
STATE-DEPENDENT FORCES IN COLD QUANTUM GASES

Christopher Billington

Submitted in total fulfilment of the requirements
of the degree of Doctor of Philosophy

Supervisory committee:

Prof Kristian Helmerson
Dr Lincoln Turner
Dr Russell Anderson



School of Physics and Astronomy
Monash University

August, 2016

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

This page intentionally left blank

Contents

Contents	i
1 Introduction	3
1.0.1 Motivation: Turbulence	5
1.0.2 Experiment control and analysis	6
2 Atomic physics: Experimental techniques and theory	9
2.0.1 Experimental techniques	9
3 Quantum mechanics on a computer	15
3.1 From the abstract to the concrete: neglect, discretisation and representation	15
3.2 Solution to the Schrödinger equation by direct exponentiation	16
3.2.1 Matrix exponentiation by diagonalisation	17
3.2.2 Time-ordered exponentials and time-ordered products	18
3.3 Continuous degrees of freedom	20
3.3.1 Spatial discretisation	21
3.4 Discrete degrees of freedom	24
3.4.1 The interaction picture	24
3.4.2 Unitary integration	24
3.5 Continuous degrees of freedom	24
3.5.1 Finite differences	25
3.5.2 The Fourier basis	25
3.5.3 Harmonic oscillator basis functions	25
3.6 Finding ground states	25
3.6.1 Imaginary time evolution	25
3.6.2 Successive over-relaxation	25
3.6.3 Generalisation to excited states via Gram–Schmidt orthonormalisation	25
3.7 The finite-element discrete variable representation	26
3.8 Fourth order Runge–Kutta in an instantaneous local interaction picture	26
3.8.1 Algorithm	28
3.8.2 Domain of improvement over other methods	29
3.8.3 Results	30
3.8.4 Discussion	34
4 Development of a cold atom physics experiment	37

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

4.0.1	Vacuum system	37
4.0.2	Cooling and trapping of atoms	40
4.0.3	Transport of atoms	40
5	Software for experiment control and analysis	41
5.1	Introduction	41
5.2	An overview of the labscript suite	42
5.3	Pseudoclock	44
5.4	The Labscript Library	45
5.5	Setting parameters—runmanager	48
5.6	experiment execution—BLACS	48
5.7	Image acquisition—BIAS	50
5.8	Analysis—LYSE	51
5.9	Optimization—MISE	51
5.10	Portability and Extensibility	53
6	Particle velocimetry of vortices in Bose–Einstein condensates	55
6.0.1	Method	55
6.1	Simulations of Sisyphus cooling	57
6.1.1	Description of cooling scheme	57
6.1.2	Methods	58
6.1.3	Results	60
6.1.4	Vortex-assisted Sisyphus cooling	61
7	Wave mixing in Bose–Einstein condensates	65
7.1	Off-resonant four wave mixing	65
7.2	Spin wave mixing	65
8	Hidden variables for semiclassical models with state-dependent forces	67
8.1	Approximate Markovian decoherence rate for separating wavepackets	67
8.2	Choice of wavepacket size	71
References		73

This page intentionally left blank

CHAPTER I

Introduction

TODO:

- Modify intro to fit ultimate fate of project
- Move Turbulence discussion to section on vortex tracking
- summarise forthcoming chapters, specifically pointing out which sections are new results and which are not, and which results are numerics/theory and which are experiment
- Ensure historical overview does not intrude on theory behind techniques, they should be in the atomic physics section.
- discuss how BECs are quantum but light is modelled classically - Monte Carlo wavefunction methods
- But MOTS have classical positions.
- Talk about semiclassical models and allude to my improvement on them with hidden variables.
- Discuss how rich the field currently is with applications to precision measurement, quantum information and quantum simulation. Advances are being made in theory, experiment and numerics, in particular with new technology and techniques allowing the latter two to become much more powerful than in the past.

BOSE-EINSTEIN CONDENSATES (BECs) in dilute atomic gases are superfluids that can be created in the lab at extremely low temperatures. This strange state of matter was predicted in 1925 by Bose and Einstein [1, 2], first produced experimentally in 1995 [3] in a cloud of rubidium atoms, and has since been made out of many other atoms, usually alkali metals [4–7]. In a BEC, a macroscopic sample of bosonic atoms all occupy the same quantum state, and many of the features of the single particle wavefunctions are exhibited by the cloud as a whole.

As superfluids, BECs have zero viscosity and as such can support persistent flows. In classical fluid dynamics the absence of viscosity means that a fluid cannot support vorticity,¹ and must be irrotational. However, fluid circulation can still occur around points of zero fluid density, known as vortices. In BECs this circulation is also quantised, in units of \hbar/m .

These quantised vortices are topological defects—the phase of the macroscopic wavefunction winds by a multiple of 2π around them, and is undefined at the center of the vortex core itself. Quantised vortices were observed in superfluid helium² in the early

¹This is because the motion of vorticity is described by a diffusion equation—with viscosity as the diffusion constant. When the diffusion constant is zero, there is no way for vorticity to enter the fluid from a boundary in the first place!

²In which 10% or so of the atoms undergo Bose-Einstein condensation.

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

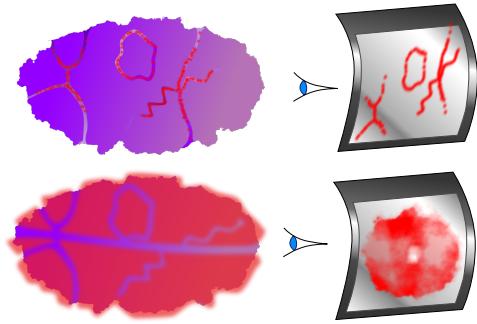


Figure 1.1: Fluorescence imaging of the condensate itself (bottom) makes it difficult to resolve vortices unless they are viewed end-on. The vortex cores are usually smaller than the imaging light's wavelength, and are thus also difficult to resolve unless the cloud is allowed to expand. Imaging tracer particles instead resolves both these problems.

1960s [8], and in BEC in a dilute atomic gas in 1999 [9]. The formation, dynamics and decay of these vortices are believed to be important for the study of superfluid turbulence [10].

This project aims to experimentally realise an imaging method for the real time tracking of quantum vortices in a turbulent ^{41}K condensate. This will involve ultracold ^{87}Rb tracer particles which will become bound to vortex lines in the condensate, and which will be imaged repeatedly to track the vortex lines as they move. The imaging of tracer particles to track vortex motion has already proved successful in superfluid helium [11–13], and the method of laser cooling and imaging atoms in high resolution with the same laser light has also been successful in cold atom systems [14]. We have tested our method *in-silico* [15] and from this, expect it to work under a number of assumptions.

If successful, this method will overcome several existing difficulties that other imaging methods face. Since vortices have previously been imaged by resonant absorption imaging of the condensate itself, they are usually viewed with the vortex line perpendicular to the image plane. If not viewed end on, the rest of the cloud obscures the low density of atoms due to the core. One solution to this problem is to slice the condensate into layers, and image them separately [16].

Our method should be able to image vortex lines side-on without destroying the condensate, since the atoms being imaged reside in the vortex core itself rather than the bulk of the condensate. This should make it possible to image the time evolution of Kelvin waves [17], vortex reconnections [18], and vortex rings [16].

This *in-situ* imaging of vortex dynamics will allow more types of vortex motion to be imaged. Dynamics of BECs are usually imaged with a shot-by-shot method, in which repeated experiments with identical initial conditions are imaged destructively after being allowed to evolve for different amounts of time. Whilst this works for many types of dynamics, it fails for experiments that are more sensitive to initial conditions and noise (quantum or otherwise), such as turbulent flow. This includes phenomena which cannot be created reliably in the same initial state, even though the evolution thereafter would be consistent from one experimental run to the next. One such phenomenon is the spontaneous generation of vortices after evaporative cooling [19].

In-situ imaging of vortex motion has been achieved previously [20], by ejecting a fraction of the atoms from the condensate periodically and imaging them. This process is limited by depletion of the condensate, and was also used only to image vortices end-on. The fraction of the condensate being imaged was also allowed to freely expand before being imaged, since vortex cores are otherwise unable to be resolved by the wavelength of

light used. Our method requires neither free expansion or depletion of the condensate.

1.0.1 Motivation: Turbulence

It is commonly said that turbulence is one of the greatest unsolved problems of classical physics. But in what sense is it an unsolved problem? Its not a problem at all if your aim is reductionism—the Navier–Stokes equation perfectly describes the evolution of a Newtonian fluid within its domain of validity, and the process of deriving it from the underlying motion of classical particles is completely understood. It's turtles all the way down [21, p 1]; what more could we ask for?

The best comparison to make at this point, I think, is to the field of thermodynamics, for precisely the same statement can be made about the energy content and exchange between systems of particles. Thermodynamics has revealed that despite the chaotic motion of individual particles in an ensemble, definite statements can still be made about the behaviour of the system as a whole, *without having to consider the dynamics of the constituent components in detail*.

This is the kind of solution people have in mind when they speak of ‘solving’ the problem of turbulence. Laws describing the average properties of a fluid without reference to its precise flow field would not simply be interesting as describing turbulence as an emergent phenomenon, but would aid practical computations immensely, which are presently quite difficult. The flow of a turbulent fluid contains detail on such a wide range of length scales that any finite-element analysis of a system such as say, an aeroplane wing, requires a very high resolution in order to be accurate. Following an estimate of computing power required to simulate a turbulent system down to its smallest length scales, Stanley Corrsin quipped [22]:

The foregoing estimate is enough to suggest the use of analog instead of digital computation; in particular, how about an analog consisting of a tank of water?

But are we asking for too much? Perhaps the statistical properties of a turbulent fluid fundamentally cannot be decoupled from the finer details. If so, then it is wishful thinking to hope that we might do so.

However there is reason to believe that this is not the case. There are several tantalising results that hint at universal properties that all turbulent flows share, and there is the simple empirical observation that the average flow of turbulent fluids at large scales is reproducible from one experimental run to the next [23, pp 13, 86].

One of these universal results is Kolmogorov’s theory of the statistics of small eddies [24, 25]. Another is the fact that the rate of energy dissipation via the action of viscosity at small scales is *independent of the viscosity itself* [23, p 77].

Then there is the Richardson energy cascade [26], in which energy is continually transferred from larger scales to smaller scales. With dissipation at the smallest scales and addition at larger scales, this allows for the existence of ‘steady state’ turbulence.

So far I haven’t mentioned superfluids at all, though a superfluid is what this project is studying. There are several interesting aspects of superfluid turbulence that differ from classical turbulence. The most obvious is the absence of viscosity; another major difference is the quantisation of vortex lines. On scales much larger than the vortex spacing, superfluid turbulence is expected to closely resemble classical turbulence³ [27]. But at smaller scales the energy dissipation mechanism is different, instead involving the production of sound waves via vortex interactions [27, 28].

In certain 2D geometries, an *inverse cascade* [29, 30] is predicted to take place in superfluids, whereby energy moves not from large scales to small, but from small to large, clustering quantised vortices of the same circulation direction together. This has not as of yet been observed.

³At large scales in classical fluids, velocity gradients are small and hence viscosity can be neglected anyway.

To emphasise the role of vortices in turbulence in general, I will finally give a definition of turbulence, taken from [23, p 53]:

Incompressible hydrodynamic turbulence is a spatially complex distribution of vorticity which advects itself in a chaotic manner in accordance with [the vorticity equation⁴]. The vorticity field is random in both space and time, and exhibits a wide and continuous distribution of length and time scales.

⁴Which is a transformation of the Navier–Stokes equation for an incompressible fluid into a form in which the vorticity field is center-stage.

When vorticity exists only in infinitely narrow lines, as it does in superfluid, the vorticity equation mentioned in the above definition reduces to a Biot–Savart type law which can be used to compute the motion of vortices without having to compute the entire flow field.

This is why we are interested in the study of the dynamics of quantised vortices. Unlike in classical fluids, the vortices in superfluids have a definite position and size; there is either a vortex or there is not. This may make it simpler to describe the motion of vortices statistically.

So far experimental studies of superfluid turbulence have mainly been in the context of liquid helium [31]; we hope to augment the existing experimental data with that obtained from BEC. The high degree of control afforded over systems of cold atoms allows the superfluid's properties to be tweaked in several ways, creating a larger parameter space in which to study turbulence than that afforded by liquid helium.

1.0.2 Experiment control and analysis

As is typical of an experimental project, much of what goes on day to day is implementation details. It's all very well to say that we want to make a turbulent condensate by pulsing some laser speckle for a few milliseconds, or that we want to ramp down a magnetic field at a certain rate, but how we achieve this in practice? First we have to set up the required equipment—a vacuum system, lasers, optics, magnetic coils, RF antennae. Progress on that front is detailed in section ??.

Once the experiment is set up, we need to be able to control how the system's state changes over time. Components of the system that need to change state during an experiment are typically controlled with analogue and digital electronic signals. Shutters are opened and closed with digital edges; acousto-optical modulators shift the frequency of light passing through them according to a driving AC signal; and magnetic coils create field profiles depending on their current, which in turn depends on the voltage driving their control box. Once we have hardware capable of producing the required digital pulses, voltage ramps, and AC radio frequency signals, running an experiment comes down to programming them and having them execute their programs.

To this end we have created what is essentially a compiler, called `labscript`, which takes high level instructions for what the devices should do, and generates the required low-level instructions for all the devices in an experiment, including clocking signals to keep all devices in sync. This is discussed in section ??.

With an experimental run compiled, it needs to be programmed into, and executed on the hardware. This is performed by a program called `BLACS`, which also provides real-time control of the hardware when not executing pre-compiled experiments.

Experimental runs can take input parameters. For adjusting which values are to be used, and for repeating the same experiment with a range of different parameters, we have a program called `runmanager`. `runmanager` is a graphical front-end for setting these parameters and creating multi-dimensional parameter-space scans.

Two aptly named programs, `lyse` and `BIAS`, perform analysis of the acquired data as experiments occur. `BIAS` interacts with `BLACS` in order to program the camera(s) before a run, and it performs image analysis once the run is complete. `lyse` is a general purpose

```
rev:    25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date:   Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

analysis system which triggers the execution of user analysis scripts whenever there is new data, computing any results and showing any plots that those scripts contain.

The software components of our control and analysis system are described in more detail in section ??.

```
rev:    25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date:   Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

This page intentionally left blank

CHAPTER 2

Atomic physics: Experimental techniques and theory

- Descriptions of the relevant physics in atomic physics experiments: Doppler cooling and magneto-optical traps, Sisyphus cooling, dipole forces, Feshbach resonances, scattering theory, Bose–Einstein statistics. Show how the Hamiltonian of a ‘two level’ (32 levels, all things considered for ^{87}Rb D line) atom with fine structure, hyperfine structure and Zeeman splitting arises from consideration of the different angular momenta. Use this to derive the differential equations for the state populations of an atom in a driving laser field. Gross Pitaevskii equation for single species, dual species and spinor condensate.
- Doppler cooling
- Magneto-optical trapping
- Optical dipole trapping
- Two-body scattering and Feshbach resonances
- -> Include stuff from 3rd year report
- Spin, fine structure, and hyperfine structure
- Equations of motion for two level atom with hyperfine structure
- The Monte-Carlo wavefunction method
- Mean field theory for Bose–Einstein condensates
- -> Superfluid velocity
- -> Vortices

2.0.1 Experimental techniques

BECs provide such a tantalising opportunity for studying quantum phenomena not only because of their interesting properties, but also because of the level of control they afford. Many of the same techniques which allow experimentalists such control over their creation are also employed in the creation thereof, and many were discovered along the way to Bose–Einstein condensation.

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

The main experimental techniques used to create BEC—and which we are and will be employing in that pursuit—are Doppler cooling, magneto-optical and dipole trapping, polarisation gradient (Sisyphus) cooling, and evaporative cooling.

These were discovered, perhaps by no coincidence, in roughly the same order as they are called for in a BEC experiment.

Doppler cooling

Doppler cooling, demonstrated in 1978 [32] is a consequence of the simple observation that atoms see the wavelength of incident light Doppler shifted depending on their velocity. This can be used to selectively transfer momentum to only fast-moving atoms, by tuning an incident laser slightly redder than would be required for a resonant absorption. If six lasers in counterpropagating pairs orthogonal to each other surround a cloud of atoms, the atoms can be cooled close to the *Doppler limit* [33, p 58]

$$k_B T_D = \frac{\hbar \Gamma}{2} \quad (2.1)$$

¹The D₂ line, 5S_{1/2} → 5P_{3/2}, approximately 780 nm.

where Γ is the linewidth of the atomic transition. For the cooling transition used for Doppler cooling ⁸⁷Rb¹, this gives 146 μK, which is approximately a factor of a thousand too high for Bose-condensation. These atoms are also not trapped.

Magneto-optical and magnetic trapping

Magneto-optical trapping, first demonstrated in 1987 [34] comes from the realisation that a magnetic field can be used to *spatially* vary the detuning from resonance that the atoms in the above mentioned arrangement of lasers see. This is possible due to the Zeeman effect [35], in which the wavelengths of atomic transitions are shifted in a magnetic field.

