty} X^j$, as

$$(AA^\dagger)^{-1} = (1 + \tau^2\hat{H}^2)^{-1} = 1 + \tau^2\hat{H}^2 + \mathcal{O}(\tau^4), \tag{4.152}$$

where the truncation is valid if $|\tau\hat{\mu}_n| < 1$ for all n. In section 4.3.5, we estimated the highest eigenvalue of the discretised Gross-Pitaevskii equation as $\mu_{\max} \approx Nd + C^{2/3}(1 + D)$, but in general it is $\mathcal{O}(h^{-1})$ for second-order elliptic partial differential equations.

A straight calculation of $U^\dagger U$, truncating all terms of order τ^4 or greater, yields

$$B^\dagger(1 - \tau^2\hat{H}^2 + \mathcal{O}(\tau^4))B = 1 + \tau^2[H, \hat{H}] - i\tau^3[H, \hat{H}^2] + \mathcal{O}(\tau^4), \tag{4.153}$$

---

[10] Non-constant H implies either a nonlinear equation (implicit time dependence), or parameters that vary in time (explicit time dependence).

[11] There is no reason from the preceding discussion for assuming that this choice is optimal also when $\mathrm{d}H/\mathrm{d}t \neq 0$, but it turns out that it is.

where $[\cdot,\cdot]$ are commutation brackets. If H varies sufficiently slowly, we can use the Taylor series of $\hat{H}$, truncated after the linear term, to write

$$[H,\hat{H}] = -2\tau[H,\dot{H}] + \mathcal{O}(\tau^2). \tag{4.154}$$

Thus equation (4.153) becomes in this case

$$U^\dagger U = 1 - 2\tau^3[H,\dot{H}] + \mathcal{O}(\tau^4), \tag{4.155}$$

and the norm is bounded by

$$\|\psi^{k+1}\|^2 = (1 + \mathcal{O}(\Delta t^3))\,\|\psi^k\|^2. \tag{4.156}$$

A norm-preserving variant. If the Crank-Nicholson step is changed to use H^k for both the forward and the backward parts of the step, such that

$$U = (1 + i\tau H^k)^{-1}(1 - i\tau H^k). \tag{4.157}$$

Then $A = B^\dagger$, all factors of $U^\dagger U$ commute, and they cancel exactly. Thus the time evolution becomes exactly unitary. This method is also computationally simpler, since only known quantities are involved at a given time step, making the resulting system of equations linear.

However, it is apparently well known that this version of the Crank-Nicholson method is susceptible to kinetic energy growth over time in nonlinear problems (see for example Dalfovo and Modugno, 2000, p. 3).

Long-term conservation of energy and norm

The midpoint rule, equation (4.143), is known to be a Gauss collocation method, and thus symplectic. These methods have very good long time stability properties; for example, all second order invariants, i.e., norm and angular momentum, are exactly preserved, and energy is preserved to $\mathcal{O}(\Delta t^2)$ over exponentially long time (Hairer et al., 2000).

The Crank-Nicholson method is conjugated to the midpoint rule. This means that the distance between the solution of the midpoint rule and the solution of the Crank-Nicholson method is $\mathcal{O}(\Delta t^2)$ at all times (ibid). We therefore expect the Crank-Nicholson solution to have norm, angular momentum and energy conserved to $\mathcal{O}(\Delta t^2)$ for the duration of our simulations. Our numerical experiments support this.

Thus, at the price of solving a nonlinear system of equations at each time step, we get a time propagator which is stable over long time. In practice we will use a time step for which H^{k+1} is well predicted by a linear approximation, so the additional work required to solve the nonlinear equations is not prohibitive. In experiments we have found that two or three nonlinear iterations are usually sufficient.

4.5 Excited states

Using the methods in section 4.3, we can usually find the non-rotating ground state without problems. However, we are also interested in finding stationary states with vortices. Some methods that allows us to find these are given here.

4.5.1 Choosing the initial state

If we start the search for the stationary state sufficiently close to the desired solution, we can use Newton's method to find the exact solution. Newton's method is agnostic as to which state it converges to; it finds the closest one. While we often cannot specify the initial state close enough to the solution for the unmodified Newton's method to converge, there are two things we can do.

One is to use Newton's method with a variable step length. This method is called *steepest descent*. It is usually implemented to fit the residual along the search direction to a polynomial, and going to the minimum of the polynomial. This avoids divergence, but because of the shorter step length it takes more steps to reach the solution.

The other is to use another nonlinear solver, such as imaginary time evolution, for a few iterations before switching to Newton's method. A few iterations of another solver allows much of the error to be reduced. The other solver will in practice eventually converge to the ground state, but if the initial state is nearly orthogonal to the ground state it takes a long time before this happens.

Either way, this method is most useful for finding the state with a single central vortex, by starting for example with the linear harmonic oscillator solution of the same. We can achieve the same in a more flexible way by modifying the wave function after each iteration, as described in the following section.

4.5.2 Externally applied phase

Any single vortex state may be created by applying rotation to the result,

$$\psi'(r,\theta,z) = e^{i\theta m}|\psi(r,\theta,z)|, \quad m = 0, 1, 2, \ldots, \tag{4.158}$$

thus forcing the state to be orthogonal to any spherically symmetric state with rotation different from m. For the method of imaginary time development, a state that is orthogonal to the current state will stay suppressed. Because of numerical noise, however, we repeat the application of equation (4.158) after each iteration to retain orthogonality. Thus we find the state which has lowest energy, subject to the given rotation.

This method also works for the inverse power method, but in our experience it is more fragile and does not converge in all cases.

Any rotation can be applied, as long as it is a multiple of 2π, and the center of rotation can be shifted away from the center of the trap. But only one vortex can be formed in this fashion. To generate a stationary state containing multiple vortices, we must use another technique, that of finding stationary states in a rotating trap.

4.5.3 Rotating lab frame

Let us look at how the state transforms when we look at it from a rotating frame. The generator of rotation is, in natural units,

$$L_z = -i\left(x\frac{\partial}{\partial y} - y\frac{\partial}{\partial x}\right) = -i\frac{\partial}{\partial \theta}. \tag{4.159}$$

The wave function ψ' in the frame rotating with angular frequency Ω is

$$\psi' = e^{i\Omega L_z t}\psi, \tag{4.160}$$

and the time-dependent Schrödinger equation, $H\psi = i\frac{\partial}{\partial t}\psi$, becomes for the transformed state

$$(H - \Omega L_z)e^{-i\Omega L_z t}\psi' = ie^{-i\Omega L_z t}\frac{\partial}{\partial t}\psi', \tag{4.161}$$

so that the state satisfies

$$H'\psi' = i\frac{\partial}{\partial t}\psi', \tag{4.162}$$

with $H' = H - \Omega L_z$ since H commutes with L_z.

Since L_z vanishes for states that are symmetric around the z axis, $\psi = \psi'$ for such states.

If we have a stationary state in the rotating frame, and want to see its time development in the stationary frame,

$$\psi(t) = e^{-i\Omega L_z t}e^{-iH't}\psi'(0) = e^{-iHt}\psi'(0). \tag{4.163}$$

Thus, we get the time development in the stationary frame by developing the transformed initial state in time using the stationary Hamiltonian.

Why do we find stationary states with multiple vortices in the rotating frame that we cannot find in a stationary frame? Our numerical methods — and experimental methods as well — most easily find the lowest energy eigenstate. In the rotating frame, the eigenvalues are shifted relative to each other. But without interaction, the *order* of the eigenstates is not changed until $\Omega = 1$, when the eigenvalue $\mu = 0$ is infinitely degenerate and the trap potential is effectively cancelled. In the interacting case however, the order of the eigenstates may change even at $\Omega < 1$. Then we can create states with multiple vortices. These states are interesting since they are regularly seen in experiments.

Chapter 5

Results

5.1 Verification

We start by verifying our model by replicating known result. First we look at the linear problem, which has a known analytical solution. We check that as the number of grid points increases, we get results that are successively closer to the analytical solution. The convergence should follow from the results of the analysis of the finite element method in section 4.1.8. We then reproduce the results by Nilsen et al. (2005). These results are for the full 3D nonlinear problem.

5.1.1 The linear problem

The linear problem takes the form of a harmonic oscillator. In one dimension, it is

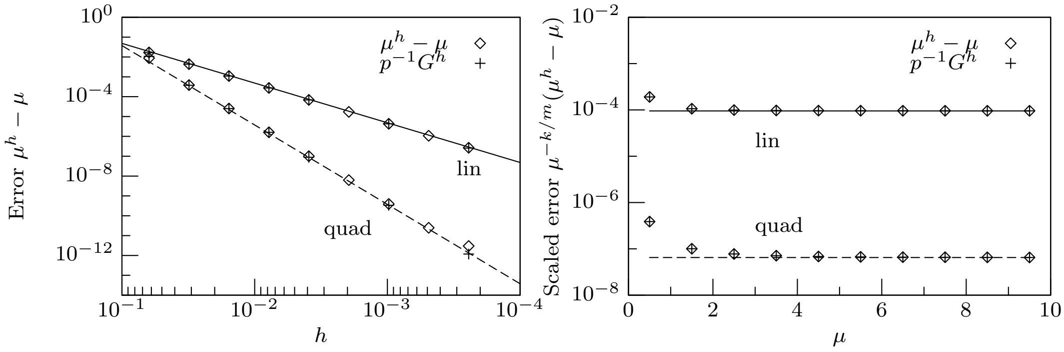
$$-\frac{\mathrm{d}^2\Psi(x)}{\mathrm{d}x^2} + \frac{x^2}{2}\Psi(x) = \mu\Psi(x), \tag{5.1}$$

which has known solutions

$$\Psi_l(x) = c_l H_l(x) e^{-x^2/2}, \quad \mu_l = l + \frac{1}{2}. \tag{5.2}$$

For brevity, we look at the convergence in eigenvalues only. If these converge as they should, then we expect the eigenvectors to do the same. In the absence of interactions, the eigenvalue — which represent the chemical potential — and the energy per particle are equal.

Figure 5.1(a) shows the error in the ground state eigenvalue as the number of elements increases. It behaves exactly as expected from the results in section 4.1.8. Furthermore, it seems that the virial theorem, derived in section C.1, is an excellent predictor of the error of stationary states.



(a) Convergence to the exact ground state eigenvalue as a function of element size h. The error goes as $h^{2(k-m)}$: h^2 for linear and h^4 for quadratic elements.

(b) Convergence to higher eigenvalues for a fixed element size h. The error goes as $\mu^{k/m}$. The plot shows $\mu^{-k/m}(\mu^h - \mu)$, for linear ($k/m = 2$) and quadratic ($k/m = 3$) elements.

Figure 5.1: The error in the eigenvalues and the virial theorem for the 1D harmonic oscillator. We see that the predictions of theory from section 4.1.8 are fulfilled, and that the virial error is close to the eigenvalue error.

We let G denote the departure from the virial theorem, constructed by inserting the different energy terms of the energy functional, equation (3.107), into equation (C.11),

$$G[\Psi] = \int -\frac{1}{2}|\nabla\Psi|^2 + \frac{V}{2}|\Psi|^2 - \frac{nC}{4}|\Psi|^4 - \frac{3nCD}{10}|\Psi|^5 \,\mathrm{d}\mathbf{x} \tag{5.3}$$

in n dimensions. We have found in all cases that we have checked that

$$E[u^h] - E[u] \approx p^{-1}G[u^h]; \tag{5.4}$$

that is, the error in energy is approximated by the error in the virial theorem, scaled by a factor $p = 1$ for linear and $p = 2$ for quadratic elements.[1] Equation (5.4) is a useful result, since the left hand side is usually unknown, and we can use it to estimate the numerical error.

The higher eigenvalues have higher error. In figure 5.1(b), the error is shown for increasing eigenvalues of a fixed number of elements. The error in the eigenvalues is increasing as predicted by equation (4.56), and the error in the virial follows it closely.

The nonlinear problem

We must verify that the results of the full nonlinear Gross-Pitaevskii equation and the modified version, the MGP, are consistent with earlier results for the

---

[1] It is a reasonable guess that p is in fact the $k - m$ of section 4.1.8, but we present no further evidence.

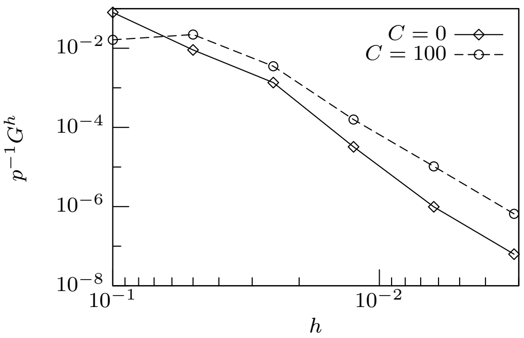


Figure 5.2: Convergence of the nonlinear problem for different element sizes. The virial error is plotted for the linear (solid line) and nonlinear problem (dashed line), using quadratic elements in 1D. The linear values are similar to those shown in figure 5.1, and we see here that the nonlinear problem shows similar convergence at decreasing h as the linear problem.

same equation. We have calculated results for the ground state and for the one-vortex state, both of which can be compared to those available in (Nilsen et al., 2005). In the following, data marked "(reference)" are from this source. The simulations were done on a regular grid of 130x130x70 nodes, using quadratic elements.