If a field profile can be found which causes the transition to come closer to resonance as the atoms move away from a central point, then it forms a trap—atoms that stray too far from the center will absorb more strongly and be deflected back².

The field configuration used in an anti-Helmholtz one, with two coils opposite each other carrying opposing currents. The resulting magnetic field profile has a zero in the middle and increases in magnitude in all directions.

With the Doppler beams off, this magnetic field still provides a trapping potential, due to the magnetic dipole interaction:

$$V(\mathbf{r}) = -\boldsymbol{\mu} \cdot \mathbf{B}, \quad (2.2)$$

where $\boldsymbol{\mu}$ is the atomic magnetic moment, and \mathbf{B} the magnetic field. This only traps some atomic spin states, and has losses due to spin-flips [36] near the field zero.

Optical trapping

Optical dipole trapping on the other hand relies on the *dipole force*, in which off-resonant light shifts the energy of the eigenstates of the combined atom-light system, the so called *dressed states*. This energy shift, called the *light shift*, depends on the intensity of the light, and so results in a potential that spatially varies as the intensity of the light. In the limit of large detuning (compared to Rabi frequency), this shift is given by [33, p 8]:

$$\Delta E = \frac{\hbar \Omega^2}{4\delta} \quad (2.3)$$

where δ is the detuning from resonance and the Rabi frequency is:

$$\Omega = \frac{eE_0}{\hbar} \langle 1 | \mathbf{x} | 2 \rangle, \quad (2.4)$$

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

where E_0 is the amplitude of the light's electric field and $\langle 1|x|2 \rangle$ is the dipole moment between the two states in a two-level system.

With the potential proportional to E_0^2 , and thus the light's intensity, the force the atom experiences is proportional to the light's intensity gradient. For this reason, the dipole force is also called the *gradient force*. The name *dipole force* comes from the fact that the force can be equivalently understood to arise from the polarisability of atoms in a light field, giving rise to a force identical to that which traps polarisable materials in optical tweezers [37].

Polarisation gradient cooling

Polarisation gradient cooling, also called Sisyphus cooling, was proposed in 1989 [38,39] to explain experimentally measured cold atom cloud temperatures [40] which, at NIST in 1988, were found to be well below the expected limit obtainable by the well understood method of Doppler cooling³, one of the few examples of experiments turning out better than expected. A one dimensional theory has been developed [38] which has found remarkable agreement with three dimensional experiments [41]

One common configuration for Sisyphus cooling comprises two counterpropagating laser beams in each spatial dimension, both linearly polarised but with their polarisation angles perpendicular to one another. The optical field resulting from the two beams' superposition has regions of linear polarisation and of both helicities of circular polarisation, and varies between them on a length scale shorter than an optical wavelength.

The effect on multi-level atoms as they move from regions of one circular polarisation to another is that they are pumped alternately from one extreme of their spin-projection states to the other, alternately climbing and descending potential hills due to the dipole forces from the regions of different polarisations⁴. And so, like the Greek legend of Sisyphus⁵, who was doomed to push a rock uphill for eternity, the atoms are climbing hills repeatedly. Due to the state dependence of the strength of the dipole forces, the atoms climb steeper hills than they descend, and are thus slowed and cooled.

This type of cooling does not work in a magnetic field; the splitting of transition frequencies makes it impossible for an atom to traverse its spin manifold on one laser frequency. For this reason the Sisyphus cooling stage is performed with magnetic fields off, though a sufficiently short period is required such that the atoms can be recaptured when the trapping field is restored.

Evaporative cooling

The final stage of cooling is forced RF evaporative cooling [3,42], which decreases the temperature of the cloud by systematically removing the hottest atoms. This is performed in a magnetic trap, which as mentioned earlier, only traps certain spin states. Evaporation proceeds by using an *RF knife* to induce spin flips in the atoms. The RF frequency is chosen such that it is only resonant with atoms some distance away from the center of the trap (via the Zeeman shift). The furthest out atoms are the most energetic, possessing the energy to climb the magnetic potential the furthest. By flipping their spins, these atoms are ejected due to the magnetic field becoming anti-trapping for them.

The cloud is given some time to rethermalise and the knife⁶ is moved inward where it removes slightly colder atoms. This is repeated until the desired compromise of lower temperature/lower atom number is reached. Usually some method is employed to prevent atoms near the center of the trap from undergoing spin flips [36] as they move across the field zero. The method we'll employ is to use an optical dipole trap in combination with the magnetic trap [43], such that the coldest atoms get trapped in the dipole trap which is offset from the magnetic field zero.

³As well as to explain other discrepancies between experiments and the theory of Doppler cooling, such as the optimal detuning of light being much greater than predicted.

⁴If you consider only one polarisation of light, its intensity varies sinusoidally in space, creating a series of potential hills and wells via the dipole force

⁵Polarisation gradient cooling is but one of a family of so called 'Sisyphus cooling' methods, all of which involve atoms repeatedly climbing potential hills.

⁶So called because it cuts the tail off the velocity distribution of the atom cloud.

The Gross-Pitaevskii equation and Vortices

Bose-condensates are described well by *mean field* theory, whereby the many-body wavefunction is approximated by a product of identical single-particle wavefunctions. Indeed, that the majority of the atoms are in the same quantum state is one of the defining features of BEC. The effect of interparticle interactions is included as a nonlinear term in the Schrödinger equation for the single particle wavefunctions, known as the Gross-Pitaevskii equation:

$$\frac{\partial \Psi}{\partial t} = \left[-\frac{\hbar^2}{2m} \nabla^2 + V(\mathbf{x}) + g|\Psi|^2 \right] \Psi, \quad (2.5)$$

⁷And is usually positive—having the effect of stabilising BECs by self-repulsion.

⁸Thus giving it the property that $|\Psi|^2$ is the particle density.

where g characterises the strength of the interparticle interactions⁷, and $\Psi = \sqrt{N}\Psi_{\text{single}}$ is the single-particle wavefunction scaled by the square root of the number of particles⁸.

In the hydrodynamic formulation of quantum mechanics [44], the flow velocity of a spatial wavefunction can be defined by considering the probability current to be a product of density and velocity. This allows us to define the superfluid velocity of a BEC as:

$$\mathbf{v} = \frac{\hbar}{m} \nabla \phi \quad (2.6)$$

where ϕ is the phase of the condensate wavefunction Ψ . Integrating this velocity over any closed path γ gives us the circulation:

$$C = \frac{\hbar}{m} \oint_{\gamma} \nabla \phi \cdot d\mathbf{s} \quad (2.7)$$

$$= \frac{\hbar}{m} 2\pi n. \quad n = 0, 1, 2 \dots \quad (2.8)$$

The fact that the circulation is quantised means that vorticity cannot exist in the condensate except in one-dimensional lines, about which the wavefunction's phase winds by a multiple of 2π . These topological defects are the quantised vortices that are central to this project.

At a vortex core, the atom density of a BEC must go to zero. This can be intuitively understood to arise from centrifugal forces, but is also required in order for the wavefunction to be continuous and single-valued across the core. This drop in density in the vicinity of a vortex core is what our method exploits in order to trap atoms within the cores.

Feshbach resonances

⁹Feshbach resonances can also be induced optically and with RF but magnetic resonances are the most commonly used.

The interparticle interaction mentioned above:

$$g = \frac{2\pi\hbar^2 a}{m_r} \quad (2.9)$$

where m_r is the reduced mass of a pair of the interacting particles, is dependent on a parameter a called the *s-wave scattering length*, which characterises low energy collisions between atoms. It is sensitive not only to what species of atoms are colliding, but also to their spin states. For each combination of spins, there is a different inter-atomic potential (called a *channel*) which determines the collision dynamics (Figure 2.1).

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

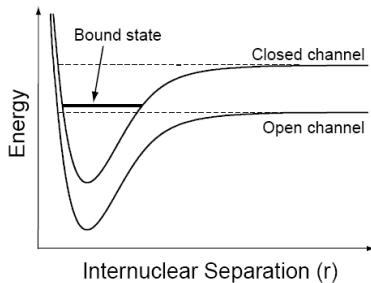


Figure 2.1: When atoms approach each other with spins aligned, they are in the *open channel*. In this channel they are unbound, but do not have enough energy to be free in the other channel - the *closed channel*. In the close range however, the atoms may have energy corresponding to a bound (molecular) state of the closed channel, a resonance which causes a divergence in the scattering length. The energy difference between the two channels can be tuned with a magnetic field and so these resonances can be induced in a wide range of situations.

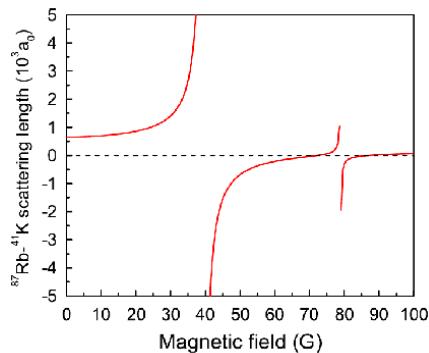


Figure 2.2: Predicted interspecies scattering length [47] as a function of magnetic field strength, for ^{41}K and ^{87}Rb both in their lowest energy hyperfine groundstate. The 35 gauss resonance is one of the main reasons for this pair of atoms being used in this project. It has a particularly low field strength and large width compared to most Feshbach resonances.

The resulting scattering length is sensitive to any bound states of this inter-atomic potential which are near the collision energy. If the channels of different spin states are coupled via the hyperfine interaction¹⁰, then the scattering length is also sensitive to bound states in the channels other than the one the atoms are in when they are far from each other. Due to the Zeeman effect, the energies between the different channels can be shifted with a magnetic field, and so a bound state can be shifted close to the collision energy, which causes the scattering length to diverge.

The end result is that at certain magnetic field strengths we find that atoms are much more strongly attracted to or repelled from each other.

We plan to use a Feshbach resonance (Figure 2.2) to enhance the interspecies repulsion between ^{87}Rb and ^{41}K , thus trapping tracer particles more strongly in vortex cores.

¹⁰Requiring that the atoms in question have a nuclear magnetic moment.

This page intentionally left blank

Quantum mechanics on a computer

This chapter comprises a summary of some of the methods used by cold atom physicists to compute numerical results pertaining to cold atom systems. Many a problem in quantum mechanics is not analytically solvable, especially when the real world of experimental physics rears its ugly head, violating theorists' assumptions of simplicity left and right. In particular, atomic physics experiments are time-dependent, with each run of an experiment generally proceeding in stages. Lasers may turn on and off, magnetic fields may vary in magnitude and direction, RF pulses may be chirped [CITE LISA?] to reliably induce particular atomic transitions. Much of the numerical calculations performed by researchers in cold atom physics groups such as ours are accordingly of the time-dependent variety, and are fairly literal simulations of specific experiments that may be carried out in the lab.

3.1 From the abstract to the concrete: neglect, discretisation and representation

To numerically simulate a quantum mechanical system, one must evolve a state vector in time according to the Schrödinger equation:

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = \hat{H}(t) |\psi(t)\rangle \quad (3.1)$$

To do this on a computer, one must first decide which degrees of freedom are to be simulated. We necessarily neglect many degrees of freedom as a matter of course; which ones can be neglected is warranted by the specific situation and we do it so often we barely notice. For example, simulating a single component Bose–Einstein condensate entails neglecting the internal degrees of freedom of the atoms—as well as reducing the atom-light interaction to a simple potential such as an optical dipole trap or magnetic dipole interaction (neglecting the quantum degrees of freedom in the electromagnetic field). We may ignore one or more spatial degrees of freedom as well, say, if we are simulating an experiment in which the condensate is confined to two dimensions by way of a tight trapping potential in one direction [CITE AN EXAMPLE]. Or, when simulating laser cooling [SEE SECTION TODO], we may care very much about the electronic state of the atom, but treat its motional state classically. In these cases we are essentially imposing the assumption that the system will only occupy one state with respect to those degrees of freedom ignored (the condensate will remain in lowest excitation level in the direction of the tight trap, the atoms will remain in one specific Zeeman sublevel), or we are assuming those degrees of freedom can be treated classically (the electromagnetic field is well described by classical electromagnetism, the atoms' motional state is described

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

¹A classic example in the cold atom community of neglected degrees of freedom leading to *disagreement* with experiment is the discovery of polarisation gradient cooling (PGC) [CITE THE LIGHT SHEET EXPERIMENT], the explanation for which requires consideration of Zeeman sublevels of the atoms. The experiment that discovered PGC was designed to measure the effect of Doppler cooling, which does not involve Zeeman sublevels, and it was not until afterwards that theorists determined [CITE DALIBARD ETC] that transitions between Zeeman sublevels cannot be neglected and indeed are crucial in explaining the lower than predicted temperatures observed.

²The D-line of rubidium 87 has an energy gap of [TODO], requiring a temperature of [TODO] or higher in order for the Boltzmann factor $e^{\frac{\Delta E}{kT}}$ describing the excited state population to be greater than 1×10^{-6} .

³Strictly speaking, these can be the same thing—discretising space on a grid is declaring one’s basis functions to be a set of functions, one for each gridpoint, each of which is zero at all gridpoints.

well by Newtonian mechanics). Which degrees of freedom can be neglected and which cannot requires knowledge of the situation at hand, often informed by best-practices of the research community in question and ultimately justified by experiment.¹

Once the degrees of freedom are known, one must decide on a basis in which to represent them concretely. The basis often cannot be complete, since for many degrees of freedom this would require an infinite number of basis states—for example the electronic state of an atom contains a countably infinite number of states, and a spatial wavefunction in free space has an uncountable number of states (one for each position in \mathbb{R}^3 [TODO BLACKBOARD BOLD]). For the internal state of an atom, therefore, we restrict ourselves to only the states we expect can become non-negligibly occupied, given the initial conditions and transitions involved. For example, at low temperature we can expect atoms to be almost completely in their electronic ground states, since energy gaps between ground and excited states are large compared to the atoms’ thermal energy.² We need only include the small number of excited states that might become occupied as a result of optical transitions present in the situation being simulated. This can still be a large number of states if one is studying Rydberg atoms [CITE] or using ultrafast (and therefore broad-band) laser pulses, but is otherwise fairly small. For example, including both the D₁ and D₂ lines of Rubidium 87, with all hyperfine levels and Zeeman sublevels gives 32 states (see section [TODO]).

For spatial degrees of freedom, one can either discretise space on a grid, or use a set of orthogonal basis functions³.

Once the degrees of freedom and basis vectors have been chosen, the state vector is then represented on a computer as an array of complex numbers, giving the coefficients of each basis vector required to represent a particular state vector. Matrix elements of the Hamiltonian in the same basis must be calculated, and the Schrödinger equation can then be written:

$$i\hbar \frac{d}{dt} \langle n|\psi(t)\rangle = \sum_m \langle n|\hat{H}(t)|m\rangle \langle m|\psi(t)\rangle, \quad (3.2)$$

or in standard matrix/vector notation (without Dirac notation):

$$i\hbar \frac{d}{dt} \psi_n(t) = \sum_m H_{nm}(t) \psi_m(t), \quad (3.3)$$

where $\psi_n(t) = \langle n|\psi(t)\rangle$ and $H_{nm}(t) = \langle n|\hat{H}(t)|m\rangle$. This is now something very concrete that can be typed into a computer. Programming languages generally don’t know about Dirac kets and operators, and so everything that is to be computed must be translated into matrices and vectors in specific bases. This may seem so obvious as to not be worth mentioning, but was nonetheless a stumbling block in my own experience of getting to grips with quantum mechanics. Once realising that every operator has a matrix representation in some basis, at least in principle, and that every ket is just a list of vector components in some basis, similarly at least in principle, expressions dense in bras and kets become much more concrete as the reader has a feel for exactly how they would type it into a computer. Without this realisation, doing quantum mechanics on paper can seem like an exercise in abstract mumbo-jumbo.

3.2 Solution to the Schrödinger equation by direct exponentiation

As an example of something seemingly abstract being more concrete than first appearances, it is sometimes said that the ‘formal’ solution to the Schrödinger equation (3.1) is:

$$|\psi(t_f)\rangle = e^{-\frac{i}{\hbar} \int_t^{t_f} \hat{H}(t') dt'} |\psi(t)\rangle. \quad (3.4)$$

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

Saying that this is the ‘formal’ solution rather than just ‘the solution’ is presumably intended to emphasise that the arithmetic operations involved in (3.4) might not make immediate sense for the types of mathematical objects they are operating on, and that we have to be careful in defining the operations such that they produce a result that is not only sensible, but also the solution to the Schrödinger equation. If both $\hat{H}(t)$ and $|\psi(t)\rangle$ were single-valued functions of time rather than an operator valued function of time (the values at different times of which don’t necessarily commute) and a vector valued function of time, then we would have no problem. However, (3.4) as written with operators and vectors is ambiguous, and we need to elaborate on it in order to ensure it is correct. I will come back to this after considering a simpler case.

If the Hamiltonian is time-independent, then (3.4) reduces to

$$|\psi(t_f)\rangle = e^{-\frac{i}{\hbar}(t_f-t)\hat{H}} |\psi(t)\rangle. \quad (3.5)$$

Given the matrix representation H of \hat{H} and vector representation $\psi(t)$ of $|\psi(t)\rangle$ in a particular basis, this can now be directly typed into a computer as the matrix multiplication:

$$\psi(t_f) = U(t, t_f) \psi(t), \quad (3.6)$$

where

$$U(t, t_f) = e^{-\frac{i}{\hbar}(t_f-t)H} \quad (3.7)$$

is the unitary evolution operator for time evolution from the initial time t to time t_f , and is computed using a matrix exponential of $-\frac{i}{\hbar}(t_f - t)H$. Exponentiation of matrices is defined via the Taylor series of the exponential,

$$e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!}, \quad (3.8)$$

which reduces exponentiation to the known operations of matrix multiplication and addition. However, any linear algebra programming library worth the bytes it occupies will have a matrix exponentiation function that should be used instead, as there are other methods of computing matrix exponentials that are more computationally efficient and numerically stable, such as the Padé approximant [CITE].

3.2.1 Matrix exponentiation by diagonalisation

Regardless of which method is used, matrix exponentiation is computationally expensive. It can be sped up however if a diagonalisation of H is known, since if

$$H = UDU^\dagger, \quad (3.9)$$

where D is a diagonal matrix and U is a unitary matrix⁴, then

$$e^{-\frac{i}{\hbar}(t_f-t)H} = U e^{-\frac{i}{\hbar}(t_f-t)D} U^\dagger. \quad (3.10)$$

This is simple to evaluate because the exponentiation of a diagonal matrix can be performed by exponentiating each diagonal matrix element individually⁵.

Even if a diagonalisation of H is not analytically known, numerically diagonalising H (using a linear algebra library function or otherwise) can form the basis for writing your own matrix exponentiation function, if needed. I found this necessary for efficiently exponentiating an array of matrices in Python, since the `scipy` and `numpy` scientific and numeric libraries at the present time lack matrix exponentiation functions that can act on arrays. Writing a `for` loop in an interpreted language such as Python to exponentiate the matrices individually in many cases is unacceptably slow, so for these cases⁶ I use a function such as the below:

```
rev:    25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date:   Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

⁴Note that the diagonals of D are the eigenvalues of H , and the columns of U are its eigenvectors.

⁵The reason for this is clear from the Taylor series definition of matrix exponentiation, since matrix multiplication and addition can both be performed elementwise for diagonal matrices.

```

1 import numpy as np
2 from numpy.linalg import eigh
3
4 def expmh(M):
5     """compute exp(M), where M, shape (... , N, N) is an array of N by N
6     Hermitian matrices, using the diagonalisation method. Made this function
7     because scipy's expm can't take an array of matrices as input, it can only
8     do one at a time."""
9
10    # Diagonalise the matrices:
11    evals, evecs = eigh(M)
12
13    # Now we compute exp(M) = U exp(D) U^{\dagger} where U is the matrix of
14    # eigenvectors (as columns) and D is the diagonal matrix of eigenvalues:
15
16    U = evecs
17    U_dagger = U.conj().swapaxes(-1, -2) # Only transpose the matrix dimensions
18    exp_D_diags = np.exp(evals)
19
20    # Compute the 3-term matrix product U*exp_D_diags*U_dagger using the
21    # einsum function in order to specify which array axes of each array to
22    # sum over:
23    return np.einsum('...ik,...k,...kj->...ij', U, exp_D_diags, U_dagger)

```

This concludes all I have to say for the moment on evolving systems with time-independent Hamiltonians using direct exponentiation. Sections [TODO - refer to operator product sections] will say a bit more about how to *approximately* exponentiate certain Hamiltonians, which is sometimes useful. In addition, it turns out that the time-dependent case can be reduced to the repeated application of evolution with time-independent Hamiltonians over small timesteps, as we will see in the next section.

3.2.2 Time-ordered exponentials and time-ordered products

As mentioned, the solution (3.4) is not the whole picture. It can only be taken at face value if the Hamiltonian at each moment in time commutes with itself at all other times [CITE SOMETHING]. In this case, once represented in a specific basis, the solution to the Schrödinger equation is again the matrix multiplication

$$\psi(t_f) = U(t, t_f)\psi(t), \quad (3.11)$$

with

$$U(t, t_f) = e^{-\frac{i}{\hbar} \int_t^{t_f} H(t') dt'}. \quad (3.12)$$

Since matrix addition can be performed elementwise, so can the integral in the exponent, yielding a matrix which once exponentiated will give the evolution operator $U(t, t + \Delta t)$ for the solution to the Schrödinger equation. If the Hamiltonian at each moment in time does not commute with itself at all other times, however, then the unitary evolution operator for the solution to the Schrödinger equation is instead given by the following *time-ordered exponential* [CITE]:

$$U(t, t_f) = T \left\{ e^{-\frac{i}{\hbar} \int_t^{t_f} H(t') dt'} \right\}. \quad (3.13)$$

In this expression, T denotes the *time-ordering operator*. The time ordering operator reorders terms within products that contain a time parameter (for us, the time parameter is the argument of the matrix H), such that the value of the time parameter is smallest on the rightmost term, largest on the leftmost term, and monotonically increasing right-to-left in between. For example:

$$T \{H(4)H(1)H(2)H(5)H(3)\} = H(5)H(4)H(3)H(2)H(1). \quad (3.14)$$

```

rev:      25 (79ce3eb23e1f)
author:  Chris Billington <chrisjbillington@gmail.com>
date:    Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

```

Despite appearances, this time-ordered exponential is perfectly concretely defined via the definitions of all the operations involved that we have described so far, and can—with some effort—be typed into a computer and evaluated directly. Even though this is not how I evaluate time-ordered exponentials in my simulations of atomic systems, I'll quickly elaborate on this just to emphasise the concreteness of all these operations.

“What products is T reordering?” you might ask, as (3.13) doesn't appear to contain any products of $H(t)$. On the contrary, it does, since exponentiation is defined by its Taylor series, and so

$$U(t, t_f) = 1 + T \left\{ \sum_{n=1}^{\infty} \frac{1}{n!} \left[-\frac{i}{\hbar} \int_t^{t_f} H(t') dt' \right]^n \right\} \quad (3.15)$$

$$= 1 + \sum_{n=1}^{\infty} \frac{(-i)^n}{n! \hbar^n} T \left\{ \left[\int_t^{t_f} H(t') dt' \right]^n \right\}. \quad (3.16)$$

Each term in this series contains the n^{th} power (and hence a product) of an integral of $H(t)$. The time ordering operator doesn't allow us to evaluate each term by computing the matrix integral once and then raising it to a power—to do so would violate time-ordering since each integral involves evaluating $H(t)$ at all times. Instead we have to write each product of integrals as the integral of a product:

$$U(t, t_f) = 1 + \sum_{n=1}^{\infty} \frac{(-i)^n}{n! \hbar^n} \int_t^{t_f} \cdots \int_t^{t_f} T \{H(t'_1) \cdots H(t'_n)\} dt'_1 \cdots dt'_n, \quad (3.17)$$

from which we can see exactly which product of matrices the time ordering operator is acting on.

Now we are close to seeing one might evaluate $U(t, t_f)$ numerically by summing each term in the Taylor series up to some order set by the required accuracy. For the n^{th} term, one needs to evaluate an n -dimensional integral over n time coordinates, with each coordinate having the same limits of integration. This can be computed in the usual way an integral is numerically computed⁷, with the minor change that each time the integrand is evaluated, the terms within it must be re-ordered to respect the required time-ordering. Alternatively, the integration region can be restricted to the region in which the terms are already time-ordered, and then the total integral inferred by symmetry, which gives:

$$U(t, t_f) = 1 + \sum_{n=1}^{\infty} \left(-\frac{i}{\hbar} \right)^n \int_t^{t_f} \cdots \int_t^{t_{n-1}'} H(t'_1) \cdots H(t'_n) dt'_1 \cdots dt'_n. \quad (3.18)$$

This is now a perfectly concrete expression, with each term comprising an integral over an n -simplex⁸ of a product of n matrices.

This expression for the unitary evolution operator is called the Dyson series [CITE]. It is not generally used in cold atom physics; I certainly have never evaluated it. However it does see use in high energy physics [CITE] for computing transition amplitudes between incoming and outgoing waves in scattering problems (in which U is called the S -matrix). In these problems, H is an interaction Hamiltonian containing terms for all particle interactions being considered. Accordingly, the integrand for the n^{th} term, being a product of n copies of H at different times, contains one term for each possible sequence of n particle interactions. The integral itself can be considered a sum of transition amplitudes over all possible times that each interaction could have occurred. Indeed, each term corresponds to a Feynman diagram with n nodes [CITE].

The Dyson series isn't really suited to the problems we face in cold atom physics. For one, the series must be truncated at some point, and the result won't be a U that is actually unitary.⁹. Also, we are usually interested in the intermediate states, not just the final state of a system as in a scattering problem.

⁷By sampling the integrand on a uniform grid, using a quadrature method, or Monte-Carlo integration [CITE], which is widely used for high dimensional integrals such as these.

⁸A simplex is the generalisation of a triangle to higher dimensions, i.e a 3-simplex is a tetrahedron.

⁹Although unitarity is not often a strict requirement - we also frequently solve [EQREF TDSE] directly with fourth order Runge-Kutta, which is also not unitary.

In any case, typically when solving the Schrödinger equation by exponentiation we use the following, alternate expression for a time-ordered exponential:

$$U(t, t_f) = T \left\{ e^{-\frac{i}{\hbar} \int_t^{t_f} H(t') dt'} \right\} \quad (3.19)$$

$$= \lim_{N \rightarrow \infty} T \left\{ \prod_{i=0}^N e^{-\frac{i}{\hbar} H(t_i) \Delta t} \right\}, \quad (3.20)$$

where $\Delta t = (t - t_f)/N$ and $t_i = t + i\Delta t$. You can convince yourself that [EQREF INTEGRAL DEFINITION] is equivalent to this by replacing the integral in the exponent with a sum—as per the Riemann definition of an integral—and expanding the exponential according to its Taylor series. Expanding each exponential in [EQREF PRODUCT DEFINITION] as a Taylor Series and collecting terms with the same number of copies of H then reveals that the two Taylor series are identical.

In any case, [EQREF PRODUCT DEFINITION] paints an intuitive picture of solving the Schrödinger equation: one evolves the initial state vector in time by evolving it according to constant Hamiltonians repeatedly over small time intervals. This has the desirable property that all intermediate state vectors are computed at the intermediate steps, meaning one can study the dynamics of the system and not just obtain the final state. This is of course useful for comparison with experiments, plenty of which involve time-dependent data acquisition and not just post-mortem analysis of some evolution.

Numerically, we can't actually take $N \rightarrow \infty$, or equivalently $\Delta t \rightarrow 0$, and so we instead choose a Δt smaller than the timescale of any time-dependence of H , and step through time using [EQREF].

Thus the case of a time-dependent Hamiltonian reduces to repeated application of the solution [EQREF] for a time-independent Hamiltonian. Furthermore, if the Hamiltonian is analytically diagonalisable at each moment in time, each of these steps can be taken with the diagonalisation method of matrix exponentiation. If it is not, and the system is small, then resorting to numerical diagonalisation may be acceptable. Otherwise, if the Hamiltonian can be decomposed into a *sum* of terms which are individually diagonalisable, the split-step method can be used (see sec [SECREF]). Failing that, a general purpose integration method such as fourth-order Runge–Kutta may have to be resorted to.¹⁰ I will elaborate on these comments with examples in the following sections.

¹⁰Though if I admit it, `RKR4` is usually what I reach for first.

- absorbing boundary conditions, reflective boundary conditions, periodic boundary conditions

3.3 Continuous degrees of freedom

The single-particle, non-relativistic, scalar Schrödinger wave equation, as distinct from the general Schrödinger equation [EQREF], is:

[EQUATION].

As mentioned in [SECREF], the equation for the single-particle wavefunction of an atom in a single-component Bose–Einstein condensate is the Gross–Pitaevskii equation [EQUATION],

where $\psi(\mathbf{r}) = \sqrt{N} \langle \mathbf{r} | \psi \rangle$ is the single-particle wavefunction scaled by the square root number of atoms N .

Both these equations are partial differential equations involving both spatial and temporal derivatives. But in quantum mechanics all state vectors can be mapped to column vectors and all operators to matrices. Spatial wavefunctions are no exception to the former and differential operators such as ∇^2 are no exception to the latter. So what do these vectors and operators look like? That depends on whether we choose to

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

discretise space on a grid, or use a functional basis (and on which functional basis we choose). As we'll see below, however, spatial discretisation is actually just a particular choice of functional basis, namely the Fourier basis.

3.3.1 Spatial discretisation

Imagine a two dimensional spatial region within which we are solving the single-component Gross–Pitaevskii equation, evolving an initial condensate wavefunction in time. Having specified which degrees of freedom we want to simulate (two continuous degrees of freedom, one for each spatial dimension), the next step according to the method outlined in [SECREF ABOVE] is to choose a basis in which to represent this state vector.

Lets say we discretise space in an equally-spaced $N_x \times N_y = 5 \times 5$ rectangular grid,¹¹ with spacing Δx , and only represent the wavefunction at those 25 points in space. The state vector can then be represented by a list of 25 complex numbers, each taken to be the wavefunction's value at the spatial position corresponding to one gridpoint. This 25-vector is now a concrete representation of our state vector.

[FIGURE SHOWING UNWRAPPING OF 2D REGION INTO COLUMN VECTOR. PERHAPS 2D REGION IS CONTINUOUS WITH A GRID SUPERIMPOSED ON IT, AND LINES JOIN EACH GRIDPOINT TO THE VECTOR AT RIGHT]

But at what point did we choose a basis just now—what are the basis vectors? This just looks like discretising space at a certain resolution, rather than the formal process of choosing a basis and projecting the state vector and operators onto each basis vector, as outlined in [SECREF ABOVE]. Assuming what we've done is equivalent to choosing a basis, that basis has a finite number (25) of basis vectors, which means it cannot be complete, since state vectors we're approximately representing with it require an infinite number of complex numbers to be described exactly.¹² So what do the basis functions look like, and what state vectors have we implicitly excluded from simulation by choosing a basis that is incomplete?

As a sidenote, spatial wavefunctions are often described as the representation of wavefunctions in the “spatial basis”—a basis in which the basis vectors $\{|\mathbf{r}\rangle\}$ are Dirac-deltas centred on each point in space [CITE DIRAC'S BOOK MAYBE]. The wavefunction $\psi(\mathbf{r})$ is then simply a coefficient saying how much of the basis vector $|\mathbf{r}\rangle$ (the spatial representation of which is a Dirac delta centred on the position \mathbf{r}) to include in the overall state vector. What we have *not* done is chosen a subset of these Dirac delta basis functions¹³ as our basis. This would be very strange—our representation of the wavefunction would allow it to have a value at one gridpoint, and at the next gridpoint, but not in between. Spatially separated Dirac deltas do not spatially overlap at all; the matrix elements of the kinetic energy operator:

$$\langle \mathbf{r}_i | \hat{K} | \mathbf{r}_j \rangle = \int \delta(\mathbf{r} - \mathbf{r}_i) \left(-\frac{\hbar^2}{2m} \nabla^2 \right) \delta(\mathbf{r} - \mathbf{r}_j) d\mathbf{r} \quad (3.21)$$

would all be zero for $i \neq j$, disallowing any flow of amplitude from one point in space to another by virtue of it not being able to pass through the intervening points. Neither have we chosen a set of two-dimensional boxcar functions centred on each gridpoint with width Δx in each direction. These cover all space in between gridpoints, but are no good because they are not twice differentiable everywhere, and hence the kinetic energy operator's matrix elements cannot be evaluated. No, neither of these bases will do. To interpret our spatial grid as a basis, we need a set of functions $[FUNC_{ij}(r)]$ (where i and j are the indices of the gridpoints in the x and y directions respectively) that have unit norm, are nonzero only at one gridpoint and are zero at all others, and are twice differentiable everywhere in our spatial region. Infinite choices present themselves to us,

¹¹For the sake of example—256 × 256 is a more realistic minimum.

¹²One for each position within the two dimensional space we're representing.

¹³Strictly, distributions, not functions, but “basis distributions” just doesn't have the right ring to it.

differing only in their incompleteness—the choice of which state vectors they will and won’t be able to represent. A sensible choice is that we want to be able to represent the state vectors whose wavefunctions do not change much between adjacent gridpoints, and we are happy for the necessary incompleteness of our basis to exclude wavefunctions with any sort of structure in between gridpoints.

The discrete Fourier transform to the rescue

It turns out that discretising space in this way is indeed equivalent to choosing an entirely sensible basis. This is made clearer by thinking in terms of what we have done in Fourier space, which I’ll quickly go through now.

One possible basis for representing all possible state vectors is the Fourier basis $\{|\mathbf{k}_{i,j}\rangle\}$. With it, any state vector (whose wavefunction is nonzero only within the 2D region) can be represented as the sum of basis vectors whose wavefunctions are 2D plane waves, also localised to the 2D region:

$$\langle \mathbf{r} | \mathbf{k}_{i,j} \rangle = \begin{cases} \frac{1}{\sqrt{A}} e^{i\mathbf{k}_{i,j} \cdot \mathbf{r}} & (\mathbf{r} \text{ within 2D region}) \\ 0 & (\mathbf{r} \text{ not within 2D region}), \end{cases} \quad (3.22)$$

where A is the area of the 2D region and the wavevector of each plane wave is

$$\mathbf{k}_{i,j} = \left[\frac{2\pi i}{L_x}, \frac{2\pi j}{L_y} \right]^T, \quad (3.23)$$

where i and j are (possibly negative) integers. Any state vector localised to the 2D region can then be written as the infinite sum:

$$|\psi\rangle = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \langle \mathbf{k}_{i,j} | \psi \rangle |\mathbf{k}_{i,j}\rangle \quad (3.24)$$

$$\Rightarrow \psi(\mathbf{r}) = \begin{cases} \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \langle \mathbf{k}_{i,j} | \psi \rangle \frac{1}{\sqrt{A}} e^{i\mathbf{k}_{i,j} \cdot \mathbf{r}} & (\mathbf{r} \text{ within 2D region}) \\ 0 & (\mathbf{r} \text{ not within 2D region}). \end{cases} \quad (3.25)$$

So $\{\langle \mathbf{k}_{i,j} | \psi \rangle\}$ are simply the coefficients of the 2D Fourier series of $\psi(\mathbf{r})$.

What does this have to do with our discretised space? These basis functions $\{|\mathbf{r}| \mathbf{k}_{i,j}\rangle\}$ don’t have the required properties for a discrete basis. For one, there are an infinite number of them, and we require 25. Secondly, all of them are nonzero everywhere within the 2D region, whereas we require each basis function to be nonzero at exactly one of out 25 gridpoints.

We can solve the first problem by truncating the Fourier series. By only including basis vectors $|\mathbf{k}_{i,j}\rangle$ for which:

$$\begin{cases} i \in [-\frac{N_x}{2}, \frac{N_x}{2} - 1] & (N_x \text{ even}) \\ i \in [-\frac{N_x-1}{2}, \frac{N_x-1}{2}] & (N_x \text{ odd}) \end{cases} \quad (3.26)$$

and

$$\begin{cases} j \in [-\frac{N_y}{2}, \frac{N_y}{2} - 1] & (N_y \text{ even}) \\ j \in [-\frac{N_y-1}{2}, \frac{N_y-1}{2}] & (N_y \text{ odd}) \end{cases} \quad (3.27)$$

we include only the N_x and N_y (both equal to 5 in our example) longest wavelengths in each respective spatial dimension. This is a sensible truncation with a physically meaningful interpretation. By making it, we are no longer able to represent state vectors with

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

short wavelength components. Because the kinetic energy operator, when represented in the Fourier basis, is:

$$\langle \mathbf{k}_{ij} | \hat{K} | \mathbf{k}_{i'j'} \rangle = \frac{\hbar^2 k^2}{2m} \delta_{ii'} \delta_{jj'}, \quad (3.28)$$

where $k = |\mathbf{k}_i|$, by excluding basis vectors with larger wavevectors, we are excluding state vectors with large kinetic energy. Thus the truncation is a kinetic energy cutoff, and is an accurate approximation whenever a simulation is such that the system is unlikely to obtain kinetic energies above the cutoff.¹⁴

Now we have 25 basis vectors, a discrete Fourier basis, but their spatial wavefunctions still don't have the property of being nonzero only at a single gridpoint each. But consider the following superposition of Fourier basis vectors:

$$|\mathbf{r}_{ij}\rangle = \sum_{i'j'} e^{-i\mathbf{k}_{i'j'} \cdot \mathbf{r}_{ij}} |\mathbf{k}_{i'j'}\rangle \quad (3.29)$$

where $N = 5$ is the number of gridpoints in each spatial direction.

Analytically, state vectors in Hilbert spaces with spatial degrees of freedom such as this are often represented in the spatial basis $|\mathbf{r}\rangle$. This basis has one basis vector for each point in space (and hence has an infinite number of basis vectors), each of which is a Dirac delta. We then call $\psi(\mathbf{r})$ the *wavefunction* corresponding to the state vector.¹⁵

If the state vector is known in some basis $|n\rangle$, then The state vector's projection onto one of them is therefore:

$$\langle \mathbf{r} | \psi \rangle = \int \delta(\mathbf{r}' - \mathbf{r}) \psi(\mathbf{r}') d\mathbf{r}', \quad (3.30)$$

which is obviously just the wavefunction $\psi(\mathbf{r})$ again. The above reasoning is of course a bit circular, as the braket has been evaluated as in integral in the spatial basis itself. These basis vectors $|\mathbf{r}\rangle$ can be represented in any basis, however, and so the projection of a state vector onto one of them can similarly be evaluated in any basis $|n\rangle$ as:

$$\langle \mathbf{r} | \psi \rangle = \sum_n \langle \mathbf{r} | n \rangle \langle n | \psi \rangle \quad (3.31)$$

$$= \sum_n U_{rn} \psi_n, \quad (3.32)$$

where $\psi_n = \langle n | \psi \rangle$ is the vector component of

. For example the Fourier transform of a delta function is [CITE]:

$$\mathcal{F}\{\delta(\mathbf{r}' - \mathbf{r})\}(\mathbf{k}) = e^{-i\mathbf{k} \cdot \mathbf{r}}, \quad (3.33)$$

which is interpreted as the projection of the spatial basis vector $|\mathbf{r}\rangle$ onto the Fourier basis vector $|\mathbf{k}\rangle$. So the spatial basis vector $|\mathbf{r}\rangle$ has a representation as the sum of Fourier basis vectors:

$$|\mathbf{r}\rangle = \int \langle \mathbf{k} | \mathbf{r} \rangle |\mathbf{k}\rangle d\mathbf{k} \quad (3.34)$$

$$= \int e^{-i\mathbf{k} \cdot \mathbf{r}} |\mathbf{k}\rangle d\mathbf{k} \quad (3.35)$$

$$(3.36)$$

and so the projection of the state vector $|\psi\rangle$ onto a spatial basis vector $|\mathbf{r}\rangle$ can be computed in the Fourier basis too, as:

$$\langle \mathbf{r} | \psi \rangle = \sum_{\mathbf{k}} \langle \mathbf{k} | \psi \rangle |\mathbf{k}\rangle \quad (3.37)$$

¹⁶ The state vector can then be represented as a column vector, with one complex number giving the amplitude of each component of the state vector in the chosen basis.

3.4 Discrete degrees of freedom

3.4.1 The interaction picture

A common situation in atomic physics is to be simulating the internal state of an atom, armed with the knowledge that only a small number of atomic states are able to become occupied.

- Sometimes called a "rotating frame"
- Is equivalent to basis change where new basis functions differ by a time-dependent phase factor
- Is defined by a time-independent Hamiltonian
- This has the effect of moving some time dependence into the operators (demonstrate, by writing some operators with the unitary in front of them. As you can see it is simply a change of basis - but a time-dependent one.)
- No need to remain in the same interaction picture - can be redefined arbitrarily often throughout a simulation and state vectors transformed into new basis.

3.4.2 Unitary integration

Direct exponentiation via diagonalisation of Hamiltonian

- Unitary - doesn't mean it's accurate but means it won't explode. Great for the bits of your simulation that are explosion-prone but don't matter (like regions of space where the wavefunction is near zero but the potential is large or steep)
- error is order [whatever it is] per timestep, not great compared to RK4
- can be combined with RK4 to improve accuracy (see later subsection)

Approximate exponentiation by operator product

[Comment in this section how the approximate total unitary can be used at each timestep to define an interaction picture, and the remaining dynamics simulated with RK4 like RKIILIP does in the spatial basis. Interaction pictures are really useful!]

3.5 Continuous degrees of freedom

Every symbol on the paper has a representation in a computer. State vectors are arrays of complex numbers, operators are matrices - differential operators are no exception. Operators must have a concrete representation, their matrix elements can be computed and then things solved with linear algebra. For discrete degrees of freedom, the matrix representation of the operators may be known, for continuous ones you can find the matrix elements once you define what basis functions you will use [show how]. Or, for the DVR it is a little more subtle (because it's not a basis) but still basically the same process.

- Have to be discretised in some way to simulate on a computer - need basis functions. Often a spatial basis is used. Any spatial basis must be combined with assumptions about what the wavefunction is doing at points in between the basis points, in order to define differential operators. Finite differences approximates wavefunction as low-order polynomial in between points (is this equivalent to a polynomial *basis*?)

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

Probably not.). Fourier method assume the Fourier series of the wavefunction at the given points can be used to interpolate between points (or the wavefunction can be Fourier transformed and calculations can be done directly in the Fourier basis). DVR is not actually a spatial basis despite appearances. It assumes the wavefunction is a sum of polynomial 'shape functions', but these shape functions are not basis functions as they are not orthonormal. This is why it is called a representation rather than a basis. Regardless, the shape functions can be used to define an interpolation of the wavefunction between points and thus define differential operators.

3.5.1 Finite differences

Show a matrix representation of a few different finite differences, to show that differential operators really are just matrices. They approximate the function as low-order polynomials about each point. You can take them to arbitrarily high order.

3.5.2 The Fourier basis

Because of properties of Fourier transforms, derivatives can be taken in Fourier space as simple multiplication. This is essentially because differential operators are diagonal in the Fourier basis. So you can use this fact to define a differential operator in the spatial basis [show matrix] ...or, you could just implement it with Fourier transforms, since FFTs are faster than matrix-vector multiplication ($O(n \log(n))$ rather than $O(n^2)$)

Split operator method

- Equivalent to approximate exponentiation via operator product with the discrete case

3.5.3 Harmonic oscillator basis functions

3.6 Finding ground states

3.6.1 Imaginary time evolution

3.6.2 Successive over-relaxation

3.6.3 Generalisation to excited states via Gram–Schmidt orthonormalisation

Directly diagonalising a Hamiltonian can be costly in a spatial basis. Another approach is to find the ground state using one of the above techniques, and then repeat the process, subtracting off the wavefunction's projection onto the already found ground state at every step. This yields the lowest energy state that is orthogonal to the first - i.e the first excited state. Repeating the process, but subtracting off *both* eigenstates found so far, then yields the second excited state and so forth. This is simply the Gram-Schmidt process for finding orthonormal vectors, with the additional step of relaxing each vector to the lowest possible energy for each one - this ensures the eigenstates of the Hamiltonian are produced, rather than a different orthogonal basis. Extra conditions can be imposed on the wavefunction at each relaxation step in order to obtain particular solutions in the case of degenerate eigenstates. For example, a phase winding can be imposed in order to obtain a particular harmonic oscillator state - otherwise this process produces an arbitrary superposition of basis states that have equal energy.