We would also like to estimate the uncertainty in our calculations independently of this source, to verify that we can deduce the accuracy of calculations in other cases where exact results are not known. To do this, we used a method of calculating the results on different grids and extrapolating the error from these. The details are found in section B.1, where we deduce the uncertainty in the MGP ground state.

For the other states — the GP ground state, the MGP single-vortex state, and the GP single-vortex state — we either use the same number of significant digits (for the ground state) or one less (for the vortex states). The reason for using fewer digits for the vortex state is that the virial error G in that case is some 20 times higher, and we assume that this increased error leads to a loss of one significant digit in each energy component. Finally, we assume that the sign of the error for these other states is the same as in the MGP ground state calculations.

These are a lot of assumptions, and with some more effort we could have done the same error fits as we did for the ground state; as it stands, the uncertainties for these other states must be viewed as rough estimates. The results are compared with those in the reference (Nilsen et al., 2005) in tables 5.1 and 5.2. The numbers are in reasonable agreement, although a few discrepancies in the last significant digits remain.[2] Still, we consider the numbers to be close enough to

---

[2] Another possible source of differences is that the nonlinear parameters C and D were converted from the reference parameters $a/d_\perp$ and N with only five digits accuracy: $a/d_\perp =$

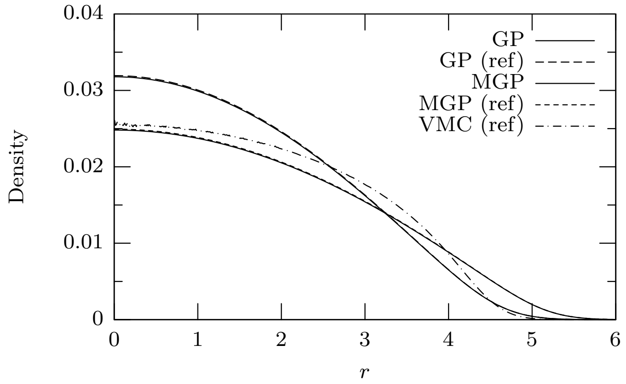


Figure 5.3: Ground state of the condensate in the large gas parameter regime, compared with earlier results. Both the Gross-Pitaevskii and the modified Gross-Pitaevskii curves are in close agreement with the reference. The variational Monte Carlo curve is not so close, but the central gas parameter $na^3 \sim 5 \cdot 10^{-2}$ is at the limit of the valid regime for the mean-field approximation. 3D calculations using $C = 952.22, D = 7.9391, \lambda = 1$.

Table 5.1: Ground state energies compared with reference finite difference calculations and a variational Monte Carlo simulation. All energies are per particle, in units of $\hbar\omega_\perp$. 3D calculations using $C = 952.22, D = 7.9391, \lambda = 1$. The superscripts indicate the sign of the error, ±1, in the last digit.

	μ	E	E_K	E_HO	E_2	E_3
GP	12.9797^-	9.4969^-	0.39495^+	5.6192^-	3.48279^+	
GP (ref.)	12.980	9.49684	0.39495	5.61911	3.482777	
MGP	15.4527^-	11.0611^-	0.35352^+	6.9410^-	2.51670^+	1.24994
MGP (ref.)	15.453	11.06108	0.35353	6.94092	2.516691	1.249938
VMC (ref.)		11.121	4.215	6.906		

validate the correctness of our numerical implementation.

The radial profiles in figures 5.3 and 5.4 agree with the reference profiles.

We have included in these comparisons the result of a variational Monte Carlo simulation, also from (Nilsen et al., 2005). It serves as a reminder of the limits of the mean-field approximation, no matter how exact our results are.

5.2 Physical applications

Having verified the accuracy of our implementation, we apply it to some interesting physical situations. We calculate the ground state vortex lattice of a

---

$0.15155, N = 500 \rightarrow C = 952.22, D = 7.9391$.

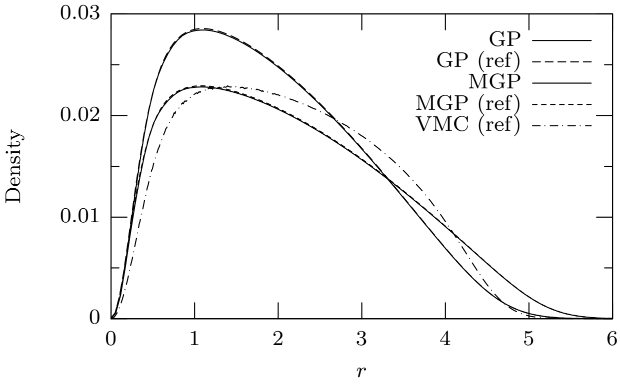
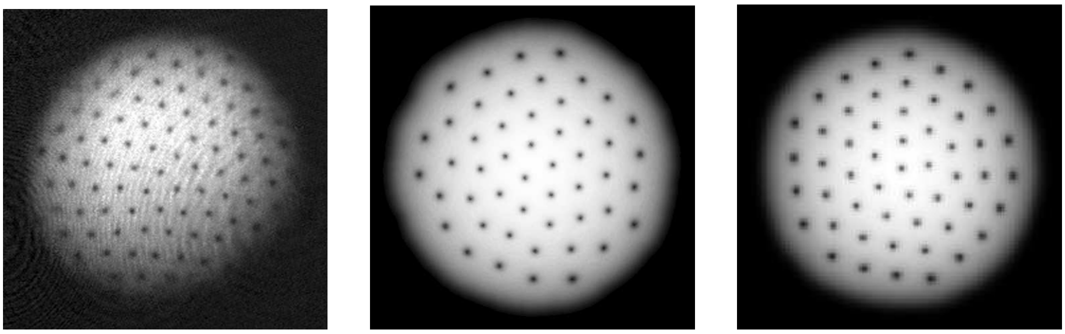


Figure 5.4: Single vortex state in the large gas parameter regime, compared with reference results. As in figure 5.3, our results are close to the reference calculations, not so close to the Monte Carlo calculations. 3D calculations using $C = 952.22, D = 7.9391, \lambda = 1$.

Table 5.2: Single vortex energies compared with reference finite difference calculations and a variational Monte Carlo simulation. All energies are per particle, in units of $\hbar\omega_\perp$. 3D calculations using $C = 952.22, D = 7.9391, \lambda = 1$. The superscripts indicate the sign of the error, ±1, in the last digit.

	μ	E	E_{K}	E_{HO}	E_2	E_3
GP	13.188^-	9.784^-	0.6366^+	5.743^-	3.4040^+	
GP (ref.)	13.187	9.7836	0.63701	5.7427	3.40387	
MGP	15.623^-	11.306^-	0.5614^+	7.039^-	2.4825^+	1.2234^-
MGP (ref.)	15.623	11.305	0.56184	7.0377	2.48242	1.22328
VMC (ref.)		11.334	4.352	6.982		



(a) MIT experimental results (Ketterle, 2001)

(b) Simulation on 2D irregular grid

(c) Simulation on 2D lattice grid

Figure 5.5: Density profile of vortex lattices. The simulations used the parameters $C = 5000, \Omega = 0.75, \lambda = 1$. Figure (a) shows experimental evidence of vortex lattice formation. The simulation in (b) shows a triangular lattice structure that is qualitatively similar to the experimental results in (a). The simulation on a regular lattice grid (c) shows a spurious angular "shower drain" symmetry in the vortex lattice; we have seen instances of grid-related defects also in other simulations under strong rotation, but do not fully understand their cause.

rotating trap in two dimensions and compare it with experimental results. As an application of the time development method in section 4.4.1, we calculate the decay dynamics of a double quantised vortex in two and three dimensions.

5.2.1 A stationary vortex lattice in 2D

The way we achieve a state with multiple vortices in simulations is basically the same as the experimental method: We spin the trap. In section 3.4.3 we saw that the energetically preferred way of absorbing angular momentum is to form several single-quantised vortices, and in section 4.5.3 we mentioned how a higher-lying state in the stationary frame becomes the ground state in a rotating frame.

Figure 5.5(a) shows an almost regular Abrikosov lattice of vortices, similar to those seen for flux lines in type II superconductors. Perfectly regular lattices have also been observed (Ketterle, 2001). In (b), a 2D simulation shows many of the same properties.[3] The most obvious difference is perhaps the lack of vortices at the edge of the condensate in the simulations. We do not know why these do not appear. The less regular lattice, and possibly the lack of edge vortices, may be caused in part by not finding the exact ground state. The imaginary time and inverse power nonlinear solvers seem to get stuck in local minima for a long

---

[3] We tuned the parameters to get a similar vortex count, but it is obviously not the same.

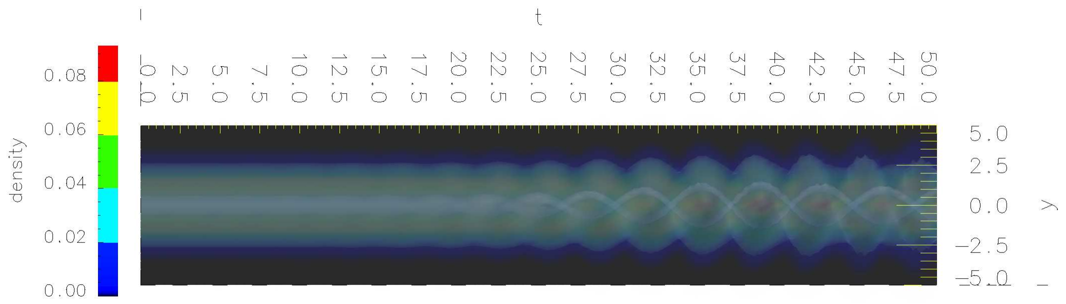


Figure 5.6: Decay of double-quantised vortex in 2D, visualised using density mapping. The time development is mapped to the third space dimension, and the resulting volume is volume-rendered and coloured according to density. A monochromatic isosurface is added to accentuate the vortex and the surface. The simulation parameters are $C = 20, \lambda = 1$ on a 60×60 space grid with linear square elements. The time step is $\Delta t = 0.05$. An animation of this sequence is found at the project home page (Haga, 2006).

---

time, and the Newton solver is not guaranteed to find the minimum. So while these simulations are of a stationary state, it is not necessarily the ground state of the rotating frame.

We should also mention that under strong rotation, when $\Omega \gtrsim 0.8$, we have seen results that are highly dependent on the element geometry for regular grids; in extreme cases, distinctly square-shaped condensates using quadratic elements and diagonally squeezed condensates using triangular elements. For this reason we do not have full confidence in the results for rotating frames. In figure 5.5(c), we have speculated that the central bend in the lattice may be caused by a similar phenomenon since it is not seen on the irregular grid (b).

5.2.2 Splitting of a double-quantised vortex in 2D

We saw in section 3.4.3, and in the previous section, that vortices with more than one quantum of rotation are not energetically favourable. Furthermore a condensate that is prepared in the double-quantised state may be dynamically unstable, depending on the nonlinear parameter (Pu et al., 1999; Möttönen et al., 2003). This has been experimentally verified, and it is clearly seen that the vortex core pulls apart to create two separate vortices (Shin et al., 2004).

Figure 5.6 shows the simulated time development of a double-quantised vortex, starting in a stationary state at $t = 0$ with a small perturbation (the ground state is mixed in at amplitude 0.01). Around $t = 20$, we see the vortex separating into two smaller vortices that circle around each other, and the excess energy manifests itself primarily as surface waves. The same sequence can be seen as

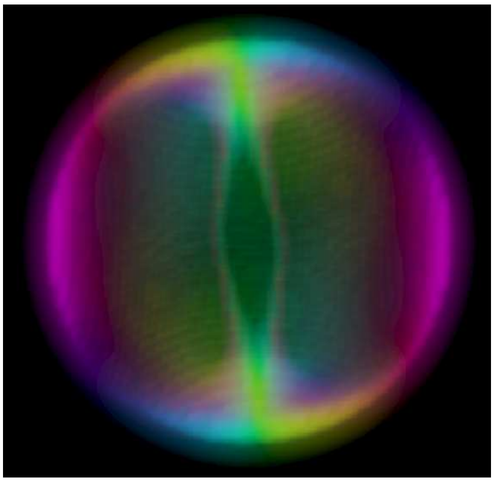


Figure 5.7: A decayed double-quantised vortex in 3D, visualised using inverse density mapping. Some time after the initial small perturbation is applied, the vortex splits in two along the z axis inside the condensate. The colour shows the phase of the wave function, and the intensity shows the inverse density. The volume is clipped to the Thomas-Fermi radius. The parameters $C = 500, \lambda = 1$ are used, and the state is allowed to evolve on a $50 \times 50 \times 50$ grid with linear cubic elements until $t \sim 30$ with $\Delta t = 0.05$. The parameters were chosen from the dynamically unstable regions in (Nilsen and Lundh, 2006).

---

an animation at the project home page (Haga, 2006).