```
rev:    25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date:   Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

3.7 The finite-element discrete variable representation

- Plots of representation of sine wave as function of number of points between FEDVR and FD. RMS error of a derivative operator perhaps.

[explanation of how it works, comparison of implementation with RSP4 vs something like RK4. RK4 is more general purpose, method does not need to be modified for different Hamiltonians. Main limitation is inability to factor out fast dynamical phases, see RK4ILIP for solution to this. MPI implementation and scaling properties with increasing cores. Superscaling at low number of cores. Mention how my implementation can tolerate high network latency due to early sending of data before all local computations are complete. Mention that it is ripe for GPU processing. Limitations: vulnerable to Runge's phenomenon for sharp potentials. Can't increase the order of the polynomials much because small spacing at the edges requires tiny timesteps. Possible solution: preconditioning the potential to be an approximation better representable in the DVR basis.]

3.8 Fourth order Runge–Kutta in an instantaneous local interaction picture

Consider the differential equation for the components of a state vector $|\psi(t)\rangle$ in a particular basis with basis vectors $|n\rangle$. This might simply be the Schrödinger equation, or perhaps some sort of nonlinear or other approximate, effective or phenomenological equation not corresponding to pure Hamiltonian evolution. Though they may have additional terms, such equations are generally of the form:

$$\frac{d}{dt} \langle n|\psi(t)\rangle = -\frac{i}{\hbar} \sum_m \langle n|\hat{H}(t)|m\rangle \langle m|\psi(t)\rangle, \quad (3.38)$$

where $\langle n|\hat{H}(t)|m\rangle$ are the matrix elements in that basis of the Hamiltonian $\hat{H}(t)$, which in general can be time dependent, or even a function of $|\psi(t)\rangle$, depending on the exact type of equation in use. If $\hat{H}(t)$ is almost diagonal in the $|n\rangle$ basis, then the solution to (3.38) is dominated by simple dynamical phase evolution, that is:

$$|\psi(t)\rangle \approx \sum_m e^{-\frac{i}{\hbar} E_m t} |m\rangle, \quad (3.39)$$

where E_m is the energy eigenvalue corresponding to the eigenstate $|m\rangle$.

A transformation into an interaction picture (IP) [48, p318] is commonly used to treat this part of the evolution analytically, before solving the remaining dynamics with further analytics or numerics. For numerical methods, integration in the interaction picture allows one to use larger integration timesteps, as one does not need to resolve the fast oscillations around the complex plane due to this dynamical phase.

Choosing an interaction picture typically involves diagonalising the time-independent part of a Hamiltonian, and then proceeding in the basis in which that time-independent part is diagonal. However, often one has a good reason to perform computations in a different basis, in which the time independent part of the Hamiltonian is only approximately diagonal,¹⁷ and transforming between bases may be computationally expensive (involving large matrix-vector multiplications). Furthermore, the Hamiltonian may change sufficiently during the time interval being simulated that the original time-independent Hamiltonian no longer dominates the dynamics at later times. In both these cases it would still be useful to factor out the time-local oscillatory dynamics in whichever basis is being used, in order to avoid taking unreasonably small timesteps.

¹⁷For example, a spatial basis which allows for partitioning the integration region over multiple nodes on a cluster or cores on a GPU.

To that end, suppose we decompose $\hat{H}(t)$ into diagonal and non-diagonal (in the $|n\rangle$ basis) parts at each moment in time:

$$\hat{H}(t) = \hat{H}_{\text{diag}}(t) + \hat{H}_{\text{nondiag}}(t), \quad (3.40)$$

and use the diagonal part at a specific time $t = t'$ to define a time-independent Hamiltonian:

$$\hat{H}_0^{t'} = \hat{H}_{\text{diag}}(t'), \quad (3.41)$$

which is diagonal in the $|n\rangle$ basis. We can then use then use $\hat{H}_0^{t'}$ to define an interaction picture state vector:

$$|\psi_I^{t'}(t)\rangle = e^{\frac{i}{\hbar}(t-t')\hat{H}_0^{t'}} |\psi(t)\rangle, \quad (3.42)$$

which obeys the differential equation:

$$\frac{d}{dt} |\psi_I^{t'}(t)\rangle = e^{\frac{i}{\hbar}(t-t')\hat{H}_0^{t'}} \frac{d}{dt} |\psi(t)\rangle + \frac{i}{\hbar} \hat{H}_0^{t'} |\psi_I^{t'}(t)\rangle, \quad (3.43)$$

where:

$$|\psi(t)\rangle = e^{-\frac{i}{\hbar}(t-t')\hat{H}_0^{t'}} |\psi_I^{t'}(t)\rangle \quad (3.44)$$

is the original Schrödinger picture (SP) state vector.

This transformation is exact, no approximations or assumptions have been made. If indeed the dynamics of $|\psi(t)\rangle$ in the given basis are dominated by fast oscillating dynamical phases, that is, the diagonals of $\hat{H}_{\text{diag}}(t)$ are much greater than all matrix elements of $\hat{H}_{\text{nondiag}}(t)$ in the $|n\rangle$ basis, then solving the differential equation (3.43) for $|\psi_I^{t'}(t)\rangle$ should allow one to use larger integration timesteps than solving (3.38) directly. And if not, then it should do no harm other than the (small) computational costs of computing some extra scalar exponentials.

Equation (3.42) defines an *instantaneous* interaction picture, in that it depends on the dynamics at a specific time $t = t'$, and can be recomputed repeatedly throughout a computation in order to factor out the fast dynamical phase evolution even as the oscillation rates change over time. It is *local* in that $H_0^{t'}$ is diagonal in the $|n\rangle$ basis, which means that transformations between Schrödinger picture and interaction picture state vectors involves ordinary, elementwise exponentiation of vectors, rather than matrix products. Thus (3.42), (3.43) and (3.44) can be written componentwise as:

$$\langle n | \psi_I^{t'}(t) \rangle = e^{i(t-t')\omega_n^{t'}} \langle n | \psi(t) \rangle, \quad (3.45)$$

$$\frac{d}{dt} \langle n | \psi_I^{t'}(t) \rangle = e^{i(t-t')\omega_n^{t'}} \frac{d}{dt} \langle n | \psi(t) \rangle + i\omega_n^{t'} \langle n | \psi_I^{t'}(t) \rangle, \quad (3.46)$$

and:

$$\langle n | \psi(t) \rangle = e^{-i(t-t')\omega_n^{t'}} \langle n | \psi_I^{t'}(t) \rangle, \quad (3.47)$$

where we have defined:

$$\omega_n^{t'} = \frac{1}{\hbar} \langle n | \hat{H}_0^{t'} | n \rangle \quad (3.48)$$

```
rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

This is in contrast to fourth order Runge–Kutta in the interaction picture (`RK4IP`) [49], in which the interaction picture uses the Fourier basis and thus transforming to and from it involves fast Fourier transforms (FFTs). `RK4IP` was developed to augment computations in which FFTs were already in use for evaluating spatial derivatives, and so its use of FFTs imposes no additional cost. Nonetheless, an interaction picture based on the kinetic term of the Schrödinger equation (which is the term of the Hamiltonian that `RK4IP` takes as its time-independent part) may not be useful if that term does not dominate the Hamiltonian, as in the case of a Bose–Einstein condensate in the Thomas–Fermi limit. We compare the two methods below.

3.8.1 Algorithm

The *fourth order Runge–Kutta in an instantaneous local interaction picture* `RK4ILIP` algorithm is now obtained by using (3.42) to define a new interaction picture at the beginning of each fourth-order Runge–Kutta (`RK4`) integration timestep. The differential equation and initial conditions supplied to the algorithm are in the ordinary Schrödinger picture, and the interaction picture is used only within a timestep, with the Schrödinger picture state vector returned at the end of each timestep. Thus differential equations need not be modified compared to if ordinary `RK4` were being used, and the only modification to calling code required is for a function to compute and return ω'_n .

Being based on fourth order Runge–Kutta integration, this new method enjoys all the benefits of a workhorse method that is time-proven, and—as evidenced by its extremely widespread use—at a sweet-spot of ease of implementation, accuracy, and required computing power [50].

Below is the resulting algorithm for performing one integration timestep. It takes as input the time t_0 at the start of the timestep, the timestep size Δt , an array ψ_0 containing the components $\{ \langle n | \psi(t_0) \rangle \}$ of the state vector at time t_0 , a function $F(t, \psi)$ which takes a time and (the components of) a state vector and returns an array containing the time derivative of each component, and a function $G(t, \psi)$ which takes the same inputs and returns an array containing the interaction picture oscillation frequency ω_n for each component at that time.

For example, for the case of the Gross–Pitaevskii equation [51] in the spatial basis $\psi(\mathbf{r}, t) = \langle \mathbf{r} | \psi(t) \rangle$, these would be:

$$F(t, \psi(\mathbf{r}, t)) = -\frac{i}{\hbar} \left[\underbrace{-\frac{\hbar^2}{2m} \nabla^2}_{\hat{H}_{\text{nondiag}}} + \underbrace{V(\mathbf{r}, t) + g|\psi(\mathbf{r}, t)|^2}_{\hat{H}_{\text{diag}}} \right] \psi(\mathbf{r}, t), \quad (3.49)$$

and

$$G(t, \psi(\mathbf{r}, t)) = \frac{1}{\hbar} \left[\underbrace{V(\mathbf{r}, t) + g|\psi(\mathbf{r}, t)|^2}_{\hat{H}_{\text{diag}}} \right]. \quad (3.50)$$

Note that each symbol in bold in the algorithm below denotes an array containing one element for each basis vector $|n\rangle$, subscripts denote the different stages of `RK4`, and all arithmetic operations between arrays are elementwise¹⁸. The only opportunity for non-elementwise operations to occur is within F , which contains the details (via \hat{H}_{nondiag}) of any couplings between basis states for whatever system of equations is being solved, for example, using FFTs or finite differences to evaluate the Laplacian in (3.49).

Algorithm 1 `RK4ILIP`

i: function `RK4ILIP`($t_0, \Delta t, \psi_0, F$)

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

¹⁸For example, the expression $\mathbf{a} \leftarrow e^{-i\omega\Delta t} \mathbf{b}$ indicates that for all n , $a_n \leftarrow e^{-i\omega_n \Delta t} b_n$, where a_n denotes the n^{th} element of \mathbf{a} etc.

```

2:    $f_1 \leftarrow F(t_0, \psi_0)$                                 ▷ First evaluation of Schrödinger picture DE
3:    $\omega \leftarrow G(t_0, \psi_0)$                                 ▷ Oscillation frequencies:  $\hbar\omega_n = \langle n|\hat{H}_{\text{diag}}(t_0)|n\rangle$ 
4:    $k_1 \leftarrow f_1 + i\omega\psi_0$                                 ▷ Evaluate (3.46) with  $t - t' = 0$ 
5:    $\phi_1 \leftarrow \psi_0 + k_1 \frac{\Delta t}{2}$                       ▷ First RK4 estimate of IP state vector, at  $t = t_0 + \frac{\Delta t}{2}$ 
6:    $\psi_1 \leftarrow e^{-i\omega \frac{\Delta t}{2}} \phi_1$                   ▷ Convert first estimate back to SP with (3.47)
7:    $f_2 \leftarrow F(t_0 + \frac{\Delta t}{2}, \psi_1)$                 ▷ Second evaluation of Schrödinger picture DE
8:    $k_2 \leftarrow e^{i\omega \frac{\Delta t}{2}} f_2 + i\omega\phi_1$     ▷ Evaluate (3.46) with  $t - t' = \frac{\Delta t}{2}$ 
9:    $\phi_2 \leftarrow \psi_1 + k_2 \frac{\Delta t}{2}$                       ▷ Second RK4 estimate of IP state vector, at  $t = t_0 + \frac{\Delta t}{2}$ 
10:   $\psi_2 \leftarrow e^{-i\omega \frac{\Delta t}{2}} \phi_2$                   ▷ Convert second estimate back to SP with (3.47)
11:   $f_3 \leftarrow F(t_0 + \frac{\Delta t}{2}, \psi_2)$                 ▷ Third evaluation of Schrödinger picture DE
12:   $k_3 \leftarrow e^{i\omega \frac{\Delta t}{2}} f_3 + i\omega\phi_2$     ▷ Evaluate (3.46) with  $t - t' = \frac{\Delta t}{2}$ 
13:   $\phi_3 \leftarrow \psi_2 + k_3 \Delta t$                       ▷ Third RK4 estimate of IP state vector, at  $t = t_0 + \Delta t$ 
14:   $\psi_3 \leftarrow e^{-i\omega \Delta t} \phi_3$                   ▷ Convert third estimate back to SP with (3.47)
15:   $f_4 \leftarrow F(t_0 + \Delta t, \psi_3)$                 ▷ Fourth evaluation of Schrödinger picture DE
16:   $k_4 \leftarrow e^{i\omega \Delta t} f_4 + i\omega\phi_3$     ▷ Evaluate (3.46) with  $t - t' = \Delta t$ 
17:   $\phi_4 \leftarrow \psi_3 + k_4 \Delta t$                       ▷ Fourth RK4 estimate, at  $t = t_0 + \Delta t$ 
18:   $\psi_4 \leftarrow e^{-i\omega \Delta t} \phi_4$                   ▷ Convert fourth estimate back to SP with (3.47)
19:  return  $\psi_4$                                          ▷ Return the computed SP state vector at  $t = t_0 + \Delta t$ 
20: end function

```

Note on imaginary time evolution

When `RK4ILIP` is used for imaginary time evolution (`ITE`) [52], the oscillation frequencies ω may have a large imaginary part. If the initial guess is different enough from the ground state, then the exponentials in (3.45), (3.46) and (3.47) may result in numerical overflow. To prevent this, one can define a clipped copy of ω ,

$$\omega_{\text{clipped}} = \text{Re}(\omega) + i \begin{cases} -\frac{\log X}{\Delta t} & \text{Im}(\omega)\Delta t < -\log X \\ \text{Im}(\omega) & -\log X \leq \text{Im}(\omega)\Delta t \leq \log X \\ \frac{\log X}{\Delta t} & \text{Im}(\omega)\Delta t > \log X \end{cases}, \quad (3.51)$$

where X is very large but less than the largest representable floating-point number, and use ω_{clipped} in the exponents instead. In the below results I used `RK4ILIP` with `ITE` to smooth initial states of a Bose–Einstein condensate after a phase printing, and performed clipping with¹⁹ $\log X = 400$.

This clipped version of ω should be used in all exponents in the above algorithm, but only in exponents—not in the second term of (3.46). If it is used everywhere then all we have done is chosen a different (less useful) interaction picture, and the algorithm will still overflow. By clipping only the exponents, we produce temporarily “incorrect” evolution²⁰, limiting the change in magnitude of each component of the state vector to a factor of X per step (remembering that X is very large). This continues for the few steps that it takes `ITE` to get all components of the state vector to within a factor of X of the ground state, after which no clipping is necessary and convergence to the ground state proceeds as normal, subject to the ordinary limitations on which timesteps may be used with `ITE`.

¹⁹400 being about half the largest (base e) exponent representable in double-precision floating point.

²⁰Of no concern since we are using `ITE` as a relaxation method, and are not interested in intermediate states. Only the final state’s correctness concerns us.

3.8.2 Domain of improvement over other methods

For simulations in the spatial basis, `RK4ILIP` treats the spatially local part of the Hamiltonian analytically to first order, and hence can handle larger potentials than ordinary `RK4`.

```

rev:      25 (79ce3eb23e1f)
author:   Chris Billington <chrisjbillington@gmail.com>
date:     Tue Sep 06 23:02:15 2016 +1000
summary:  More work on continuous DOF part of numerics chapter

```

However, since a global energy offset can be applied to any potential with no physically meaningful change in the results, ordinary RK4 can also handle large potentials — if they are large due to a large constant term which can simply be subtracted off.

So RK4ILIP is only of benefit in the case of large *spatial variations* in the potential. Only one constant can be subtracted off potentials without changing the physics — subtracting a spatially varying potential would require modification of the differential equation in the manner of a gauge transformation in order to leave the system physically unchanged²¹.

However that's not quite all: large spatial variation in potentials often comes with the prospect of the potential energy turning into kinetic energy, in which case RK4ILIP is also of little benefit, since in order to resolve the dynamical phase due to the large kinetic term, it would require timesteps just as small as those which ordinary RK4 would need to resolve the dynamical phase evolution from the large potential term.

This leaves RK4ILIP with an advantage only in the case of large spatial variations in the potential that do not lead to equally large kinetic energies. Hence the examples I show in the next section are ones in which the condensate is trapped in a steep potential well—the trap walls are high and hence involve large potentials compared to the interior, but do not lead to large kinetic energies because the condensate is trapped close to its ground state.

The Fourier split-step (FSS) method [53] (see section [TODO]) also models dynamical phases due to the potential analytically to low order. As such it is also quite capable of modeling large potentials. However, it requires that all operators be diagonal in either the spatial basis or the Fourier basis [53]. Therefore BECs in rotating frames, due to the Hamiltonian containing an angular momentum operator, are not amenable to simulation with FSS²².

This use of FFTs in both the FSS and RK4IP methods necessarily imposes periodic boundary conditions on a simulation, which may not be desirable. By contrast, if different boundary conditions are desired, finite differences instead of FFTs can be used to evaluate spatial derivatives in the RK4 and RK4ILIP methods, so long as a sufficiently high-order finite difference scheme is used so as not to unacceptably impact accuracy.

Along with the ability to impose arbitrary boundary conditions, finite differences require only local data, that is, only points spatially close to the point being considered need be known in order to evaluate derivatives there. This makes finite differences amenable to simulation on cluster computers [55, 100], with only a small number of points (depending on the order of the scheme) needing to be exchanged at node-boundaries each step. By contrast, FFT based derivatives require data from the entire spatial region. Whilst this can still be parallelised on a GPU, where all the data is available, it cannot be done on a cluster without large amounts of data transfer between nodes [56]. Thus, RK4 and RK4ILIP, being implementable with finite difference schemes, are considerably friendlier to cluster computing.

Table 3.1 summarises the capabilities of the four methods considered in the following results section. RK4ILIP is the only method capable of modelling a large spatial variation in the potential term whilst being locally parallelisable, and supporting arbitrary operators and boundary conditions.

3.8.3 Results

Here I compare four numerical methods: Fourier split-step (FSS), fourth order Runge–Kutta in the interaction picture (RK4IP), ordinary fourth order Runge–Kutta (RK4), and my new method — fourth order Runge–Kutta in an instantaneous local interaction picture (RK4ILIP).

The example chosen is a 2D simulation of a turbulent Bose–Einstein condensate, in both a rotating and nonrotating frame. For the nonrotating frame the differential

```
rev:    25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date:   Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

Method	RK4	RK4IP	RK4ILIP	FSS
Error	$\mathcal{O}(\Delta t^4)$	$\mathcal{O}(\Delta t^4)$	$\mathcal{O}(\Delta t^4)$	$\mathcal{O}(\Delta t^2)$
FFTs per step	4	4	4	2
Large ΔV	No	No	Yes	Yes
Large kinetic term	No	Yes	No	Yes
Arbitrary operators	Yes	Yes [†]	Yes	No
Locally parallelisable	Yes	No	Yes	No
Arbitrary boundary conditions	Yes	No	Yes	No

Table 3.1: Advantages and disadvantages of four timestepping methods for simulating Bose-Einstein condensates. *Large ΔV* refers to whether the method can simulate potentials that vary throughout space by an amount larger than the energy scale $2\pi\hbar/\Delta t$ associated with the simulation timestep Δt . *Arbitrary operators* refers to whether the method permits operators that are not diagonal in either the spatial or Fourier basis, such as angular momentum operators. *Locally parallelisable* means the method can be formulated so as to use only spatially nearby points in evaluating operators, and thus is amenable to parallelisation by splitting the simulation over multiple cores in the spatial basis. [†] Whilst one can include arbitrary operators within the RK4IP method, only operators diagonal in Fourier space can be analytically treated the way RK4IP treats the kinetic term, and so there is no advantage for these terms over ordinary RK4.

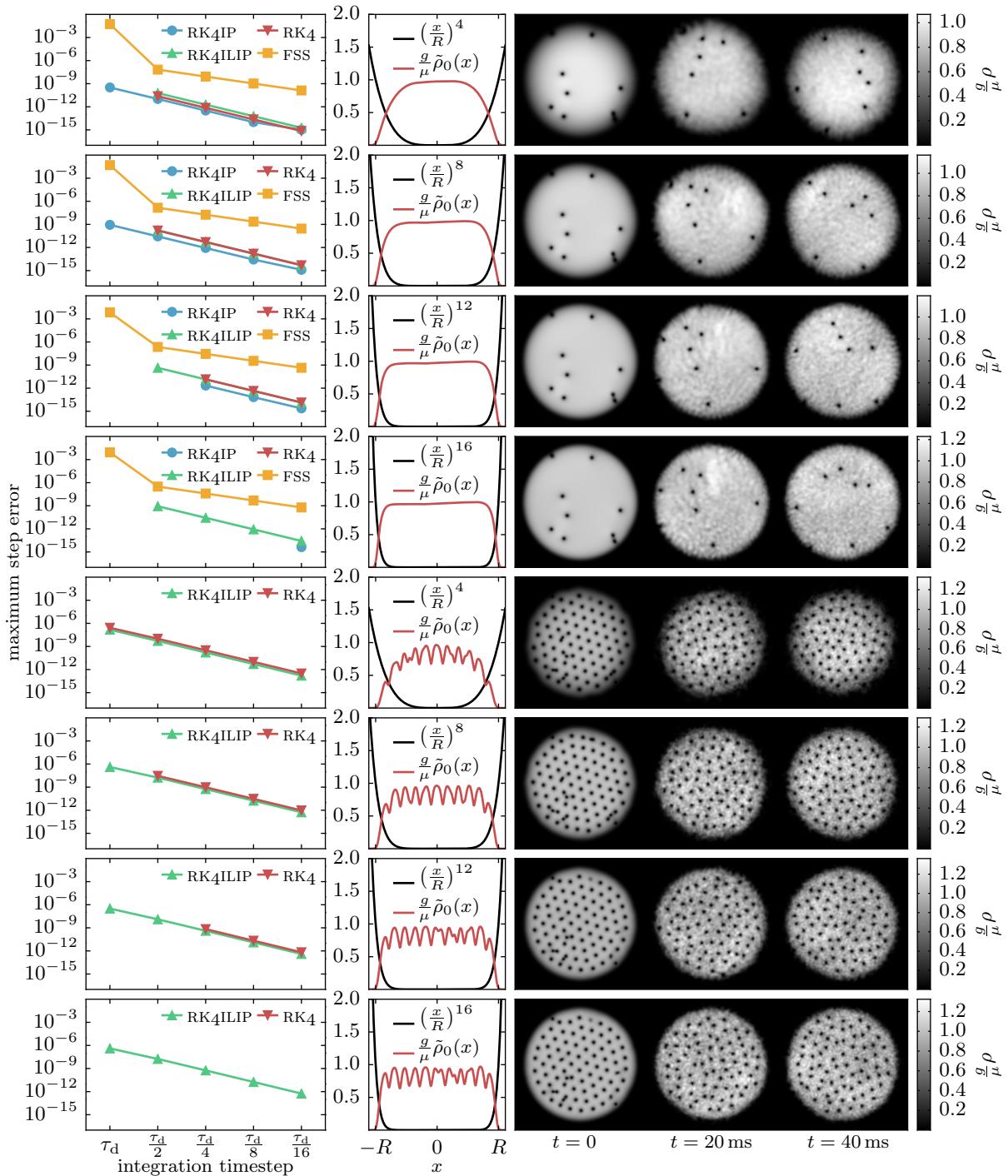


Figure 3.1: Results of simulations to compare `rk4ILIP` to other timestepping methods. Top four rows: Nonrotating frame simulations with four different radial power-law potentials. Bottom four rows: Rotating frame simulations with same four potentials. Left column: maximum per-step error $\int |\psi - \tilde{\psi}|^2 dr / \int |\tilde{\psi}|^2 dr$ of fourth order Runge–Kutta (`RK4`), its interaction picture variants (`RK4IP` and `RK4ILIP`) and Fourier split-step (`FSS`) as a function of timestep. Solutions were checked every 100 timesteps against a comparison solution $\tilde{\psi}$ computed using half sized steps for `RK4` methods, and quarter sized steps for `FSS`. Simulations encountering numerical overflow not plotted. Centre column: potential (black) and average density $\bar{\rho}_0$ of the initial state (red) over a slice of width $R/5$ in the y direction. Right column: Density of solution at initial, intermediate and final times for each configuration simulated (taken from `RK4ILIP` results). `RK4ILIP` is the only method usable in rotating frames and not encountering overflow in the steeper traps for the timesteps considered.