5.2.3 Splitting of a double-quantised vortex in 3D

Some of the interesting dynamics of vortex splitting lie in the excitations in z direction that become more apparent in elongated traps. To investigate these numerically, we must do full 3D simulations. Figure 5.7 is an example of such a simulation, where the initially double-quantised vortex with a small perturbation has split into two single-quantised vortices.

Due in part to the time-consuming nature of the visualisation process, we have not been able to prepare an animation of vortex splitting in 3D in time for this thesis.

5.3 Numerical and algorithmic results

We present some results on the efficiency of the various numerical methods that we introduced in chapter 4, and our experience with them for different problems.

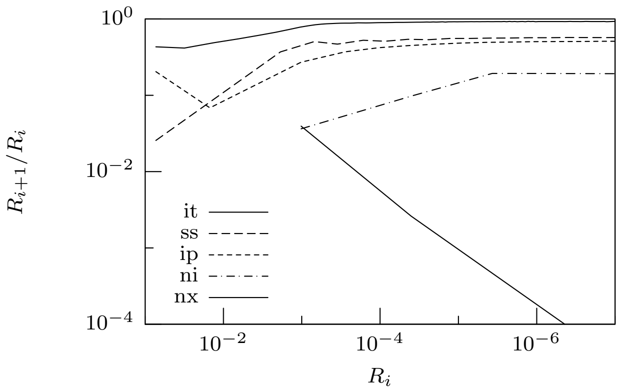


Figure 5.8: Convergence of the nonlinear solvers on a weakly nonlinear problem. The ground state is sought, starting from a randomised state. The graph shows the amount of error reduction in one nonlinear iteration as a function of the initial error. The methods are imaginary time (solid, top), successive substitutions using arpack (dashed), inverse power (stippled), implicit eigenvalue Newton (dash-dot), and explicit eigenvalue Newton (solid, bottom). The only method that shows higher than constant factor error reduction is the explicit eigenvalue Newton's method from section 4.3.4. When the error is above 10^{-3}, the Newton methods do not converge to the ground state. The nonlinear parameters are $C = 10$, $D = 0$.

5.3.1 Convergence of the nonlinear solvers

In this section, we discuss specifically the nonlinear solvers for use in the eigenvalue problem; that is, for finding a stationary state. We shall discuss the solvers to use in time development in section 5.3.3.

Of the nonlinear methods listed in section 4.3, most have their uses in different problems. Specifically, they can be placed on different positions on the stability—efficiency axis, where the method of imaginary time propagation is very stable but reduces the error just a little each iteration, the Newton methods need to be quite close to the solution but converge fast, and the inverse power method falls in between. The only exception is the method of successive substitutions, which is neither fast nor stable.

Figures 5.8 and 5.9 show the convergence of two sample problems in 1D, one of the GP equation with $C = 10$ and one of the MGP equation with $C = 100, D = 10$. The results are typical of what we have seen in other experiments. In summary:

Successive substitutions. The method of successive substitutions, which uses arpack as a linear eigenvalue solver, is not useful for the present problem. If it converges at all, it does so very slowly — each iteration is an order of magnitude slower than an iteration of the regular solvers, because the linear eigenvalue problem is solved exactly.

Inverse power method. The inverse power method in most cases converges to

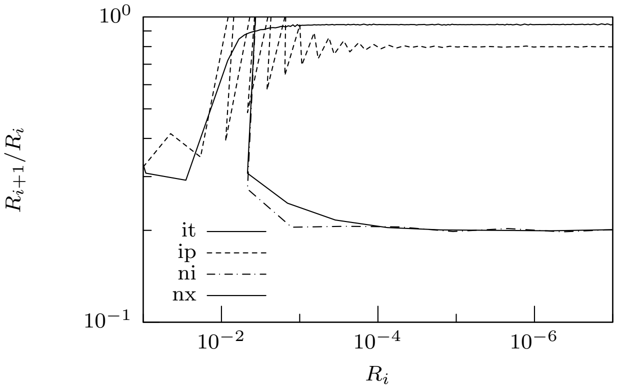


Figure 5.9: Convergence of the nonlinear solvers on a strongly nonlinear problem. The ground state is sought, starting from a randomised state. The graph shows the amount of error reduction in one nonlinear iteration as a function of the initial error. The methods are imaginary time (solid, top), inverse power (stippled), implicit eigenvalue Newton (dash-dot), and explicit eigenvalue Newton (solid, bottom). The method of successive substitutions failed to converge for this problem. All remaining methods exhibit a constant factor error reduction as $R_i \to 0$. When the error is above $5 \cdot 10^{-3}$, the Newton methods do not converge to the ground state. The nonlinear parameters are $C = 100$, $D = 10$.

the ground state from any starting point. But in highly nonlinear cases, the step length must be reduced to avoid divergence, and it can also fail to converge when conditions are applied to the wave function between iterations, such as the externally applied phase in section 4.5.2. When this method converges, it usually does so faster then the imaginary time method, below.

The Newton methods. As opposed to the other methods, the Newton methods have no inherent preference for which eigenstate they should converge to. Thus, they need to be brought close to the desired solution before they are used, and they can then be used to quickly refine the solution. The two Newton methods have this in common, and figures 5.8–5.9 show that they both need to be brought to about the same distance from the solution before they converge. Any of the other methods can be used to find the approximate solution.

Despite the explicit eigenvalue Newton's method converging quadratically whereas the implicit eigenvalue method is only linear, we generally prefer the implicit eigenvalue method. The reason is that the matrix manipulations that are needed for the explicit eigenvalue method connects all nodes in the matrix-vector product, thus increasing the communication overhead in a parallel computation. The error reduction in the implicit method, although constant, is good enough that only a few extra iterations are needed in practice.

Imaginary time propagation. The imaginary time method is the most stable of the methods we have considered, if the time step is not too large. In our experience, it converges from any state to the ground state, as long as the initial state is not orthogonal to the ground state. Because of numerical noise, this does not happen in practice, but a nearly orthogonal state will take a long time to appear. This can be an advantage, in that it reliably finds the ground state under externally applied conditions, e.g., externally applied phase to find the ground state with a given rotation. The inverse power method is less reliable in this sense.

Finding the right time step for the imaginary time method is quite easy in practice. We can start with the estimate in equation (4.134), but a few short test runs are a good idea: Starting with a randomised state, any step size which reduces the residual in the first few iterations is good.

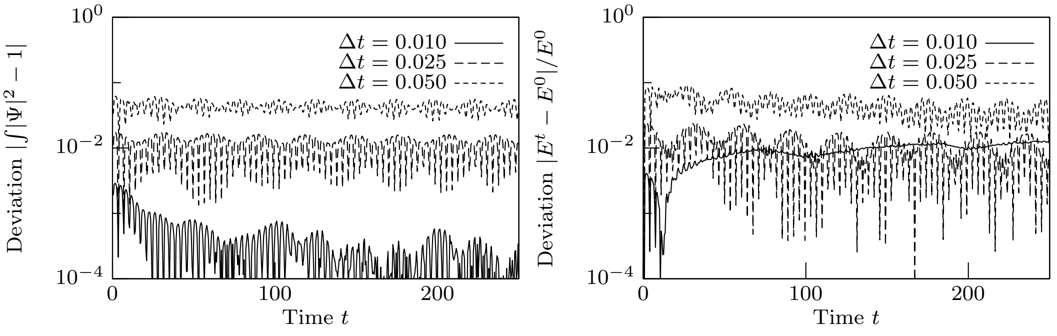
5.3.2 Convergence of the linear solvers

The conjugate gradients-like methods in section 4.2.1 have been tested on various coefficient matrices produced by the nonlinear methods in the previous section and in the next section. The detailed results are found in appendix B.2.

As a glance at the tables in the appendix shows, it is hard to draw general conclusions from this material, because of the number of different trends. If we for example limit ourselves to one linear solver, the BiCGStab method, we see that

- The RILU preconditioner is very effective, except for the Newton methods on the stationary problem, where the effect is small (fine grid) or negative (coarse grid).
- The classical SSOR preconditioner is usually a bit less effective than the RILU preconditioner, except for the explicit eigenvalue Newton's method and for time evolution using large step sizes, where it fails to converge in reasonable time.
- The number of linear iteration increases with the number of grid points (decreasing h) for the inverse power method and Newton methods in the stationary problem, but decreases for the imaginary time methods; in time evolution, the number of iteration decreases with increasing grid size when Δt is low, but increases when Δt is high.
- The linear solver converges quicker for short imaginary time steps than for long on fine grids, but not on coarse grids.

Therefore, to find the optimal linear solver, we must first decide which nonlinear solvers to use, at what grid size, and with what time steps. We did not want to limit our investigations in this way, and thus we looked for a linear solver that performs reasonably well in all cases. We settled for BiCGStab, because even when it fails to converge, it reduces the residual somewhat, so that the next non-



(a) Norm

(b) Energy

Figure 5.10: Evolution of conserved quantities in the Crank-Nicholson method over long time. The initial state at $t = 0$ is an equal mix of the nonlinear ground state and the first excited state. The simulations are in 1D with $C = 50$.

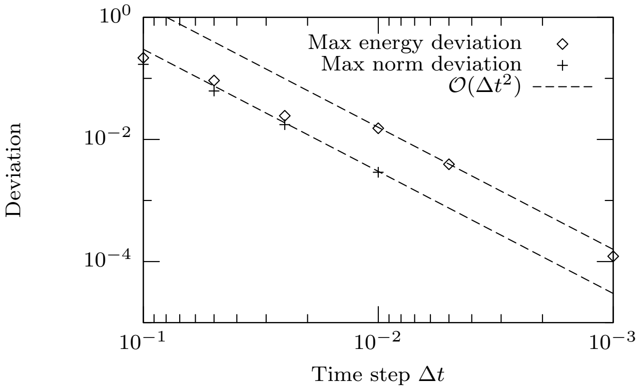


Figure 5.11: Maximum deviation from energy and norm for 10^4 steps of different step sizes Δt. Both the energy and the norm deviations seem to be bounded by a factor $\mathcal{O}(\Delta t^2)$. This is consistent with the results quoted in section 4.4.1. The simulations are in 1D with $C = 50$.

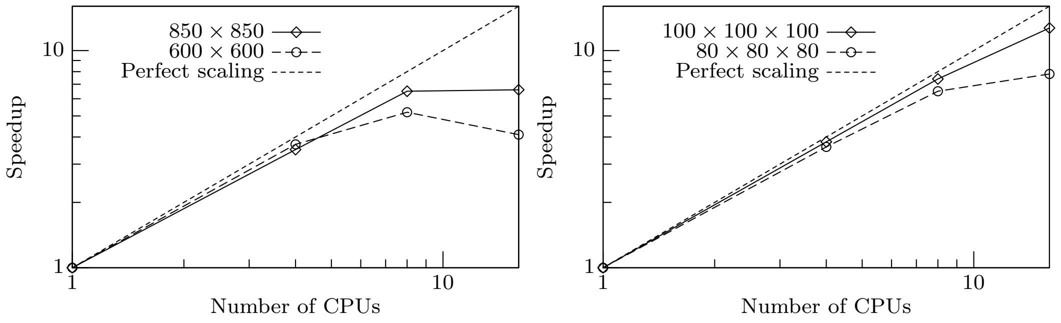
---

linear iteration is nevertheless better than the last. Conjugate gradients, while often faster, tends to fail spectacularly with greatly increased residual.

There is of course no reason to expect the conjugate gradient method to converge at all in these tests, since the coefficient matrices are not symmetric positive definite. But in most cases it does.

5.3.3 Stability of time evolution

Figures 5.10 and 5.11 show the time evolution of the conserved quantities of norm and energy. We saw in section 4.4.1 that the long term time evolution of these are bounded by $\mathcal{O}(\Delta t^2)$; our results are consistent with this bound.



(a) 2D grid (b) 3D grid

Figure 5.12: Parallel scaling on up to 16 processors.

5.3.4 Parallel efficiency

We have done some of the larger simulations, for example those reproducing the reference calculations in section 5.1.1, on several computers in parallel.

The computers that form the distributed computing nodes are also used for other purposes, and the network between them is shared. Thus it is difficult to get performance numbers that are reproducible. Nevertheless, we believe that the speedup shown in figure 5.12 is representative. On a loosely coupled distributed-memory computer, parallel scaling beyond perhaps 8 nodes in 2D or 16 nodes in 3D has quickly diminishing efficiency. On a shared-memory or tightly coupled parallel computer the scalability may be better, but we have not had the opportunity to test this.

We believe that parallelisation at a higher level, using for example the domain decomposition method, would allow the problem to be more scalable on distributed memory architecture.

The simulations we did in section 5.1.1 took about 8 CPU hours each, spread over 16 AMD 3000+ CPUs. A conservative nonlinear solution strategy of forced-phase inverse power iterations followed by implicit eigenvalue Newton iterations was used, for a total of around 35 nonlinear iterations. The linear solver was BiCGStab with no preconditioning. We estimate that time integration on the same grid using the same number of CPUs would proceed at a around one time step per minute.