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

equation simulated was equation (3.49), and for the rotating frame the same equation was with an additional two terms added to the Hamiltonian:

$$\hat{H}_{\text{rot}} + \hat{H}_{\text{comp}} = -\boldsymbol{\Omega} \cdot \hat{\mathbf{L}} + \frac{1}{2}\hbar m^2 \Omega^2 r^2 \quad (3.52)$$

$$= i\hbar\Omega \left(x \frac{\partial}{\partial y} - y \frac{\partial}{\partial x} \right) + \frac{1}{2}\hbar m^2 \Omega^2 r^2. \quad (3.53)$$

The addition of the first term transforms the original Hamiltonian into a frame rotating at angular frequency Ω in the (x, y) plane, and is equivalent to the Coriolis and centrifugal forces that appear in rotating frames in classical mechanics [57]. The second term is a harmonic potential that exactly compensates for the centrifugal part of this force. In this way the only potential in the rotating frame is the applied trapping potential, and the only effect of the rotating frame is to add the Coriolis force.

Four trapping potentials were used, all radial power laws with different powers. These examples were chosen to demonstrate the specific situation in which RK4ILIP provides a benefit over the other methods for spatial Schrödinger-like equations, as discussed above.

The results of 120 simulation runs are shown in Figure 3.1. Each simulation was of a ^{87}Rb condensate in the $|F = 2, m_F = 2\rangle$ state, in which the two-body s -wave scattering length is $a = 98.98$ Bohr radii [58]. The simulation region was $20 \mu\text{m}$ in the x and y directions, and the Thomas–Fermi radius of the condensate was $R = 9 \mu\text{m}$. The chemical potential was $\mu = 2\pi\hbar \times 1.91 \text{ kHz}$, which is equivalent to a maximum Thomas–Fermi density $\rho_{\text{max}} = 2.5 \times 10^{14} \text{ cm}^{-3}$ and a healing length $\xi = 1.1 \mu\text{m}$. There were 256 simulation grid points in each spatial dimension, which is 14 points per healing length.

Four different potentials were used, all of the form $V(r) = \mu(r/R)^\alpha$ with $\alpha = 4, 8, 12, 16$. For the rotating frame simulations, the rotation frequency was $\Omega = 2\pi \times 148 \text{ Hz}$. This is 89% of the effective harmonic trap frequency, defined as the frequency of a harmonic trap that would have the same Thomas–Fermi radius given the same chemical potential.

All ground states were determined using successive over-relaxation (See section [TODO]) with sixth-order finite differences for spatial derivatives. For the nonrotating simulations, convergence was reached with $\Delta\mu/\mu < 1 \times 10^{-13}$, with:

$$\Delta\mu = \sqrt{\frac{\langle \psi | (\hat{H} - \mu)^2 | \psi \rangle}{\langle \psi | \psi \rangle}}, \quad (3.54)$$

where \hat{H} is the nonlinear Hamiltonian and $\langle \mathbf{r} | \psi \rangle$ is the condensate wavefunction, which does not have unit norm. For the rotating frame simulations the ground states converged to $\Delta\mu/\mu \approx 9 \times 10^{-7}, 2 \times 10^{-6}, 3 \times 10^{-6}$ and 2×10^{-6} for $\alpha = 16, 12, 8$, and 4 respectively.

After each ground state was found, it was multiplied by a spatially varying phase factor corresponding to the phase pattern of a number of randomly positioned vortices:

$$\psi_{\text{vortices}}(x, y) = \psi_{\text{groundstate}}(x, y) \prod_{n=1}^N e^{\pm_n i \arctan 2(y - y_n, x - x_n)} \quad (3.55)$$

where $\arctan 2$ is the two-argument arctan function,²³ $N = 30$, \pm_n is a randomly chosen sign, and (x_n, y_n) are vortex positions randomly drawn from a Gaussian distribution centred on $(0, 0)$ with standard deviation equal to the Thomas–Fermi radius R . The same seed was used for the pseudorandom number generator in each simulation run, and so the vortex positions were identical in each simulation run.

After vortex phase imprinting, the wavefunctions were evolved in imaginary time [52]. For the nonrotating frame simulations, imaginary time evolution was performed for a time interval equal to the chemical potential timescale $\tau_\mu = 2\pi\hbar/\mu$, and for the rotating

²³Defined as the principle value of the argument of the complex number $x + iy$: $\arctan 2(y, x) = \text{Arg}(x + iy)$.

frame simulations, for $\tau_\mu/10$. This was done to smooth out the condensate density in the vicinity of vortices, producing the correct density profile for vortex cores. However, since imaginary time evolution decreases the energy of the state indiscriminately, it also had the side effect of causing vortices of opposite sign to move closer together and annihilate. This decreased the number of vortices, and is the reason the smoothing step in the rotating frame simulations was cut short to $\tau_\mu/10$, as otherwise all vortices had time to annihilate with one of the lattice vortices. A vortex pair in the process of annihilating is visible in Figure 3.1 as a partially filled hole in the initial density profile near the top of the condensate in the $\alpha = 4, 12$, and 16 rotating frame simulations.²⁴

²⁴The initial states for the four different potentials are not identical, so by chance the corresponding vortex in the $\alpha = 8$ case was not close enough to a lattice vortex to annihilate.

The smoothed, vortex imprinted states were then evolved in time for 40 ms. For each simulation, five different timesteps were used: $\Delta t = \tau_d, \tau_d/2, \tau_d/4, \tau_d/8, \tau_d/16$, where $\tau_d = m\Delta x^2/\pi\hbar \approx 2.68 \mu\text{s}$ is the dispersion timescale associated with the grid spacing Δx , defined as the time taken to move one gridpoint at the phase velocity of the Nyquist mode.

For the nonrotating frame simulations, spatial derivatives for the **RK4** and **RK4ILIP** methods were determined using the Fourier method [see section TODO]. This was to ensure a fair comparison with the other two methods, which necessarily use Fourier transforms to perform computations pertaining due to the kinetic term.

For the rotating frame simulations, sixth-order finite differences with zero boundary conditions were used instead for the kinetic terms of the **RK4** and **RK4ILIP** methods, which were the only two methods used for those simulations (due to the other methods being incompatible with the angular momentum operator required for a rotating frame). This choice was fairly arbitrary, but did allow the condensate to be closer to the boundary than is otherwise possible with the periodic boundary conditions imposed by use of the Fourier method for spatial derivatives. This is because the rotating frame Hamiltonian is not periodic in space, and so its discontinuity at the boundary can be a problem if the wavefunction is not sufficiently small there.

As shown in Figure 3.1, all methods tested generally worked well until they didn't work at all, with the per-step error of **RK4**-based methods being either small and broadly the same as the other **RK4**-based methods, or growing rapidly to the point of numerical overflow (shown as missing datapoints). The break down of **FSS** was less dramatic, though it too had a clear jump in its per-step error for larger timesteps. Comparing methods therefore came down to mostly whether or not a simulation experienced numerical overflow during the time interval being simulated.

The main result was that **RK4ILIP** and **FSS** remained accurate over the widest range of timesteps and trap steepnesses, with **RK4** and **RK4IP** requiring ever smaller timesteps in order to not overflow as the trap steepness increased.

For the rotating frame simulations, which were only amenable to the **RK4** and **RK4ILIP** methods, the same pattern was observed, with **RK4** only working at smaller timesteps as the trap steepness was increased, and ultimately diverging for all timesteps tested at the maximum trap steepness. By contrast, **RK4ILIP** remained accurate over the entire range of timesteps at the maximum trap steepness.

3.8.4 Discussion

As mentioned, **RK4ILIP** is mostly useful for continuum quantum mechanics only when there are large spatial differences in the potential, which cannot give rise to equally large kinetic energies²⁵. Furthermore, the advantage that **RK4ILIP** has over other methods with that same property is that it does not require a particular form of Hamiltonian or a particular method of evaluating spatial derivatives. The former means it is applicable in rotating frames or to situations with unusual Hamiltonians, and the latter means it can be used with finite differences or **FEDVR** [54] and thus is amenable to parallelisation on a cluster computer.

²⁵This is essentially due to such a situation violating the condition we laid out at the beginning of this section — that the simulation basis must be nearly an eigenbasis of the total Hamiltonian.

The ability to model large spatial variations in the potential provides only a narrow domain of increased usefulness over other methods. If a large kinetic energy results from the large potential, then the method requires just as small timesteps as any other. And if the large potential is supposed to approximate an infinite well, then an actual infinite well may be modelled using zero boundary conditions, negating the need for something like `RK4ILIP`. However, when potential wells are steep, but not infinitely steep, here `RK4ILIP` provides a benefit. The only other model that can handle these large potentials—Fourier split-step—has the disadvantage that it cannot deal with arbitrary operators such as those arising from a rotating frame, and is not parallelisable with local data. The benefits of parallelisability are obvious, and the above results demonstrate `RK4ILIP`'s advantage at simulating BECs in tight traps and rotating frames.

For systems with discrete degrees of freedom, `RK4ILIP` may be useful in the case where an approximate diagonalisation of the Hamiltonian is analytically known, and when the Hamiltonian's eigenvalues vary considerably in time (making a single interaction picture insufficient to factor out dynamical phases throughout the entire simulation). In this situation an analytic transformation into the diagonal basis can be performed at each timestep (or the differential equation analytically re-cast in that basis in the first place), and `RK4ILIP` can be used to factor out the time-varying dynamical phase evolution at each timestep. An example may be an atom with a magnetic moment in a time-varying magnetic field which varies over orders of magnitude. The transformation into the spin basis in the direction of the magnetic field can be analytically performed, and if the field varies by orders of magnitude, so do the eigenvalues of the Hamiltonian. Although the eigenvalues in this case and other similar cases can be computed analytically too, unless all time dependence of the Hamiltonian is known in advance of the simulation, it would be difficult to incorporate this into a re-casting of the differential equation in a time-dependent interaction picture. `RK4ILIP` may be useful in these cases to automate this process and evolve the system in the appropriate interaction picture at each timestep.

This page intentionally left blank

CHAPTER

4

Development of a cold atom physics experiment

- Describe how the apparatus was developed over time, what it was designed to do and everything we learned along the way. Include vacuum system design and bakeout, optical setups, atomic transport, etc. Not comprehensive, as I have not played a large role in much of the development of the apparatus itself.
- Dual species apparatus
- -> Vacuum chamber and bakeout
- Research visit report
- -> Describe setup of Tübingen experiment (despite lack of results). It's a simple MOT setup, so describing it includes details of optics required for a MOT, which I didn't work on in the Monash apparatus, so this is a good place to describe everything that goes into a MOT experimentally. Include full optics layout I designed.
- -> Optics setup
- -> [include layout diagram of setup I designed]

4.0.1 Vacuum system

Our vacuum system comprises three chambers—two *source* chambers and a central chamber. These sections are separated by differential pumping tubes, which are simply small tubes that restrict the passage of gas between the chambers¹.

After assembly of the vacuum system, we constructed an oven around it and baked it out at approximately 200°C for about two weeks. This increases the rate of outgassing from the interior surfaces, decreasing the extent to which later outgassing of water and hydrogen can limit our pressure. After bakeout the system's pressure was measured with a residual gas analyser (RGA) to be in the vicinity of 10^{-11} Torr. This was in the central chamber, which was and still is being pumped by two ion pumps and a titanium sublimation pump.

During the bakeout, the two side chambers were being pumped on by a turbomolecular pump, which was removed after baking was complete. This pumping was required to remove the products of outgassing in those chambers since the conduction to the central chamber through the differential pumping tubes is so poor.

After bakeout, the two side chambers² were brought up to atmospheric pressure with

¹This works very effectively at very low pressures due to the fact that atoms very rarely collide with each other—and so even a factor of a thousand difference in pressure across the tube results in very little pressure-gradient force on the gas.

²Which can be sealed from the central chamber with gate valves.

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter



Figure 4.1: The temporary oven built around the vacuum system during bakeout. Temperatures were controlled with variacs providing variable voltage to the heater tape, and temperatures were monitored using thermocouples and an SR630 thermocouple monitor set to cut off the power if the temperature went too high. The SR630 was also being polled constantly over the network so that the bakeout status could be monitored remotely.

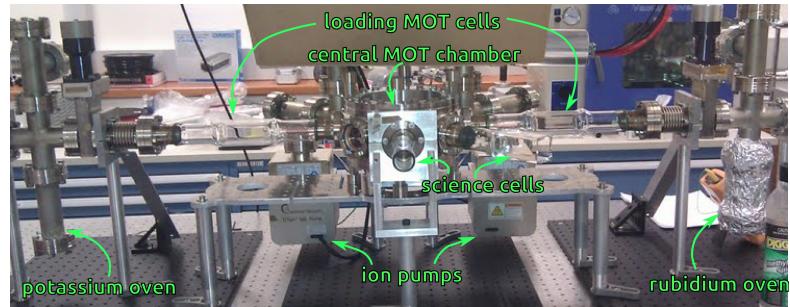


Figure 4.2: The vacuum system after bakeout and insertion of alkali metal ampoules.

³Which comes off the interior surfaces without much baking, and in fact protects the system from other—more difficult to remove—contaminants such as water from depositing on the surfaces.

⁴Operating under a safety and control system written on embedded electronics by Phil Starkey.

⁵39°C and 64°C for rubidium and potassium respectively.

dry nitrogen³, and ampoules of the alkali metals we'll be using in our experiments—rubidium and potassium—were inserted. After pumping down and lightly baking once more, two metal weights that had been magnetically suspended above the ampoules were released and — after a few attempts—used to break the ampoules open. Heating elements on both chambers⁴ have since been heating the metals to near their melting points⁵, which should, based on the known dependence of the metals' partial pressures on temperature, provide a background pressure of approximately 10^{-6} Torr [59, 60].

Unfortunately it seemed that the pressure of hydrogen in the rubidium source chamber had been increasing since the bake, as recently noticed by RGA measurements of hydrogen in the central chamber, which diminished upon closing the gate valve in between. To address this, we are modified that end of the vacuum system to include an ion pump. This was possible to do without having to break existing vacuum. Only the rubidium end was affected, presumably because it has a o-ring valve (where the pump during bakeout was attached) which a) could not be baked as hot as the rest of the system and b) may have been leaking. The addition of the new pump and a short bake of the local area appears to have alleviated the problem (Figure 4.4).

```

rev:    25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date:   Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
  
```

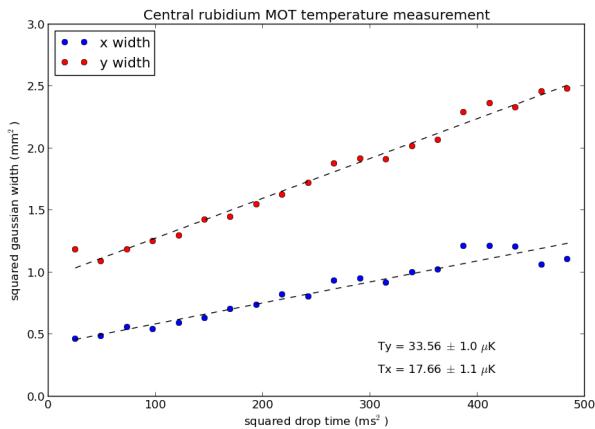


Figure 4.3: A temperature measurement of the rubidium MOT in the central chamber. Temperature was determined by measuring the rate of expansion of the atom cloud when released from the magnetic trap. The magnetic trap was switched off, and then the atoms were imaged with a pulse of light after being allowed to expand for a time. The expansion time was varied and compared to the size of the condensate in order to determine the expansion rate, which has a simple relation to temperature. This temperature was obtained without a polarisation gradient cooling stage, however it is lower than the Doppler limit ($\approx 140 \mu\text{K}$) due to some PGC taking place regardless.

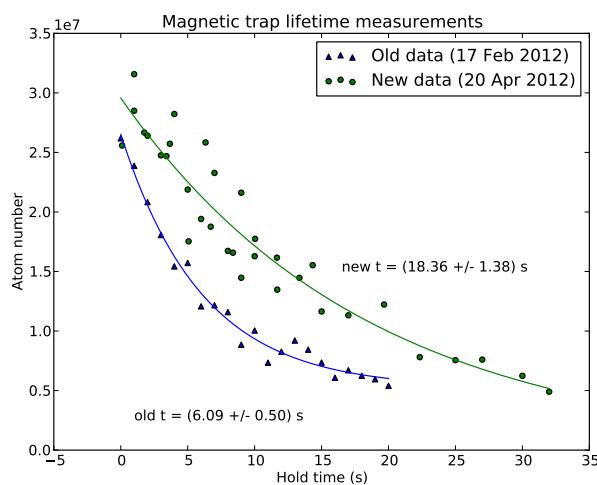


Figure 4.4: A measurement of the magnetic trap lifetime before and after an extra pump was added to the rubidium source chamber. The trap lifetime was measured by turning off the cooling beams such that atoms were trapped only magnetically, then waiting a time before fluorescence imaging with a pulse of light. By computing the number of atoms remaining from the fluorescence, and comparing with the delay time, the decay rate due to collisions with background gases was determined. An increase in the atom number has since been obtained, resulting in 1.2×10^8 atoms, with the same trap lifetime.

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

4.0.2 Cooling and trapping of atoms

MOTs have been formed in both source chambers using anti-Helmholtz magnetic coils and Doppler beams. The cooling beams are circularly polarised in order to pump the cooling transition of the atoms, and there is also repump light to move atoms back into the cooling cycle when they decay to an undesired groundstate. Both MOTs collect atoms from the relatively high pressure background gas from the alkali ovens in the source chambers, before those atoms are transported to the central chamber.

4.0.3 Transport of atoms

The initial idea was to use magnetic transport to move atoms into the central chamber of the vacuum system. Due to a delay in the construction of magnetic coil control electronics, another alternative was investigated in the meantime—the push-beam method. Magnetic transport, once attempted, proved to be less efficient (due high collisional losses during the slow transit), as well as technically challenging, and so we have continued to use the push-beam method, in which atoms are given momentum by resonant light from a single beam. Atoms are pushed by this beam into the central chamber, where they are caught in the MOT there. This has been done for rubidium but not yet for potassium.

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

Software for experiment control and analysis

- Describe the problem of controlling experiments of this nature, our proposed solutions, and the implementation thereof in the *labscript suite*. Unix philosophy, sandboxing, lack of requirement of trust in third party software -> fault tolerance. Describe runmanager, BLACS, lyse in detail. Argue benefits of code-based experiments, especially that standard software development industry practices can be very useful to doing science. techniques Draw heavily on paper.
- Possibly include content from transfer report (as of yet decided not to include)
- Possibly include content from documentation - or write updated versions thereof.
- New screenshots
- labscrip
- runmanager
- BLACS
- mise

We present the *labscript suite*, an open-source experiment control system for automating shot-based experiments and their analysis. Experiments are composed as Python code, which is used to produce low-level hardware instructions. They are queued up and executed on the hardware in real time, synchronized by a pseudoclock. Experiment parameters are manipulated graphically, and analysis routines are run as new data are acquired. With this system, we can easily automate exploration of parameter spaces, including closed-loop optimization.

5.1 Introduction

Modern experiments in quantum science demand flexible, autonomous control of heterogeneous hardware. Many such experiment are *shot-based*: a single experiment shot comprises analog, digital, and radiofrequency (rf) outputs operating under precise timing, as well as synchronized camera exposures and voltage measurements. Bose–Einstein condensation (BEC) experiments [61], for example, require a timing resolution down to a few hundred nanoseconds, and may last for up to a minute. Output must, therefore, be hardware timed, requiring devices be programmed with instructions in advance of an

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

experiment shot. Most measurements of interest require numerous shots, to build up statistics, or to observe the response of the system to varying parameters. Such repetition is common to experiments employing cold quantum gases or trapped ions for precision metrology [62, 63], quantum computation [64, 65], and quantum simulation [66, 67].

Individual shots are typically complex, requiring the coordination of many devices. This coordination is the role of a *control system*. A good control system should automate the programming of devices based on a high-level description of the experiment logic [68]. It should handle the repetition of shots and automated variation of experiment parameters, the increasingly complex demands of which cannot be rapidly, robustly, and continuously met by human operators. It should automate analysis, leading to the prospect of closed-loop control: the results of analysis influencing subsequent experiment shots. Applications of such closed-loop control include autonomous algorithmic optimization of parameters, and automatic recalibration in response to environmental drifts.

Most existing control systems take one of two approaches for providing a human interface to programming hardware. One is text-based, in which experiments are written using a general purpose programming language [69]. In the other, experiments are instead described graphically using a custom user interface [70–74]. The text-based approach natively offers the advantages of a programming language, particularly control-flow tools such as conditional statements, loops, and functions. Its disadvantage is that frequently varied settings and parameters may be hidden in hundreds of lines of code. Conversely, the graphical-user-interface (GUI) approach makes experiment parameters more accessible to the user, but features providing for complex experiment logic must be anticipated and implemented specifically [73, 74].

The two approaches need not be mutually exclusive: by separating experiment parameters from experiment logic, parameters can be manipulated graphically and logic textually [75, 76]. We contend that by using a high-level programming language with appropriate hardware abstraction, text-based control can be more comprehensible to a newcomer than an equivalent graphical representation of hardware instructions.

We present the *labscript suite* which utilizes a hybrid text-and-GUI approach for control and builds on previous work by addressing the need for autonomous control, analysis, and optimization. Hardware control is abstracted, providing an identical software interface to devices of a common type. Graphical interfaces are dynamically generated based on the current hardware set in use. Analysis is an integral part of the control system, with user-written analysis routines run automatically on new data. Finally, analysis results can modify subsequent experiment shots, closing the feedback loop on analysis and control.

5.2 An overview of the labscript suite

The labscript suite comprises several programs, each performing one main function; the flow of data between programs is shown in Fig. 5.1. Each experiment shot is associated with a single file: each program writes to and reads from this file as required before passing it on to the next program. Programs may be run on separate computers, communicating over the network using the ZeroMQ messaging library [77], exchanging references to the experiment file.

We use the Hierarchical Data Format (HDF version 5) [78] which provides cross-platform storage of large scientific datasets. Exploiting the extensibility of HDF, each file is a complete description of the experiment shot. The HDF file begins life containing only experiment parameters. As it is passed between components of the labscript suite, the file grows to contain the hardware instructions, acquired, data and analysis results. Metadata is also stored including user-written scripts and version control information. This maintains a comprehensive record of the experiment shot for post-hoc analysis, reproducibility, and publication preparation.

```
rev:      25 (79ce3eb23e1f)
author:   Chris Billington <chrisjbillington@gmail.com>
date:    Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

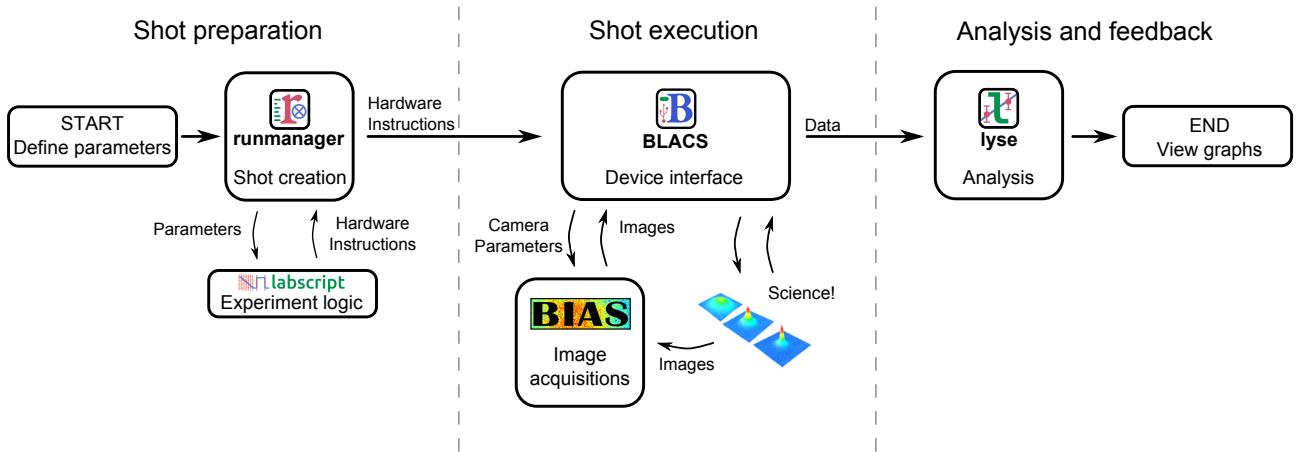


Figure 5.1: Each experiment shot comprises three stages: preparation, execution, and analysis. Arrows indicate how the HDF file for an experiment shot passes between software components of the labscript suite. Only the shot execution stage is coupled to hardware timing, allowing new shots to be created and queued while others are running. Similarly, analysis can be performed on executed shots at any time.

Attempts to standardize laboratory device programming have largely failed, with only a minority of devices conforming to standards such as SCPI (Standard Commands for Programmable Instruments). [79] This calls for abstraction to shield the user from low-level interaction. We have created a software library for Python [80, 81], **labscript** (Sec. 5.4), which provides a common interface for commanding output and measurements from devices. The user writes the experiment logic in Python, and **labscript** generates the required hardware instructions, including a clocking signal for timing (Sec. 5.3).

The **labscript** suite separates experiment logic (written in Python) from experiment parameters, which are manipulated in a GUI. The GUI, **runmanager** (Sec. 5.5), creates the HDF file for the experiment shot and stores the parameters within. If a parameter is a list of values, rather than a single value, **runmanager** creates an HDF file (a prospective shot) for each value. If lists are entered for more than one parameter, **runmanager** creates a file for each point in the resulting parameter space.

For each shot, **labscript** inserts the parameters from the HDF file into the experiment logic, compiles hardware instructions for each device, and writes them to the same file. **runmanager** sends the compiled HDF files to **BLACS** (Sec. 5.6) which places them in a queue. **BLACS** interfaces with hardware devices either directly, or via secondary control programs such as **BIAS** (Sec. 5.7). **BLACS** programs the hardware and triggers the experiment shot to begin. The experiment then proceeds under hardware-timed control.

Once the experiment shot has finished, acquired data such as voltage time-series and images are added to the HDF file. **BLACS** then passes the file to a dedicated analysis system, **lyse** (Sec. 5.8). **lyse** coordinates the execution of analysis routines, which are Python scripts written by the user. These scripts may analyze individual shots or a sequence of shots as a whole. This facilitates autonomous analysis of results from parameter space scans, as experiment shots are completed.

The **labscript** software library can be applied to automatically generate shots based on the results of analysis. We have used this to implement a closed-loop optimization system, **mise** (Sec. 5.9).

```

rev:      25 (79ce3eb23e1f)
author:   Chris Billington <chrisjbillington@gmail.com>
date:     Tue Sep 06 23:02:15 2016 +1000
summary:  More work on continuous DOF part of numerics chapter

```

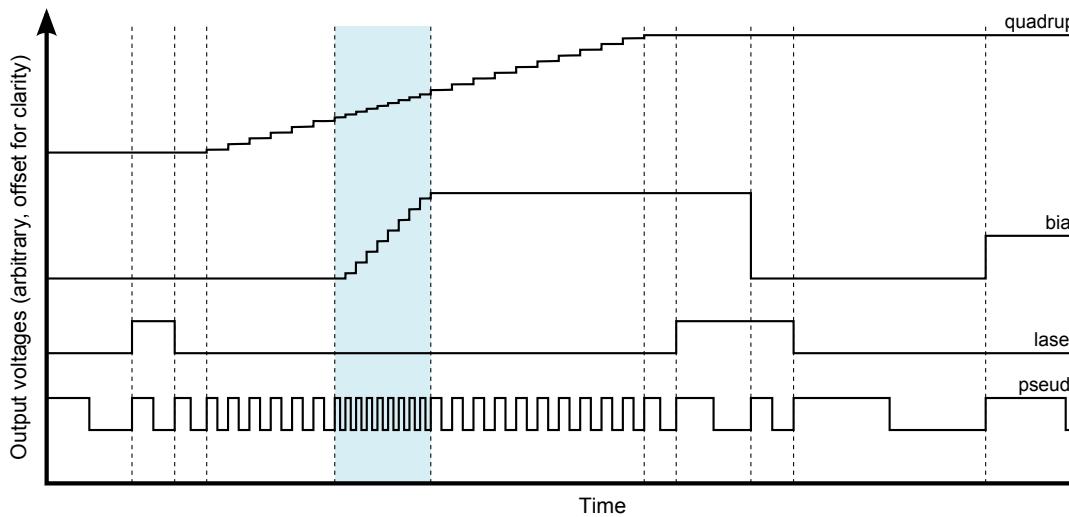


Figure 5.2: An example of digital and analog voltage outputs generated by the labscript code in Fig. 5.3. The pseudoclock (lower trace) ticks when a digital output must change, or at the requested sample rate for time-varying analog outputs (upper two traces). Dashed vertical lines indicate a change in the pseudoclock frequency. When multiple analog outputs are varying at the same time (shaded region), the pseudoclock ticks at the highest of their sampling rates.

5.3 Pseudoclock

A typical BEC experiment requires precise timing over a large range of time scales [61]. There are periods during which magnetic fields or laser intensities, for example, may change with sub-microsecond resolution. Conversely, there are periods during which no devices change their output for several seconds, e.g., loading a magneto-optical trap (MOT). To ensure accurate output during the rapid changes, hardware devices must be preloaded with a set of instructions that can be stepped through by a clock once the experiment begins. Stepping through instructions at a constant rate requires repetitive instructions during the more inactive periods. As many devices only support a limited number of instructions, a constant-rate clock limits the maximum sample rate. A common solution [71, 72, 74, 75] is a variable frequency master clock, or *pseudoclock*, which steps through instructions only when a clocked device needs to update an output (see Fig. 5.2). This removes the need for redundant instructions.

All devices sharing a pseudoclock must have an instruction when any one of their outputs changes value. This can lead to redundant instructions if only some of the devices are changing at a given time. The instruction limitations of one device may then limit another, e.g., some devices hold only a few thousand instructions in their internal memory, whereas others are limited only by the RAM of the host computer refilling their buffers. To solve this problem, we employ multiple pseudoclocks, assigning devices of similar memory limitations to the same clock. At the beginning of a shot, the software starts one pseudoclock (the *master clock*), which then triggers other clocks.

To be a useful pseudoclock, a device must be able to deterministically generate arbitrary digital signals, be hardware triggerable, and hold enough instructions for the required experiment. We currently use two pseudoclocks: the SpinCore PulseBlaster DDS-II-300-AWG, a commercial device based on a field-programmable gate array (FPGA); and the PineBlaster, a device developed in house based on a microcontroller. Both devices are externally referenced to a stable 10 MHz source.

The PineBlaster is a low-cost device using commodity hardware, based on the Arduino-

```

rev:      25 (79ce3eb23e1f)
author:   Chris Billington <chrisjbillington@gmail.com>
date:     Tue Sep 06 23:02:15 2016 +1000
summary:  More work on continuous DOF part of numerics chapter

```

```
(a) # Device definitions
PulseBlaster(name='pseudoclock_0', board_number=0)
NI_PCIE_6363(name='ni_card_0', parent_device=pseudoclock_0, clock_type='fast clock',
    MAX_name='ni_pcie_6363_0', clock_terminal='/ni_pcie_6363_0/PFI0')

# Channel definitions
Shutter (name='laser_shutter', parent_device=ni_card_0, connection='port0/line13')
AnalogOut(name='quadrupole_field', parent_device=ni_card_0, connection='ao0')
AnalogOut(name='bias_x_field', parent_device=ni_card_0, connection='ao1')

(b) # Experiment logic
start()
t = 0

# first laser pulse at t = 1 second
t += 1; laser_shutter.open(t)
t += 0.5; laser_shutter.close(t)
t += 0.4;

t += quadrupole_field.ramp(t, duration=5, initial=0, final=3, samplerate=4) # samplerate in Hz
# start ramping the bias field 3 seconds before the quadrupole ramp ends
bias_x_field.ramp(t-3, duration=1, initial=0, final=2.731, samplerate=8)
# t is now 6.9s, the end of the quadrupole field ramp

# second laser pulse
t += 0.4; laser_shutter.open(t)
t += 1; bias_x_field.constant(t, value=0.0)
t += 0.5; laser_shutter.close(t)
t += 2

# hold bias field at bias_x_final_field for 2 seconds before finishing shot
bias_x_field.constant(t, value=bias_x_final_field)
t += 2
stop(t)
```

Figure 5.3: An example `labscript` file. The connection table (a) defines a pseudoclock and a multifunction DAC object and configures three output channels. This is followed by the experiment logic (b) which commands output from these channels by name at times specified by the variable `t`. The experiment logic refers to the parameter `bias_x_final_field` which is set in `runmanager` (Sec. 5.5).

like Digilent ChipKIT Max32 microcontroller prototyping board¹. The board is flashed with a program that accepts clock instructions over universal serial bus (USB) and executes them with deterministic timing. It is capable of clocking at up to 10 MHz (100 ns between rising edges) with a resolution of 25 ns. The PineBlaster needs one instruction for each change in clock rate (see Fig. 5.2) and supports up to 15 000 instructions.

¹The source code for turning a ChipKIT Max32 into a PineBlaster is available at <http://hardware.labscriptsuite.org/>

`labscript` provides support for adding new pseudoclocks. It uses an intermediate format for storing timing instructions; implementing a new pseudoclock entails translating them into the required format for the hardware.

Some experiments require the time between instructions to be determined *during* a shot. This can be achieved by pausing the pseudoclocks until some condition is met. A common example [74–76] is waiting for a sufficient level of fluorescence from a loading MOT. Both the PulseBlaster and the PineBlaster support *wait instructions*, which pause output until resumed by a trigger. These instructions, when used in tandem with devices such as voltage comparators, can command the experiment to wait for events of interest.

5.4 The Labscript Library

We have created a Python software library, `labscript`, for defining experiment logic. `labscript` provides *hardware abstraction*, a common interface to control heterogeneous