Although execution efficiency is important for large and long-running simulations, this was not our main concern. What made parallel execution vital was the high memory requirements for 3D problems. In our implementation, we store two sparse matrices plus a few vectors. We consider the memory requirements of the simulations mentioned above. At two degrees of freedom per node, each vector uses

$$N_{\text{nodes}} \cdot N_{\text{dof}} \cdot \text{sizeof(real)} = 130 \cdot 130 \cdot 70 \cdot 2 \cdot 8\text{B} \approx 18\text{MB},$$

using double precision floating point numbers. This is negligible. The matrices, however, are not negligible. In table 4.2, the memory usage for general sparse matrices is estimated as $3N(p+2)^d/2$ in d dimensions. We find the memory usage of one coefficient matrix for quadratic elements to be

$$\frac{3}{2}N_{\text{nodes}}(p+2)^3 \cdot N_{\text{dof}}^2 \cdot \text{sizeof(real)} = \frac{3}{2}130 \cdot 130 \cdot 70 \cdot 4^3 \cdot 2^2 \cdot 8\text{B} \approx 3.4\text{GB},$$

so two such matrices add to around 6.8GB of memory, close to the measured total memory consumption of 7.0GB.[4] We did not have access to any computers with more that 1GB of memory installed. Therefore, to be able to solve problems of this size, we needed to spread it across several computers.

We note that in our tests the scalability of memory consumption was near perfect, indicating that not much data needs to be duplicated at the computing nodes.

5.4 Visualisation

While not part of the simulation per se, visualisation of the results is an important — and often time consuming — part of the process. In this section we look at some examples of how multidimensional and multivalued data may be visualised. Presenting such data in an intuitive yet informative way is as much an art as it is a science,[5] and as such there is no single best way to do it.

To state the obvious: how to visualise a set of results depends on what one wants to show. The purpose may be to highlight hidden patterns in the results; it may be to present an intuitive holistic view of the results, within which more detailed or quantitative results are easier to interpret; or it may be an attempt to spruce up a report with pretty pictures. We focus on the second goal, but hope that the two other are also served to some degree.

All 1D and some 2D plots are generated by `gnuplot`; most 2D plots and all 3D plots are generated by `opendx` (see bibliography for references).

5.4.1 Two-dimensional condensates

For spherically symmetric condensates, plotting of the density profile is trivial, since we only need to consider one dimension, either through integration or by

---

[4] Measured as the total of eight computing nodes, so any duplication overhead is included.

[5] For the reader interested in the display of general quantitive data beyond the standard approaches, reference (Tufte, 1997) is recommended.

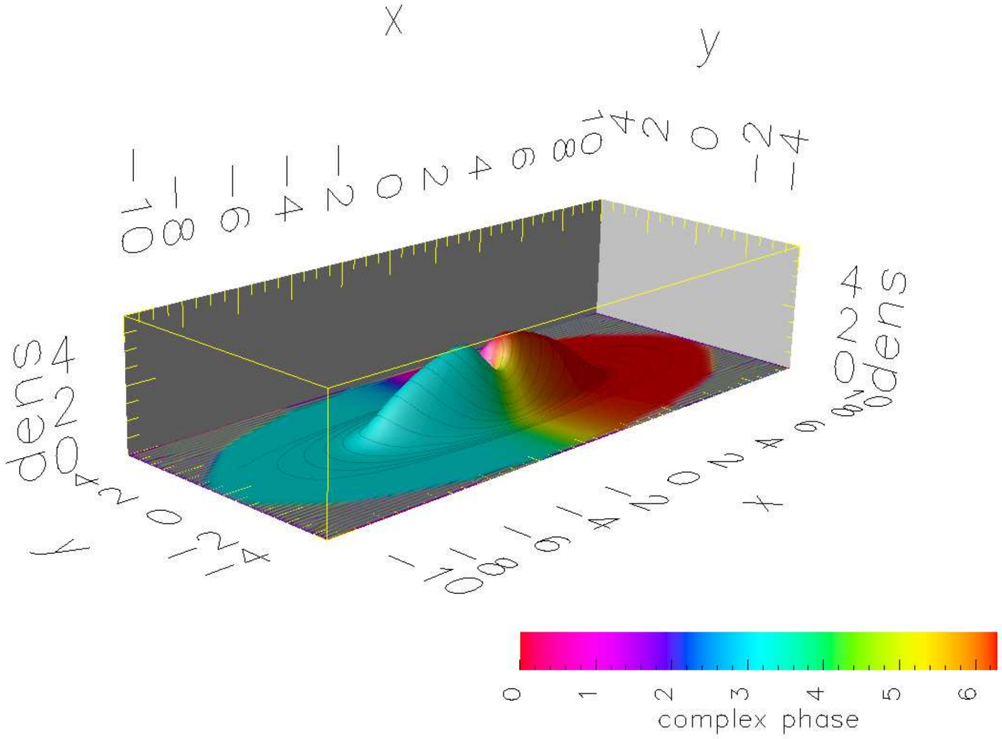
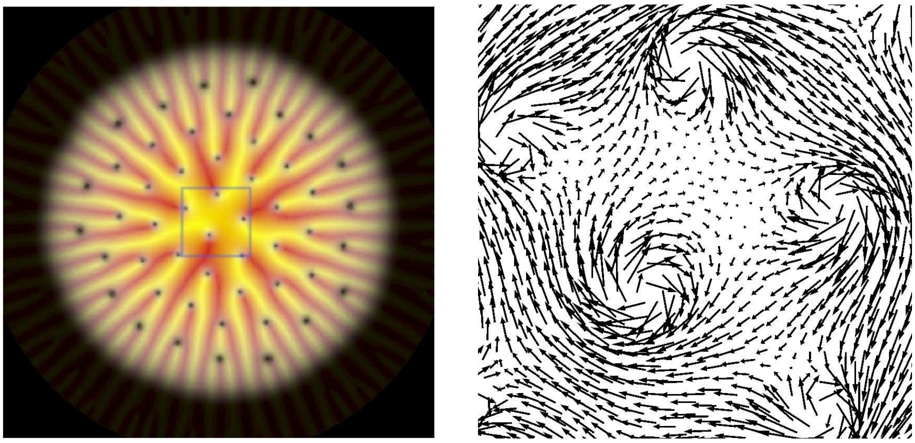


Figure 5.13: A single vortex in a 2D anisotropic trap with parameters $C = 150, \lambda = 4$. The density of the condensate is given by the elevation; the colouring shows the phase of the wave function. The concentric rings are the path of a massless particle moving in the condensate velocity field. The velocity is proportional to the change in phase, in the direction of greatest change.



(a) Phase of the vortex lattice (b) Velocity field of the central part

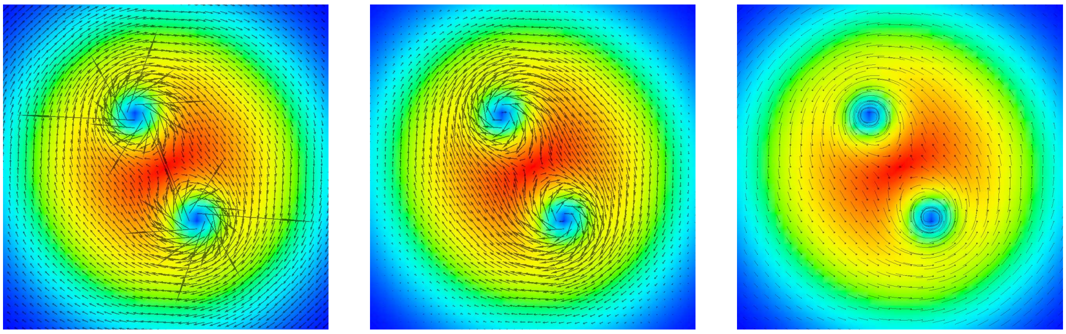
Figure 5.14: Phase and density of the vortex lattice in figure 5.5 (p. 88). The colour in (a) shows the phase of the condensate: Each red radial line signifies a 2π phase change and therefore one quantum of rotation. The density is shown by the intensity of the colour. Note that the full range of colours is not used for the phase, as it was in figure 5.13. In (b), the central part (boxed) of the same simulation is shown as a velocity field.

taking a slice through the origin. Non-symmetric 2D states also have well established plotting methods, by mapping the density to colours. Figure 5.5(b) (p. 88) shows an example of this. This is similar to how experimental results are visualised.

But in our simulations we can access the wave function that underlies the density, not only the density itself. We want to present this additional information, and one way to show it is through the complex phase. Figure 5.14(a) shows the phase imprinted on the density profile of the vortex lattice. As expected, it shows the phase change around each vortex to be 2π. However, it is hard to discern visually the rate of change of the colours. In figure 5.13 we used the full range of colours for the phase, and that makes it a bit easier; but in the vortex lattice case, that gives a very messy result.

Another way is to show the velocity field associated with the phase in the hydrodynamical formulation, see section 3.4.2. This, in principle, allows an easier way to see changes in phase than to try to discern the rate of change in colour. Figure 5.14(b) shows the velocity field of just the central part of the vortex lattice; but even this small section is messy because of the strong rotation near the vortex cores. It is obvious that the presence of vortices makes a direct plot of the velocity field less useful.

Figure 5.15 shows ways of fixing this problem. In (a), the velocity field is plot-



(a) Velocity field

(b) Momentum field, or density flow

(c) Massless particles moving in velocity field

Figure 5.15: Density and velocity fields of a two-vortex system, the equilibrium state for $C = 150, \Omega = 0.5$. Three different ways to visualise the movement of the condensate. A problem with plotting the velocity vector field is that it is diverges at the core of a vortex, and even if extreme values are culled, strong rotation near the core dominates the figure, as we see in (a). One solution is to plot the momentum field instead, figure (b), but this obscures the movement where density is low at the edges. Figure (c) plots the streamlines instead; since they follow the curl of the velocity field they stay confined near the vortex core.

ted, with the dominating velocities near the vortex cores,[6] while figure (b) shows the momentum field — that is, the velocity field multiplied by the density. The momentum plot is much better behaved, since the very high velocities near the core are matched by low densities. However, this makes it hard to see the movement at the surface of the condensate, where the low density is not matched by divergent velocity. Thus, a third approach is shown in figure (c), using streamlines. The streamlines are defined as the path of massless particles moving in the velocity field, and as such the high velocities near vortices do not dominate the visualisation because the particles there move in circles.

5.4.2 Three-dimensional condensates

When possible, the best way to visualise three-dimensional condensates is the simplest, namely either to integrate out some of the dimensions or to look at a 2D slice and visualise it as described in the previous section. In figures 5.3 and 5.4 (pp. 86–87), the integration approach is used. In this section we concentrate on volume visualisations.

Again the vortices are a challenge, in this case because they are inside the condensate, obscured by the higher densities that surrounds them. We present two

---

[6] Some even higher velocities inside the core have been removed from the plot.

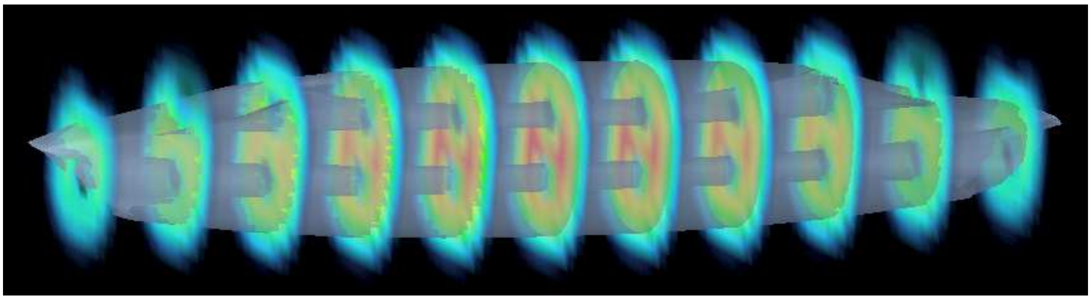


Figure 5.16: Double vortex in an elongated rotating trap, rendered using a monochromatic, semi-transparent isosurface, with a number of slices showing the density distribution.

methods that allow 3D condensates containing vortices to be visualised.

Figure 5.16 shows a double vortex in a 3D cigar-shaped trap. This figure combines two methods. The first is to use an isosurface, a surface that is drawn to cover points at equal density. When the isosurface is made translucent, it allows the vortex structure to be seen. The same technique is used in figure 5.6 (p. 89) to accentuate the vortex structure. This does not, however, give any insight into how the density varies outside the isosurface. Therefore we have in figure 5.16 also drawn slices through the condensate at intervals along the z axis, which shows the density distribution. These slices could equally well be used to represent the phase, in the same way as figure 5.14(a).

The other method is due to Ketcham et al. (2001), and an example is shown in figure 5.17; another in figure 5.7 (p. 90). It uses so-called inverse density rendering, which means that the density is volume rendered, but that higher densities are more transparent, lower densities are more opaque. While this makes vortices inside the condensate visible in contrast with the transparent higher densities around them, it has the problem that the empty space around the condensate is totally opaque, blocking all view. Therefore the Thomas-Fermi radius is used to clip the object, and nothing outside the radius is rendered. The phase is typically used for the colour channel of the volume rendering.

Visualising dynamics

Again we stress this: If a more concise quantity can be used to characterise the system, it is usually best to present this. For dynamics, examples of such quantities may be center-of-mass position or energy. While less visually pleasing, they often provide more insight. Nevertheless we sometimes want to look at the time development of the full system, maybe to gain intuition into how such characterisation can be achieved.

For time development of 2D systems, we can use the trick of considering the time

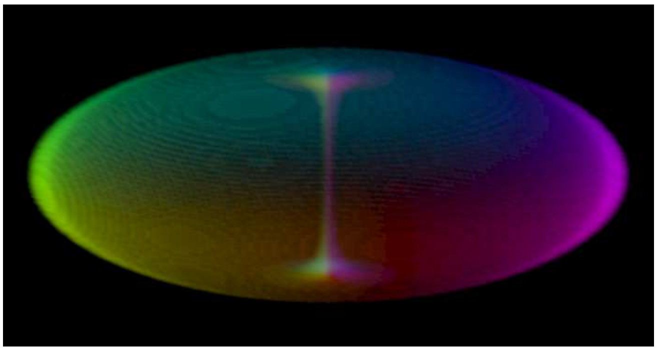


Figure 5.17: Inverse density rendering of the single-vortex condensate of figure 5.4 (p. 87). Clipped to the Thomas-Fermi radius, inverse density rendered and coloured according to phase.

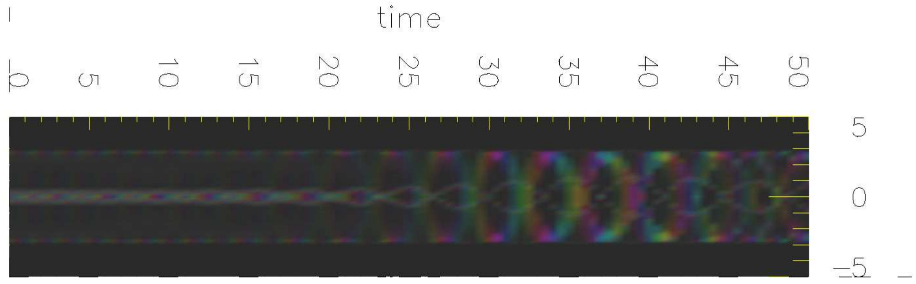


Figure 5.18: Decay of double-quantised vortex visualised using phase mapping. The solution is clipped at the Thomas-Fermi radius, and the interior is volume rendered with inverse density for opacity and phase for colour. The time series shown is the same as in figure 5.6 (p. 89).

---

dimension as the third spatial dimension, and use the 3D rendering techniques of the previous section. Figure 5.6 (p. 89) is an example of this, where a translucent isosurface combined with volume rendering of the density shows the vortex splitting and the surface excitations. Figure 5.18 shows the same sequence by inverse density rendering, as described in the previous section. This illustrates a weakness of this technique for non-stationary states: the Thomas-Fermi radius is no longer a good approximation of the surface of the condensate. Still, the parts of the surface waves that are inside the limit are clearly seen.

For visualisation of 3D dynamics, animation remains the only alternative.

Chapter 6

Conclusion

In this thesis we have investigated the use of the finite element method for the nonlinear Schrödinger equation. We have implemented and tested a solver using the `diffpack` toolkit, and applied the solver to a few practical examples of the Gross-Pitaevskii and modified Gross-Pitaevskii equation describing dilute-gas Bose-Einstein condensation. We have implemented and evaluated various linear and nonlinear solution methods for the stationary problem, and preliminary results for time evolution have shown the nonlinear Crank-Nicholson time integrator to be stable and reasonably efficient.

We saw in chapter 3 how Bose-Einstein condensates at zero temperature can be described by a mean-field approximation, and how the variational problem generates a nonlinear partial differential equation on the same form as the nonlinear Schrödinger wave equation. In chapter 4, we introduced the finite element method for variational problems and for partial differential equations, and investigated linear and nonlinear methods for solving the resulting generalised eigenvalue problem and integrating it in time.

Using these methods, we reproduced results from the finite difference calculations in (Nilsen et al., 2005), and from experiments in (Ketterle, 2001), and tested numerically some of the analysis in chapter 3. We also collected convergence data and compared them to the convergence analysis in chapter 4. This showed that our implementation worked as expected.

Numerical outlook. It is clear at this point that the finite element method is a powerful one, but it is not so clear what its advantages are in the current model problem, since we have not really played to its strengths. Most of our experiments have been on simple geometries — in practice unbounded ones, as we place the boundary where the density is negligible — and with regular grids.

We believe, however, that the flexibility of the finite element method makes it worthwhile in the general case. Its flexibility in geometry modelling and locallyadaptive grids makes modelling for example of investigation of superfluid flow with obstacles more accurate at minimal cost. Introduction of element types specialised to the problem at hand is also a promising approach; we saw in chapter 4, and verified in chapter 5, that just using quadratic instead of linear elements brought us a higher order accuracy in the eigenvalues. It is possible that using Hermitian elements which are continuous in the derivatives gives even better results.

Whether we use finite elements or finite differences for the discretisation, we need to solve linear systems of equations efficiently. Our analysis and experience have not picked any clear best choice among the iterative methods that we have considered, but our experience is that the BiCGStab iterative linear solver is reliable and efficient enough for practical use, see section 5.3.2. A natural next step would be to consider multigrid methods, since they are known to have near-optimal linear convergence properties (Barrett et al., 1994). There are also multigrid methods for the nonlinear problem, but we do not know whether these method allows us enough control over which solution, i.e., which eigenvalue, is found. This would be the topic of an investigation in itself.

While the topic of parallelisation has not been given much space in this thesis, it becomes essential when we want to simulate long time development, or calculate on large grids. We have used the recipe of parallelising on the lowest level, that of linear algebra. While this is simple and foolproof, it is likely that scaling beyond 16 processors or so requires a higher degree of decoupling of the nodes, and thus requires parallelising at a higher level. Domain decomposition is such a method, and it is well suited to use with the finite element method (Cai et al., 2003). Whether it is possible to use domain decomposition on the full nonlinear problem, or if it should be restricted to the linear sub-problems, again becomes a question of control of convergence to the desired eigenvalue.

Finally we found the Crank-Nicholson (trapezoidal rule) time integration method to be stable for long time evolution. This method preserves the Hamiltonian structure of the problem (Hairer et al., 2000), at the price of having to solve a nonlinear system of equations at each time step. Thus, further investigation of time integration methods might focus on faster methods, perhaps through time-splitting techniques, rather than more accurate ones.

Physics outlook. Most simulations to date of Bose-Einstein condensates have used simple geometries, suitable for treatment by finite differences and spectral methods. This is of course no coincidence, since it fits well with the conditions in experiments — electromagnetic trapping in a vacuum cell — and simulations are primarily used to understand previous experiments and suggest new avenues of experimentation. Simulations of conditions that are far from what is realisable are less interesting.

Nevertheless, we believe that when Bose-Einstein condensation in the future is used in applications, the flexibility of the finite element method will be valuable.

Use of BEC on chips has recently been demonstrated to be a promising method for measuring magnetic fields (Wildermuth et al., 2005), and implementation of an interferometer has also been achieved (Wang et al., 2005). Both of these examples show the need for methods to solve the Gross-Pitaevskii equation on more flexible domains.

Appendix A

Implementation Notes

We have implemented a solver for the modified Gross-Pitaevskii equation, derived in chapter 3, using the numerical methods described in chapter 4. Some results of experimenting with this solver were shown in chapter 5. In this chapter, we shall describe the implementation. The code itself is available for inspection at the project homepage Haga (2006).

The implementation is in C++, utilising the `diffpack` library, kindly provided by Simula Research Laboratory (see bibliography for references). Diffpack is used for nearly everything: matrix storage, building the finite element matrices, solving the linear systems, parallelisation, and general structure. Since diffpack is so central to the implementation, we describe briefly the parts of it that we have used.

A.1 Diffpack

Diffpack is a library that covers many aspects of the solution of PDEs. For complete coverage, we refer to the very readable introduction (Langtangen, 2003), which also includes a number of programming examples. Diffpack uses the object oriented paradigm, making it easy to integrate specialised methods when this is required.

Figure A.1 shows the central diffpack classes that we use. We start by describing the classes that we inherit from, which are also those that control the flow of execution.

`SimCase` This class contains basic diffpack infrastructure, such as the menu system. Most diffpack applications inherit this class, and must implement the method `solveProblem` method which is called from the menu system to solve the problem. In our case, solving the problem implies calling

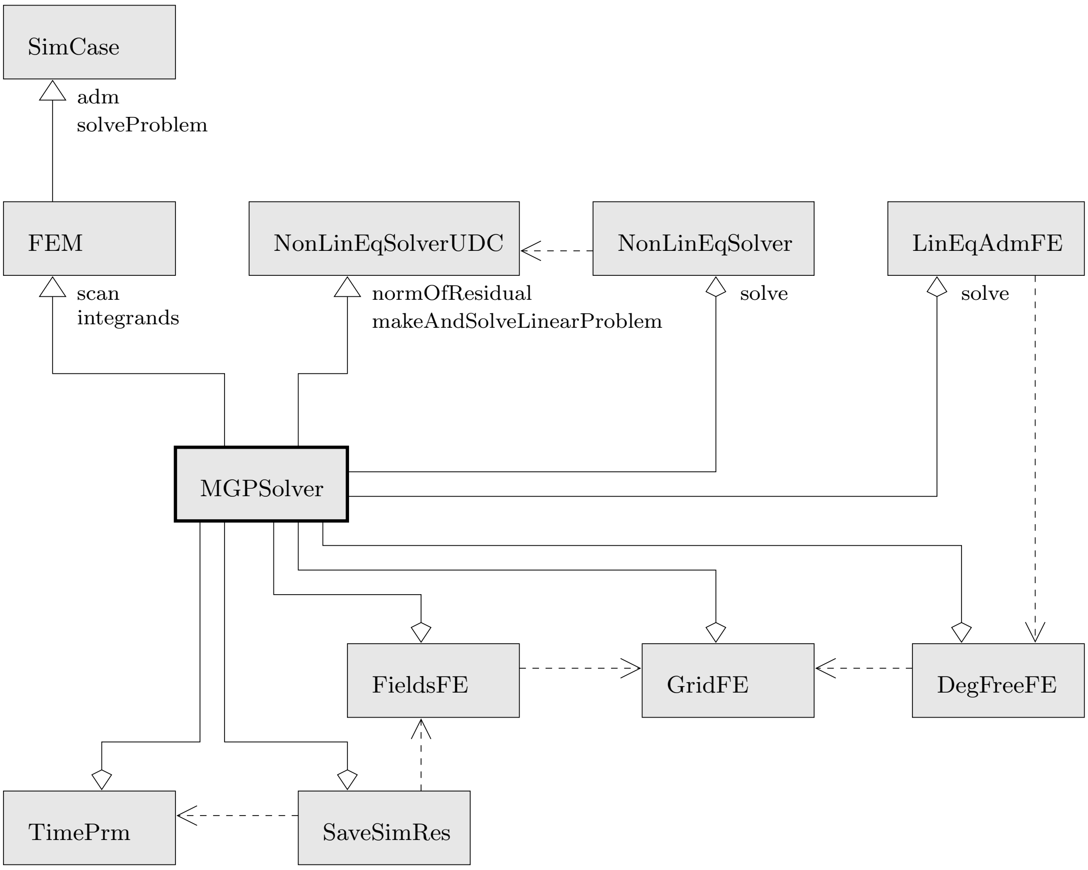


Figure A.1: Simplified class diagram. The relationship between our class (bold) and the central functional diffpack classes. Data structures and parallel execution administration classes are not included. Solid arrows represent "is-a" relationships, solid diamonds represent "has-a" relationships, and dashed arrows represent "uses" relationships. Class names are inside boxes and methods are listed below the boxes. Only those methods that are either overridden in MGPSolver or otherwise central to program flow are listed.

`NonLinEqSolver::solve`, to handle the problem as a nonlinear one. In addition, we implement the `adm` method to set menu choices for our problem parameters.

`FEM` The `FEM` class represents the finite element method. By inheriting this class and implementing the `integrands` method, we get the logic to perform integration over the finite elements and assemble the coefficient matrix. We also implement the `scan` method, in which general setup is performed.

`NonLinEqSolverUDC` User-defined code for the nonlinear equation solver. Nonlinear problems must implement two methods that tells the nonlinear solver how to solve the problem. By implementing the method `makeAndSolveLinearProblem`, we tell the solver how one nonlinear iteration is set up; and `normOfResidual` allows the solver to apply its stopping criteria.

The classes that we call upon to solve subproblems are

`NonLinEqSolver` We call upon this class to handle the nonlinear iterations; it, in turn, calls our class to set up and solve the linearised system and to report the current nonlinear residual. While several variants of this class are available to choose from at runtime, we only use the one meant for Newton-type solvers, and make sure that it is actually the correction vector that we calculate in each iteration, not the next iterative solution.

`LinEqAdmFE` This class handles the linear solvers. Through the `DegFreeFE` class it knows the size and structure of the matrices, and can thus handle the storage of the vectors and the coefficient matrix. We call its `solve` method to solve a linear system of equations.

`DegFreeFE` The `DegFreeFE` class handles the size and structure of the unknowns of the system, given a `GridFE` and a number of unknowns per grid node. It is used among other things to convert between a vector of unknowns and a `FieldsFE` representation.

`GridFE` Represents a grid, and its associated node numbers and coordinates. The grid may be regular or irregular.

`FieldsFE` This is a collection of `FieldFE` objects, each representing a nodal degree of freedom. In our case there are two `FieldFE` objects, one for the real part and one for the imaginary part of the wave function. From the `FieldsFE` object, we can retrieve the field values at any coordinate inside the domain, interpolated by the finite element basis methods.

`TimePrm` The current time, and information about time step lengths and so on, are kept in this class.

`SaveSimRes` This is a utility class which allows us to save the fields to a `simres` file. Simres is the native file format for data in diffpack.

The main data types and structures are

`NUMT` The data type used for the fields; `real` or `Complex`. Both are double precision.

`Matrix(NUMT)` This is the general matrix class in diffpack. If the creation of the matrices is left to the `LinEqAdmFE` class above, the type of the matrix is runtime configurable. All the matrix types discussed in section 4.2.3 are available, and usually we do not need to know in the code which we are using. This is no longer true if we need to manipulate the matrix structure, as with the explicit eigenvalue Newton's method, or if we need to access the individual matrix elements efficiently.

`MatSparse(NUMT)` This is the matrix class that we actually use. It is implemented using compressed row storage.

`Vec(NUMT)` A vector class.

Much of the behaviour of these classes is configurable at runtime through the menu system. This includes the linear and nonlinear method and stopping criteria, the matrix storage scheme, space and time grid specification, and so on. Through compile-time options debugging, matrix and vector index checks, real or complex numbers and so on can be controlled.

We can also define our own classes that reimplement parts of the library, for example to define our own finite element types.

A.2 The solver

A.2.1 Numerical integration over the finite elements

Numerical integration is performed in the method `integrands`, which is overridden from the class `FEM`. This method is called by the diffpack element assembly methods, and its purpose is to add the local contribution of the integrand to the element matrix.

Since we need to build a number of different matrices, we use a switch statement inside the `integrands` method. The switch statement decides, based on the two class-scoped variables `what` (of type `CalcType`) and `when` (of type `TimeState`), which integrand to calculate. The two types are defined as follows:

MGPSolver.h

```
enum CalcType  { CALC_M, CALC_K, CALC_dK, CALC_Lz };
enum TimeState { TIME_INITIAL_IMAG, TIME_INITIAL_NEWTON,
                 TIME_INITIAL_SUCC_SUBST, TIME_INITIAL_ARPACK,
                 TIME_INITIAL_X_NEWTON, TIME_PROPAGATION };
```

The purpose of the `CalcType` variable is perhaps obvious, but the `TimeState` one is not, at least not in this context. The reason it is used here is that for coupled real/imaginary equations, multiplication by i becomes multiplication by a 2×2 block matrix,

$$\mathbf{i} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \tag{A.1}$$

which is easy to perform on the element matrix, but in general hard to do efficiently on the full matrix. Thus, we do it in `integrands`. If we are currently solving for the initial state, we calculate $\mathbf{K}, \delta\mathbf{K}$, or $\mathbf{L}_z$; in time propagation, we calculate $i\mathbf{K}, i\delta\mathbf{K}$, or $i\mathbf{L}_z$.

Before showing the numerical integration itself, we should also present the helper functions used to address the real and imaginary parts of the matrix. We find these useful to keep track of the indices.

MGPSolver.h

```
static const int R = 1;
static const int I = 2;
static inline int Re(int i) { return 2*i-2+R; }
static inline int Im(int i) { return 2*i-2+I; }
```

Thus, for a vector `vec`, representing nodes each having a real and an imaginary degree of freedom, `vec(Re(k))` is the real-valued part of node k, and `vec(Im(k))` is the magnitude of the imaginary-valued part.

We look at how one of the integrands is calculated, $\delta\mathbf{K}$. In equation (4.119), it was found to be

$$\delta K_{ij} = \int C(2 + 3D|\Psi|)\chi_i\chi_j N_i N_j \,\mathrm{d}\mathbf{x},$$

with χ being either the real or the imaginary part of Ψ, depending on the index. The effect of this splitting is seen in lines 10–14 in the listing below. Lines 19–23 are the same, but multiplied by equation (A.1). The variables `re_pt` and `im_pt` contains the values at the current integration point, from the previous iteration of the nonlinear solver.

In MGPSolver::integrands()

```
1  case CALC_dK:  // dF_i/du_j, multiplied by i if TIME_PROP
   {
     for (int i = 1; i <= nbf; i++)      // for each basis function i
       for (int j = 1; j <= nbf; j++) { // for each basis function j

6        real val = C * (2 + 3*D*sqrt(sqr(re_pt) + sqr(im_pt)))
                    * fe.N(i) * fe.N(j) * detJxW;

         if (when != TIME_PROPAGATION) {
           elmat.A(Re(i),Re(j)) += val * sqr(re_pt);
11         elmat.A(Im(i),Im(j)) += val * sqr(im_pt);

           elmat.A(Re(i),Im(j)) += val * re_pt * im_pt;
```

```
        elmat.A(Im(i),Re(j)) += val * re_pt * im_pt;
      }
      else {
        // Multiply by i

        elmat.A(Im(i),Re(j)) += val * sqr(re_pt);
        elmat.A(Re(i),Im(j)) -= val * sqr(im_pt);

        elmat.A(Re(i),Re(j)) -= val * re_pt * im_pt;
        elmat.A(Im(i),Im(j)) += val * re_pt * im_pt;
      }
    }
}
```

The other integrands are calculated in a similar way. The choice of integration points, calculation of the weighted determinant `detJxW` of the transformation to local coordinates, and assembly of the global matrix from the element matrices are all handled automatically by diffpack.

A.2.2 Building the linear system

After the relevant matrices have been built, we combine them to create the linear system to solve in each iteration of the nonlinear method. That is, we create the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ to fit the linear form

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{A.2}$$

or, for the generalised linear eigenvalue solver, arpack, the form

$$\mathbf{A}\mathbf{x} = \mu\mathbf{B}\mathbf{x}. \tag{A.3}$$

In the latter case we must also modify the matrices to enforce the boundary conditions explicitly.

How to create $\mathbf{A}$ and either $\mathbf{b}$ or $\mathbf{B}$ is covered in section 4.3. We show just one example here, for the implicit eigenvalue Newton's method. In this case the code is straightforward,

In MGPSolver::makeAndSolveLinearProblem()

```
switch (when) {
case TIME_INITIAL_NEWTON:
{
  what = CALC_dK;                   // Calculates K+dK
  makeSystem(*dof, lineq->A());

  lineq->A().add(lineq->A(), -mu, *M);
  lineq->b().add(mu, Mu, -1.0, Ku); // Mu and Ku are matrix-vector
```

```
                                            // products calculated previously

  dof->insertEssBC(lineq->bl().getVec());
  linear_solution.fill(0);

  lineq->solve();
}
```

An added complication arises for the explicit eigenvalue Newton's method, since it requires storage in the matrix for an additional row and an additional column. We can achieve the extra storage without increasing the dimension of the matrix by using a row and column normally occupied by a boundary node, and resetting the nodal value after the linear solver has finished. We must still reserve space in the sparse matrix structure, a task which is usually performed automatically by diffpack by its knowledge of the structure of the grid.

The extra matrix element storage is created as follows, provided the last node is a boundary node. This seems to always be the case on for a single regular grid, but a check should be performed. It is not implemented for parallel computation. The data structure that we manipulate is structured as a compressed row storage scheme, depicted in figure 4.8 (p. 66); the `irow` vector corresponds to the one marked *index*, and `jcol` corresponds to the *column* vector. Since we only manipulate the structure, *values* in the figure is not manipulated, it is instead created later by the MatSparse class, with the same length as the *column* vector.

In MGPSolver::scan()

```
SparseDS pattern(noDof, 2*noDof-2);

int entry = 1;
// Create storage for column vector
for (int row = 1; row <= noDof-1; row++) {
  pattern.irow(row) = entry;
  pattern.jcol(entry++) = noDof;
}

// Create storage for row vector
pattern.irow(noDof) = entry;
for (int col = 1; col <= noDof-1; col++)
  pattern.jcol(entry++) = col;
pattern.irow(noDof+1) = entry;

// Merge with precomputed pattern
Handle(SparseDS) pat; pat.rebind(new SparseDS());
pat->createUnion(lineq->Al().getMatSparse().pattern(), pattern);
lineq->Al().getMatSparse().redim(*pat, false);
```

Although diffpack provides infrastructure for switching between nonlinear solution strategies, we found it easier to pretend that all methods calculate a correction vector, instead of some methods calculating the correction and some calculating the next solution. Thus, for some methods we also perform a subtraction of the previous result after the linear solver has done its job.

Finally, the result must be scaled, and the eigenvalue must be updated. The former is done through dividing by the square root of its norm,

$$\hat{\mathbf{u}} = (\mathbf{u}^{\mathrm{T}}\mathbf{M}\mathbf{u})^{-1/2}\mathbf{u}, \tag{A.4}$$

while the latter is achieved through taking the Rayleigh quotient, simplified by just having set its divisor to 1,

$$\mu = \hat{\mathbf{u}}^{\mathrm{T}}\mathbf{K}\hat{\mathbf{u}}. \tag{A.5}$$

A.2.3 Nonlinear solvers

Apart from setting up the linear systems for each nonlinear iteration, not much needs to be done for the nonlinear solvers. By implementing the `normOfResidual` method inherited from `NonLinEqSolverUDC`,[1] diffpack has enough information to drive the iterations until specified convergence criteria are reached. Depending on which nonlinear method is being used, we may need to calculate the residual, but that is simple:

MGPSolver::normOfResidual()

```
switch (when) {
case TIME_INITIAL_IMAG:
case TIME_INITIAL_SUCC_SUBST:
case TIME_INITIAL_ARPACK:
  lineq->b().add(Ku, -mu, Mu); // Ku and Mu are K*u and M*u
}
return pl_norm(lineq->bl(), normt);
```

The function `pl_norm` is just a simple wrapper around the standard vector norm, to handle the parallel case.

We have also implemented a line search strategy for cases where the full nonlinear step leads to divergence. The method, adapted from (Dennis and Schnabel, 1983), fits a cubic polynomial to the residuals along the search direction and uses the minimum of the polynomial as the step length. While this approach works well to avoid divergence, it is only needed in extreme cases; and in those cases, stagnation in a local minimum is often the result. The line search method is in a separate class, `LineSearch`, which derives from `MGPSolver` and overrides the appropriate methods.

---

[1] UDC is short for User Defined Code.

A.2.4 Parallel execution

We implemented parallel execution at the level of linear algebra. Each participating computing node keeps a portion of the matrices and vectors. Whenever a matrix-vector product or an inner product is requested, the computing nodes must communicate for those vector elements that are shared between different finite elements. Thus this method requires quite a lot of communication in the linear solver phase.

The implementation was trivial, since diffpack supports this mode of parallelisation with few changes to the code. The total modification to our code was a few tens of lines, none of them dealing with details of the communication or implementation. We followed the recipe found in (Cai et al., 2003).

While this approach showed good performance for up to 16 CPUs on large 3D grids, as seen in section 5.3.4, we believe that parallelisation on a higher level is required for scalability beyond this.

A.2.5 Parameters

The problem parameters are the parameters that describe the system that we simulate. In particular, they are the various coefficients in the Hamiltonian, equation (3.109) and the rotating frame Hamiltonian, equation (3.117).

Parameter	Symbol	Description
`C`	C	Strength of second order mean field term
`D`	D	Strength of third order mean field term
`lambdasq`	λ^2	Trap anisotropy parameter
`Omega`	Ω	Rotation of lab frame

The simulation parameters, listed below, describe what to simulate, and how to do it.

Parameter	Description
`IC`	Initial conditions; allowed values are `Gauss`, `Rot`, `Rand`, or the name of a file containing the nodal values.
`strategy`	A string describing how to solve the system; see next table
`lumping`	1 generates a row-sum lumped mass matrix, 0 does not.
`time integ`	A regular diffpack time specification, listing time step size and interval to simulate
`gridfile`	A regular diffpack grid specification, listing a file name or a grid specification on PreproBox form

The `strategy` parameter takes a list of actions on the form `action/parameter`, that is parsed at runtime to switch between solvers, apply perturbations, and so on. The actions are as follows:

Action	Parameter	Description
`arpack`	Number of eigenvalue	Use successive substitutions as nonlinear solver
`dump`	What to dump; optional	Dump current solution to file
`idt`	Size of imaginary time step	
`imag`	Apply a phase of m rotations; optional	Use imaginary time evolution as nonlinear solver
`newton`	—	Use implicit eigenvalue Newton's method as nonlinear solver
`origin`	Distance in x direction	Shift the origin of the applied phase for `imag` and `picard`
`perturb`	Magnitude of perturbation	Mix values from a file named `perturb.m` into the current solution
`picard`	Apply a phase of m rotations; optional	Use the inverse power method as nonlinear solver
`relax`	Step length	Set the nonlinear relaxation parameter
`restart`	Threshold value	Start from the beginning of the `strategy` string if current nonlinear error is higher than threshold
`shift`	Distance in x direction	Shift the current solution in space
`time`	Time	Do not process the `strategy` string further until after the specified time
`until`	Threshold value, or number of iterations	Do not process the `strategy` string further until the current nonlinear error is below this threshold
`xewton`	—	Use explicit eigenvalue Newton's method as nonlinear solver

An example of a strategy string for studying the decay of a double-quantised vortex may be

`relax/.8 picard/2 until/1e-5 relax/1 newton time/0 perturb/.01`,

which sets a step length of 0.8, uses the inverse power method[2] with a double applied rotation (4π phase change around the origin), and iterates until the nonlinear error falls below 10^{-5}; then the step length is set to 1, and the method is switched to implicit eigenvalue Newton's methods. When the initial state has converged, according to standard diffpack criteria, and the next time step is started, a perturbation of magnitude 10^{-2} is applied.

In addition to these parameters that MGPSolver itself uses, the various diffpack classes must in some cases be changed from their defaults for proper operation.

---

[2] This method is called *picard iterations* in the code.

We list here the recommended changes.

input.i

```
sub NonLinEqSolver_prm
 set nonlinear iteration method = NewtonRaphson
 set max no of eps increases = 100
ok

sub LinEqAdm
 sub Matrix_prm
  set matrix type = MatSparse
 ok
 sub LinEqSolver_prm
  set basic method = BiCGStab
  set max iterations = 100
 ok
 sub Precond_prm
! set preconditioning type = PrecRILU
 ok
ok

sub SaveSimRes
 set field storage format = BINARY
 set grid storage format = BINARY
ok

sub GridPart_prm
 set number overlaps = 1
ok
```

A.2.6 Arpack integration

We integrated the arpack linear eigenvalue solver with diffpack using the procedure outlined by Kvaal (2003), with modifications.

A.3 Post-processing

We have tried to only do simulation in the simulation application, and postpone any processing of the results to a later stage.

The simres databases that diffpack saves are well suited to this, as they save the field and the grid at chosen times, in a format which is easy to read back in and process further. For this purpose we have created a standalone application which does things like calculate expectation values of various operators, integrate in spherical and z directions to create graphs like in figures 5.3 (p. 86) and 5.4 (p. 87), and so forth.

We have also created a filter that translates simres data files into dx data files, in order to use the `opendx` graphics package.

Appendix B

Auxilliary Results

B.1 Error estimation

In order to estimate the uncertainty in our calculations, we try to find the error as a function of the mesh size h. In section 4.1.8, we saw that both the error of the discrete solution in energy norm and the error in the eigenvalues are given by

$$\epsilon \leq ch^{2(k-m)} \mu_l^{k/m}. \tag{B.1}$$

for the l^{th} eigenvalue. We assume this is also true for the components of the total energy — the kinetic energy, the potential energy, and so on.

Thus, since μ_l is a constant for a given state, we can for sufficiently small h write ϵ as a function of h. For a given quantity Q, be it energy or eigenvalue, we can then write the discrete approximation as a perturbation of the exact solution as

$$Q^h \approx Q(1 + \epsilon(h)) = Q(1 + Ch^{2(k-m)}). \tag{B.2}$$

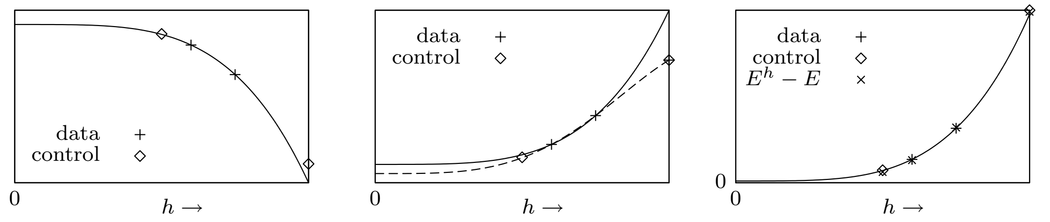
If we calculate Q^h at two different values of h, we find the estimated value of Q as

$$Q \approx \frac{h_2^{2(k-m)} Q^{h_1} - h_1^{2(k-m)} Q^{h_2}}{h_2^{2(k-m)} - h_1^{2(k-m)}}. \tag{B.3}$$

This is equivalent to fitting a polynomial curve $ah^{2(k-m)} + c$ for two values of h and extrapolating to $h \to 0$.

A problem is that we do not know if h is sufficiently small. Thus we should also calculate a third value, Q^{h_3}, at yet another grid size, to verify that the curve is reasonable.

In the case of the MGP ground state in section 5.1.1, we have $2(k - m) = 4$, thus the curve is fourth order. Figure B.1 shows three examples of the fitted curves, along with the calculated points. In all three cases, the middle two



(a) The kinetic term, E_{K} (b) The MGP term, E_3 (c) The virial error, $p^{-1}G$

Figure B.1: Examples of extrapolation curves. The curves are fitted from the two central points, the other points are included as reference. Figure (a) shows the kinetic energy, which is quite well approximated by a quartic curve. Figure (b) includes a quintic term in the extrapolation (dashed line), which takes it close to the inner control point. Figure (c) shows the remainder of the virial term along with a quartic curve; also plotted is the estimated error in total energy (circles), which is also here indistinguishable from the virial error.

Table B.1: Extrapolation of the ground state energies of the MGP to $h \to 0$, to measure the uncertainty in the results. The trailing + or − indicates the sign of the uncertainty ±1 in the last digit. All energies are per particle, in units of $\hbar\omega_\perp$.

	μ	E	E_{K}	E_{HO}	E_2	E_3
Base	15.45277	11.06116	0.353523	6.94099	2.516700	1.249944
Extrap.	15.45276	11.06114	0.353528	6.94096	2.516705	1.249942
Signif.	15.4528^-	11.0612^-	0.35352^+	6.9410^-	2.51670^+	1.24994

points ("data") are used as basis for fitting the curve, and the rightmost point is used to verify the fit. In (b), we judged the fit to be not good enough, and included a fifth order term,

$$Q^h \approx Q(1 + Ch^4 + Dh^5), \tag{B.4}$$

and used all three points to fit the curve. The leftmost point is in all three cases used only to check the resulting curve. We would normally not compute this point.

The calculations were done on grid sizes 130 × 130 × 70 and 104 × 104 × 56 for the original fit and 88 × 88 × 42 for the rightmost control point; a detailed calculation at 156 × 156 × 84 was used to check the results.

We believe this procedure gives a good estimate of the error. We have not used the results at $h \to 0$ directly, but only for the error calculation. In table B.1, we show the base numbers at 130 × 130 × 70 and the extrapolated numbers, and how we derive the significant digits from those.

Finally, figure B.1(c) shows that the error in the virial theorem follows closely the estimated error in total energy also in the nonlinear case.

B.2 Results for linear solvers

Listing of the results of a survey of iteration counts for a selection of linear solvers and preconditioners for different stages of the solution. All tests were performed on two different 2D grids with moderate nonlinearity, $C = 100$. The following were tested:

- Eigenvalue (stationary) solution with high initial nonlinear error (but not too high for the Newton solvers to converge: $\|\mathbf{u}\| \sim 10^{-4}$): Tables B.2 (coarse grid) and B.3 (fine grid).
- Eigenvalue (stationary) solution with low initial nonlinear error ($\|\mathbf{u}\| \sim 10^{-8}$): Tables B.4 (coarse) and B.5 (fine).
- Time evolution, initial Newton step: Tables B.6 (coarse) and B.7 (fine).

Table B.2: Linear solver convergence for initial state on coarse grid, high initial error. Number of linear iterations to solve for one nonlinear iteration. Triplets are for preconditioner none/RILU/SSOR. The grid is 20×20, $C = 100$.

	ConjGrad	GMRES(5)	GMRES(10)	CGS	BiCGStab
InvPow	15/4/4	25/10/20	30/20/20	9/2/3	10/2/2
ImpNewt	20/24/21	35/–/110	40/100/50	16/20/17	12/25/19
ExpNewt	23/23/–	290/–/–	50/110/–	18/24/–	17/19/–
Imag 0.1	18/3/6	25/10/15	30/20/20	11/2/3	12/2/3
Imag 0.01	18/2/6	25/10/15	30/20/20	11/1/4	12/1/3

Table B.3: Linear solver convergence for initial state on fine grid, high initial error. Number of linear iterations to solve for one nonlinear iteration. Triplets are for preconditioner none/RILU/SSOR. The grid is 80×80, with $C = 100$.

	ConjGrad	GMRES(5)	GMRES(10)	CGS	BiCGStab
InvPow	27/10/13	40/15/20	50/20/30	24/6/8	19/6/8
ImpNewt	43/33/48	95/200/230	70/60/100	44/27/29	35/27/46
ExpNewt	47/32/–	695/115/–	140/60/–	48/25/–	37/27/–
Imag 0.1	11/5/6	20/10/15	30/20/20	9/3/4	6/3/3
Imag 0.01	5/3/3	10/10/10	20/20/20	3/2/2	3/2/2

Table B.4: Linear solver convergence for initial state on coarse grid, low initial error. Number of linear iterations to solve for one nonlinear iteration. Triplets are for preconditioner none/RILU/Jacobi. The grid is 80×80, with $C = 100$.

	ConjGrad	GMRES(10)	CGS	BiCGStab
InvPow	4/1/1	20/20/20	4/1/1	2/1/1
ImpNewt	6/7/4	20/20/20	5/6/3	3/5/3
ExpNewt	9/7/–	20/20/–	9/8/–	6/6/–
Imag 0.1	18/3/6	30/20/20	11/2/3	12/2/3
Imag 0.01	17/2/6	30/20/20	11/1/4	12/1/3

Table B.5: Linear solver convergence for initial state on fine grid, low initial error. Number of linear iterations to solve for one nonlinear iteration. Triplets are for preconditioner none/RILU/Jacobi. The grid is 80×80, with $C = 100$.

	ConjGrad	GMRES(10)	CGS	BiCGStab
InvPow	8/3/3	20/20/20	9/2/3	4/2/2
ImpNewt	13/5/6	30/20/20	14/4/11	7/3/4
ExpNewt	16/4/–	30/20/–	18/3/–	11/3/–
Imag 0.1	11/5/6	30/20/20	10/3/4	6/3/3
Imag 0.01	5/3/3	20/20/20	3/2/2	3/2/2

Table B.6: Linear solver convergence for time evolving state on coarse grid. Number of linear iterations to solve for one nonlinear iteration. Triplets are for preconditioner none/RILU/SSOR. The grid is 20×20, with $C = 100$.

Δt	ConjGrad	GMRES(5)	CGS	BiCGStab
0.1	–/4/–	90/10/65	36/2/29	47/2/25
0.01	–/2/6	25/10/10	11/1/3	11/1/3
0.001	8/1/3	15/10/10	5/1/2	5/1/2

Table B.7: Linear solver convergence for time evolving state on fine grid. Number of linear iterations to solve for one nonlinear iteration. Triplets are for preconditioner none/RILU/SSOR. The grid is 80×80, with $C = 100$.

Δt	ConjGrad	GMRES(5)	CGS	BiCGStab
0.1	–/447/–	100/15/–	76/4/–	524/4/–
0.01	–/3/8	15/10/15	6/2/5	6/2/3
0.001	2/1/2	10/10/10	2/1/1	1/1/1

Appendix C

Mathematical and Physical Topics

C.1 The virial theorem

The quantum virial theorem is based on the property that the energy of the system must be invariant on a change of scale. As such it is applicable to all quantum systems, but its exact form depends on the Hamiltonian. We derive the virial equation for a Hamiltonian with homogenous polynomial potential and polynomial mean-field interaction.

We define the N-particle wave function

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N), \tag{C.1}$$

which is an eigenstate of the mean-field Hamiltonian

$$H = T + V + \sum_i U_i, \tag{C.2}$$

where $T = a\nabla^2$ is the kinetic term, $V = \sum_j b_j x_j^u$ is the potential term of order u, and $U_i = c_i|\Psi|^i$ is the mean-field term of order i.

We now look at the perturbed system where all coordinates are scaled by λ, and require that the eigenstate is recovered at $\lambda = 1$.

We define this perturbed state as

$$\Phi_\lambda(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) = \Psi(\lambda\mathbf{x}_1, \lambda\mathbf{x}_2, \ldots, \lambda\mathbf{x}_N). \tag{C.3}$$

The variational principle dictates that the quotient

$$R[\Phi_\lambda] = \frac{\langle\Phi_\lambda|H|\Phi_\lambda\rangle}{\langle\Phi_\lambda|\Phi_\lambda\rangle} \tag{C.4}$$

has a stationary point when Φ_λ is an eigenstate of H (and R is then the energy eigenvalue).[1]

Each of the terms of equation (C.2) can now be integrated for N particles in n dimensions. The potential term is

$$\begin{aligned}
\langle\Phi_\lambda|V|\Phi_\lambda\rangle &= \int \mathrm{d}^n\mathbf{x}_1 \ldots \int \mathrm{d}^n\mathbf{x}_N\, |\Phi_\lambda(\mathbf{x}_1\ldots\mathbf{x}_N)|^2\, V(\mathbf{x}_1\ldots\mathbf{x}_N) \\
&= \lambda^{-u} \int \mathrm{d}^n\mathbf{x}_1 \ldots \int \mathrm{d}^n\mathbf{x}_N\, |\Psi(\lambda\mathbf{x}_1\ldots\lambda\mathbf{x}_N)|^2\, V(\lambda\mathbf{x}_1\ldots\lambda\mathbf{x}_N) \\
&= \lambda^{-u}\lambda^{nN} \int \mathrm{d}^n\mathbf{y}_1 \ldots \int \mathrm{d}^n\mathbf{y}_N\, |\Psi(\mathbf{y}_1\ldots\mathbf{y}_N)|^2\, V(\mathbf{y}_1\ldots\mathbf{y}_N) \\
&= \lambda^{-u-nN}\langle\Psi|V|\Psi\rangle, \qquad \text{(C.5)}
\end{aligned}$$

where the change of variables $\lambda\mathbf{x}_i \to \mathbf{y}_i$ in the penultimate line is the key step of the calculation. The other terms are obtained in a similar way, with a factor λ^{-nN} from the change of integration variables and an additional factor $f(\lambda)$ from the scaling of the operator,

$$\begin{aligned}
\langle\Phi_\lambda|\Phi_\lambda\rangle &= \lambda^{-nN}\langle\Psi|\Psi\rangle = \lambda^{-nN}, \qquad \text{(C.6)} \\
\langle\Phi_\lambda|T|\Phi_\lambda\rangle &= \lambda^{2-nN}\langle\Psi|T|\Psi\rangle, \qquad \text{(C.7)} \\
\langle\Phi_\lambda|U_i|\Phi_\lambda\rangle &= \lambda^{ni/2-nN}\langle\Psi|U_i|\Psi\rangle. \qquad \text{(C.8)}
\end{aligned}$$

Note in equation (C.8), that U_i scales as $\lambda^{ni/2}$ instead of unity because of the normalisation condition on $|\Psi|^2$ (cf. the scaling in equation (3.103)).

Thus equation (C.4) can be written as

$$R(\lambda) = \langle\Psi|\lambda^2 T + \lambda^{-u}V + \sum_i \lambda^{ni/2}U_i|\Psi\rangle, \qquad \text{(C.9)}$$

with the condition that $\partial R/\partial\lambda = 0$ at $\lambda = 1$:

$$2\langle T\rangle - u\langle V\rangle + \sum_i \frac{ni}{2}\langle U_i\rangle = 0. \qquad \text{(C.10)}$$

The Gross-Pitaevskii equation (3.109) has quadratic potential, and the only non-zero mean-field terms are U_2 and U_3,

$$E_\mathrm{K} - E_\mathrm{HO} + \frac{n}{2}E_2 + \frac{3n}{4}E_3 = 0. \qquad \text{(C.11)}$$

Equations (C.10) and (C.11) are valid when the expectation values are taken at an eigenstate. In the numerical calculations, this quantity is a useful check of the accuracy of the results.

---

[1] In linear algebra, R is known as the Rayleigh quotient, and we saw it used it in this guise in section 4.1.8.

Bibliography

J. R. Anglin and W. Ketterle. Bose-Einstein condensation of atomic gases. *Nature*, 416:211–218, 2002.

`arpack`. URL `http://www.caam.rice.edu/software/ARPACK/`. Library for solving linear eigenvalue problems.

L. E. Ballentine. *Quantum Mechanics: A Modern Development*. World Scientific, 1998.

R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, PA, 2nd edition, 1994. URL `http://netlib2.cs.utk.edu/linalg/html_templates/Templates.html`.

BEC Homepage. URL `http://www.colorado.edu/physics/2000/bec/index.html`. Introduction to BEC with animations.

M. S. Berger. A Sturm-Liouville theorem for nonlinear elliptic partial differential equations. *Proc. Nat. Acad. Sci. USA*, 53(6):1277–1279, 1965.

J.-P. Blaizot and G. Ripka. *Quantum Theory of Finite Systems*. MIT Press, 1986.

N. N. Bogoliubov. On the theory of superfluidity. *J. Phys. (Moscow)*, 11(23), 1947.

X. Cai. Overlapping domain decomposition methods. In Langtangen and Tveito (2003), chapter 2.

X. Cai, E. Acklam, H. P. Langtangen, and A. Tveito. Parallel computing. In Langtangen and Tveito (2003), chapter 1.

E. A. Cornell and C. E. Wieman. Nobel lecture: Bose-Einstein condensation in a dilute gas, the first 70 years and some recent experiments. *Rev. Mod. Phys*, 74:875–893, 2002.

F. Dalfovo and M. Modugno. Free expansion of Bose-Einstein condensates with quantized vortices. *Phys. Rev. A*, 61(2):023605, 2000.

K. B. Davis, M.-O. Mewes, M. R. Andrews, N. J. van Druten, D. S. Durfee, D. M. Durn, and W. Ketterle. Bose-Einstein condensation in a gas of sodium atoms. *Phys. Rev. Lett.*, 75(22):3969–3973, 1995.

J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice-Hall, 1983.

`diffpack`. URL `http://www.diffpack.com/`. Library for solving PDEs.

Z. Dutton, N. S. Ginsberg, C. Slowe, and L. V. Hau. The art of taming light: ultra-slow and stopped light. *Europhys. News*, 35(2), 2004.

A. L. Fetter. Theory of a dilute low-temperature trapped Bose condensate. In Inguscio et al. (1998), pages 201–263.

A. L. Fetter and J. D. Walecka. *Quantum Theory of Many-Particle Systems.* McGraw-Hill, 1971.

S. Giorgini, J. Boronat, and J. Casulleras. Ground state of a homogenous Bose gas: A diffusion Monte Carlo calculation. *Phys. Rev. A*, 60:5129–5132, 1999.

`gnuplot`. URL `http://www.gnuplot.info/`. Plotting package.

A. Griffin. A brief history of our understanding of BEC: From Bose to Beliaev. In Inguscio et al. (1998), pages 1–13.

C. J. B. Haga. Project home page, 2006. URL `http://folk.uio.no/cjhaga/master/`. The present thesis in PDF format, code and animations.

E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration, Structure-Preserving Algorithms for Ordinary Differential Equations.* Springer, 2000.

D. M. Harber, J. M. Obrect, J. M. McGuirk, and E. A. Cornell. Measurement of the Casimir-Polder force through center-of-mass oscillations of a Bose-Einstein condensate. *Phys. Rev. A*, 72(3):033610(6), 2005.

N. M. Hugenholtz and D. Pines. Ground-state energy and excitation spectrum of a system of interacting bosons. *Phys. Rev.*, 116(3):489–506, 1959.

M. Inguscio, S. Stringari, and C. E. Wieman, editors. *Proceedings of the International School of Physics "Enrico Fermi", Course CXL*, 1998. Italian Physical Society, IOS Press, Amsterdam.

P. M. Ketcham, D. L. Feder, C. W. Clark, S. G. Satterfield, T. J. Griffin, W. L. George, B. I. Schneider, and W. P. Reinhardt. Volume visualization of Bose-Einstein condensates. *NASA STI/Recon Technical Report N*, 3:448–458, 2001.

W. Ketterle. The atom laser, 1997. URL `http://cua.mit.edu/ketterle_group/Projects_1997/atomlaser_97/atomlaser_comm.html`.

W. Ketterle. The magic of matter waves, 2001. URL `http://cua.mit.edu/ketterle_group/Projects_2001/Vortex_lattice/Vortex.htm`.

S. Kvaal. *Solving eigenvalue problems with Diffpack and ARPACK++.* 2003. URL `http://folk.uio.no/simenkva/hovedfag.shtml`.

S. Kvaal. A critical study of the finite difference and finite element methods for the time dependent Schrödinger equation. Master's thesis, Department of Physics, University of Oslo, 2004.

L. D. Landau and E. M. Lifshitz. *Quantum Mechanics, Non-relativistic Theory.* Pergamon Press, 1958.

L. D. Landau and E. M. Lifshitz. *Statistical Physics.* Pergamon Press, 2nd edition, 1969.

H. P. Langtangen. *Computational Partial Differential Equations: Numerical Methods and Diffpack Programming.* Springer, 2nd edition, 2003.

H. P. Langtangen and A. Tveito, editors. *Advanced Topics in Computational Partial Differential Equations.* Springer, 2003.

A. E. Leanhardt, T. A. Pasquini, M. Saba, A. Schirotzek, Y. Shin, D. Kielpinski, D. E. Pritchard, and W. Ketterle. Cooling Bose-Einstein condensates below 500 picokelvin. *Science*, 301:1513–1515, 2003.

T. D. Lee, K. Huang, and C. N. Yang. Eigenvalues and eigenfunctions of a Bose system of hard spheres and its low-temperature properties. *Phys. Rev.*, 106 (6):1135–1145, 1957.

E. H. Lieb, R. Seiringer, and J. Yngvason. Bosons in a trap: A rigorous derivation of the Gross-Pitaevskii energy functional. *Phys. Rev. A*, 61:043602(13), 2000.

K.-A. Mardal, G. W. Zumbusch, and H. P. Langtangen. Software tools for multigrid methods. In Langtangen and Tveito (2003), chapter 3.

J.-P. Martikainen. Degenerate quantum gases (lecture notes), 2006. URL `http://theory.physics.helsinki.fi/~quantumgas/Lecture4.pdf`.

M. Möttönen, T. Mizushima, T. Isoshima, M. M. Salomaa, and K. Machida. Splitting of a doubly quantized vortex through intertwining in Bose-Einstein condensates. *Phys. Rev. A*, 68(2):23611, 2003.

H. Møll Nilsen and E. Lundh. Stability of double quantized vortices in Bose-Einstein condensates. URL `http://www.cma.uio.no/reports/preprints/`. Preprint., 2006.

J. K. Nilsen, J. Mur-Petit, M. Guilleumas, M. Hjorth-Jensen, and A. Polls. Vortices in atomic Bose-Einstein condensates in the large-gas-parameter region. *Phys. Rev. A*, 71(5):053610(7), 2005.

`opendx`. URL `http://www.opendx.org/`. Visualisation package from IBM.

C. J. Pethick and H. Smith. *Bose-Einstein Condensation in Dilute Gases.* Cambridge University Press, 2001.

W. D. Phillips. Laser cooling and trapping of neutral atoms. *Rev. Mod. Phys*, 70:721–741, 1998.

W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 2nd edition, 1992. URL `http://www.nr.com/`. Beware of licence terms.

H. Pu, C. K. Law, J. H. Eberly, and N. P. Bigelow. Coherent disintegration and stability of vortices in trapped Bose condensates. *Phys. Rev. A*, 59(2): 1533–1537, 1999.

L. Ramdas Ram-Mohan. *Finite Element and Boundary Element Applications in Quantum Mechanics.* Oxford University Press, 2002.

The Royal Swedish Academy of Sciences. Press release: The 1997 Nobel prize in physics, 1997. URL `http://nobelprize.org/physics/laureates/1997/press.html`.

The Royal Swedish Academy of Sciences. Advanced information on the Nobel prize in physics 2001, 2001. URL `http://nobelprize.org/physics/laureates/2001/phyadv.pdf`.

D. V. Schroeder. *An Introduction to Thermal Physics.* Addison Wesley Longman, 2000.

Y. Shin, M. Saba, M. Vengalattore, T. A. Pasquini, C. Sanner, A. E. Leanhardt, M. Prentiss, D. E. Pritchard, and W. Ketterle. Dynamical instability of a doubly quantized vortex in a Bose-Einstein condensate. *Phys. Rev. Lett.*, 93 (16):160406(4), 2004.

Simula Research Laboratory. URL `http://simula.no/`.

G. Strang and G. J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, 1973.

C. H. Townes. The first laser. In L. Garwin and T. Lincoln, editors, *A Century of Nature: Twenty-One Discoveries that Changed Science and the World*, pages 107–112. The University of Chicago Press, 2003.

E. R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative.* Graphics Press, Cheshire, CT, 1997.

Y.-J. Wang, D. Z. Anderson, V. M. Bright, E. A. Cornell, Q. Diot, T. Kishimoto, M. Prentiss, R. A. Saravanan, S. R. Segal, and S. Wu. Atom Michelson interferometer on a chip using a Bose-Einstein condensate. *Phys. Rev. Lett.*, 94(9):90405, 2005.

S. Wildermuth, S. Hofferberth, I. Lesanovsky, E. Haller, L. M. Andersson, S. Groth, I. Bar-Joseph, P. Krüger, and J. Schmiedmayer. Microscopic magnetic-field imaging. *Nature*, 435:440, 2005.

T. T. Wu. Ground state of a Bose system of hard spheres. *Phys. Rev.*, 115(6): 1390–1404, 1959.