```
rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

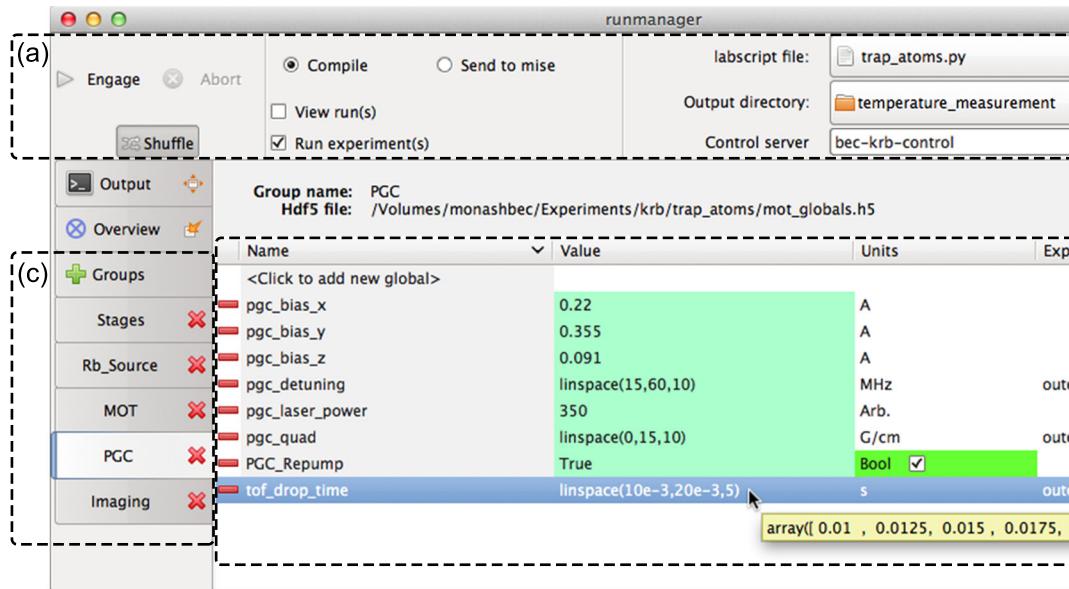


Figure 5.4: The `runmanager` interface for configuring experiment parameters. (a) The experiment logic is specified by the labscript file (here `trap_atoms.py`). HDF files for experiment shots created by `runmanager` are saved in the output directory. (b) The value of experiment parameters (“globals”) are specified by Python expressions and may have units. These can be single values (i.e., `350` or `True`), lists, or expressions creating lists (as shown for the globals `pgc_detuning`, `pgc_quad`, and `tof_drop_time`). A tooltip shows the evaluation of the global. The “Expansion” column specifies how lists of values are combined to construct a parameter space. (c) Globals can be separated into groups for convenience.

hardware. For example, the `DigitalOut` class provides `go_high(t)` and `go_low(t)` functions to set the state of a digital output at time `t`. The user calls these functions without regard to the underlying device, its method of programming, or the state of other digital outputs connected to the same device. Based on an experiment script containing such function calls, `labscript` automatically generates instructions for output and measurement devices as well as pseudoclocks. The automatic generation of pseudoclock instructions saves the user from dividing overlapping function ramps into segments (Fig. 5.2), or manually interpolating output values when a new time point is created on another channel.

An experiment script consists of two parts: a *connection table* (Fig. 5.3 (a)), and code defining the logic of the experiment (Fig. 5.3 (b)). The connection table provides a complete description of devices that are required for the experiment and how they are connected. `labscript` creates a set of Python objects based on the connection table, each with associated functions for commanding output or measurement from devices. The logic of the experiment is then defined by calling these functions with parameters such as time and output value.

As the experiment script is executable Python code, the user has full access to standard Python control flow tools, as well as standard and third party Python libraries. Using a high level language such as Python spares the user from low-level tasks such as memory management [68]. User-created functions can be stored in modules and imported into other experiment scripts. This allows complex experiments to be constructed from simple components, whilst maintaining comprehensibility, resulting in a gentler learning curve for new students. For example, one might define a `make_BEC()` function which contains the logic to form a Bose–Einstein condensate. While students might not fully understand

```

rev:      25 (79ce3eb23e1f)
author:   Chris Billington <chrisjbillington@gmail.com>
date:     Tue Sep 06 23:02:15 2016 +1000
summary:  More work on continuous DOF part of numerics chapter

```

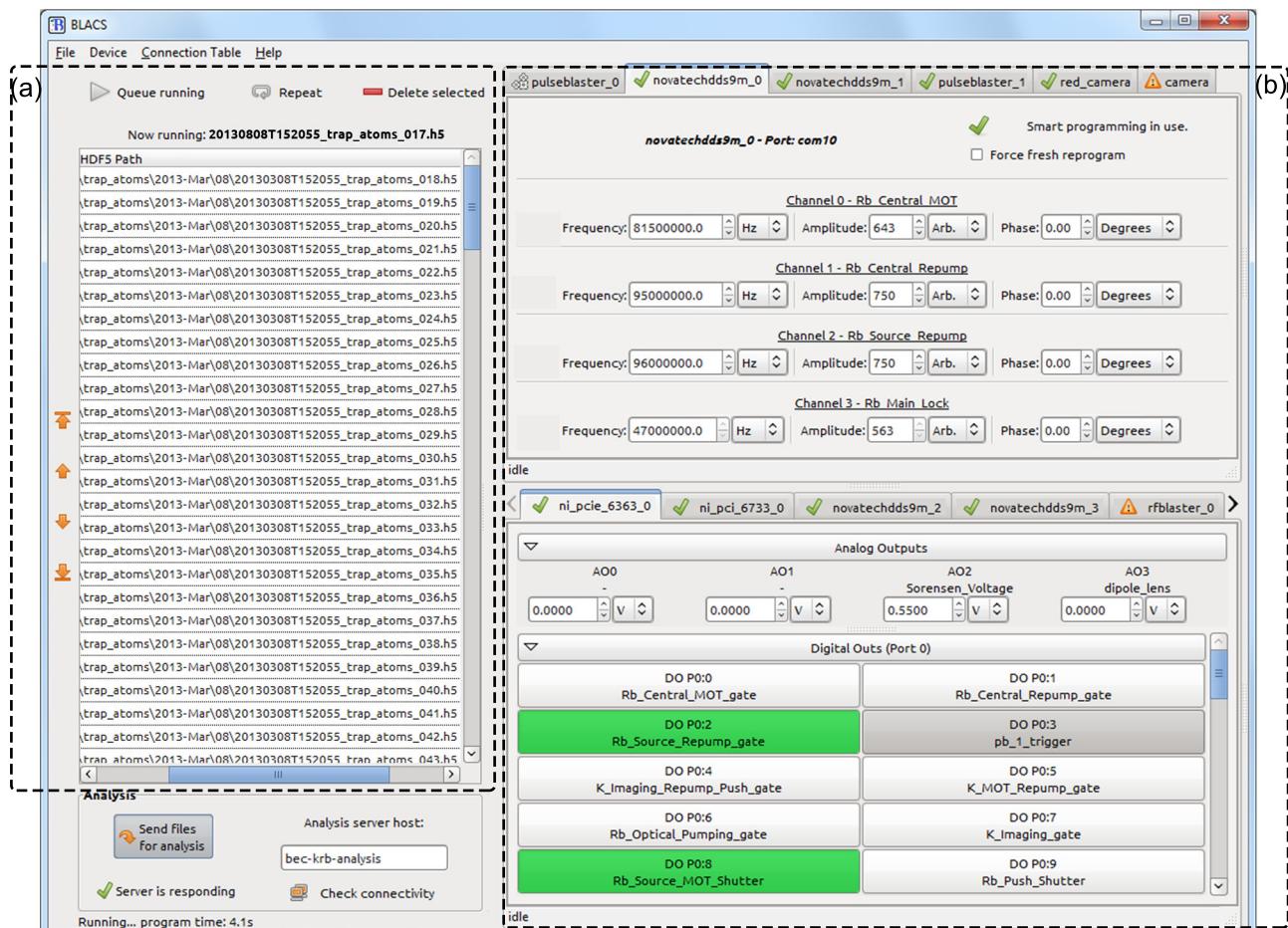


Figure 5.5: The BLACS interface for controlling hardware. (a) The queue of shots submitted via `runmanager`. (b) The manual control interface. Each tab controls one device. Controls for all outputs are automatically generated and are named based on the `BLACS` connection table.

the experiment logic to create a BEC, they can focus on subsequent experiment logic after a BEC is made. We have found that text based experiment scripts benefit not just from code re-use but also version control, bug tracking, and comparison of incremental changes (diffs).

When the experiment script is run and a timing sequence created, the `labscript` functions take into account hardware limitations and provide error messages if these are exceeded. If no errors are found, the hardware instruction set for all devices in the connection table is written to the HDF file.

While a text-based definition of experiment logic gives a broad overview of the timing sequence, it is not ideal for visualizing the device outputs to ensure the experiment logic is as intended. The hardware instructions generated by running experiment scripts are difficult to interpret (indeed, `labscript` was created to mitigate this very problem). Our program (`runviewer`) produces plots (similar to Fig. 5.2) of the hardware instructions generated by `labscript`, allowing quick diagnosis of the timing sequence before reaching for the oscilloscope.

```
rev:    25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date:   Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

5.5 Setting parameters—runmanager

Repeating experiments while varying parameters is a fundamental part of the scientific method. Anyone who has performed a quantum science experiment will be familiar with tweaking parameters to find a resonance, calibrating a measurement, or acquiring a large amount of scientific data prior to publication. The logic of the experiment does not change every time a parameter is adjusted, and it is cumbersome to edit numbers in a text file for each modification.

To ameliorate this, `labscrip`t experiments can take a series of parameters as input. The names and values of these parameters are defined in the graphical interface of `runmanager` (Fig. 5.4). The values can be any valid Python expression (such as `0.74`, `1E-3`, `sin(pi/2)`, or `True`) and can refer to each other. We call these parameters *globals* because they are available as global variables in experiment scripts, where they are simply referred to by name. For example, these globals might be used to specify the duration of a π -pulse, the delay between releasing atoms from a trap and imaging them, or the field strengths of bias magnetic coils. This provides a clean separation between code, which defines the nature of the experiment (such as creating a BEC with a vortex or performing a matter-wave mixing experiment), and parameters that modify individual shots.

The user may enter a list of values for a global, such as `[1, 2, 3]`, or `linspace(0, 10, 100)`. In this case `runmanager` produces a corresponding list of experiment shots: one for each value. If multiple globals are entered as lists, `runmanager` performs a Cartesian product, creating one shot for each point in the resulting parameter space. Two or more lists can be *zipped*, in which case `runmanager` iterates over these lists in lock-step when producing shots.

Specifying globals as lists makes it possible to explore complicated parameter spaces containing hundreds or thousands of shots. For example, one might investigate how the temperature of laser cooled atoms varies with laser detuning and magnetic field gradient. Taking the Cartesian product of ten field strengths and ten detunings results in a parameter space of one hundred points. Thermometry at each point in this parameter space commonly requires multiple shots to characterize the expansion rate of the atom cloud. A five-shot temperature measurement brings the number of shots to five hundred. Producing these shots amounts to entering three lists in `runmanager` and clicking on the “Engage” button, as shown in Fig. 5.4. `runmanager` then creates five hundred HDF files containing the globals for each shot. The experiment script is run for each shot, storing hardware instructions in each file². The HDF files are then submitted to `BLACS` for execution.

5.6 experiment execution—BLACS

`BLACS` coordinates input and output through hardware devices. These devices can be local, and thus under the direct control of `BLACS`, or connected to a different computer as part of a secondary control program such as `BIAS` (Sec. 5.7). `BLACS` provides both manual control of devices (through a GUI) and buffered execution of experiment shots.

The GUI for manual control is dynamically generated from a *lab connection table* that describes the current configuration of all connected devices. Each device is allocated a tab in the interface, containing controls for commanding output when in manual control mode (Fig. 5.5).

Upon submission to `BLACS`, HDF files containing hardware instructions are checked for validity and placed in a queue. The queue can be reordered, paused, or put on repeat. The validity check compares the connection table of each shot to the lab connection table, rejecting those with incompatible hardware. This prevents unintended device output that would produce nonsensical results and possibly damage equipment.

```
rev:      25 (79ce3eb23e1f)
author:   Chris Billington <chrisjbillington@gmail.com>
date:    Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

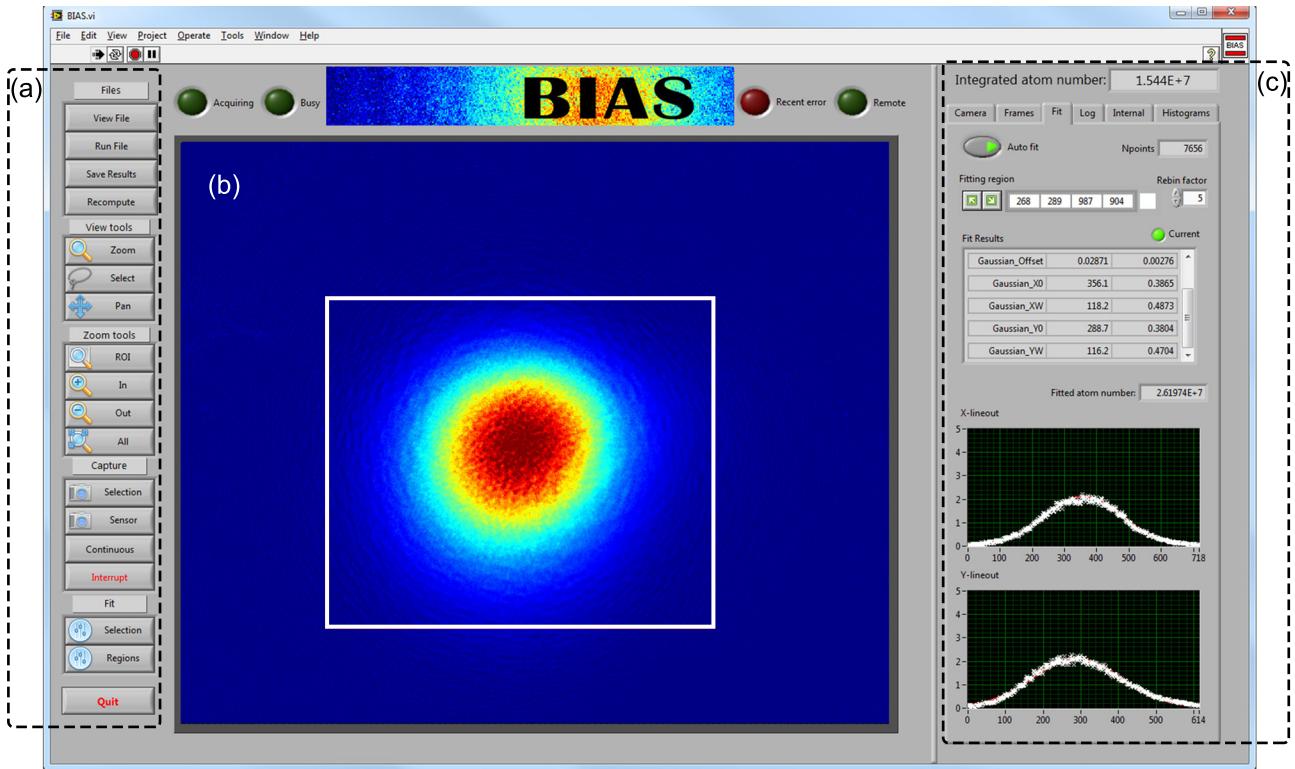


Figure 5.6: The **BIAS** interface displaying a laser cooled atom cloud. (a) Manual controls for loading and capturing images, selecting regions of interest and zooming. (b) Computed optical depth (OD) image of the atoms, with a region of interest (white) selected for fitting. Multiple regions of interest may be selected for multi-component atom clouds. (c) Atom number and cloud size are displayed for immediate feedback.

BLACS takes the first experiment in the queue, coordinates hardware programming, and sends a start trigger to the master pseudoclock. The experiment then proceeds under hardware timing. At the end of a shot, **BLACS** coordinates saving data acquired by devices to the HDF file, and returns to manual control mode. Each GUI control is updated to the final values of the shot, maintaining output continuity.

Laboratories are a hostile environment for hardware interface libraries. Power cycling of devices and unplugging of cables are common occurrences. A student tripping over a USB cable (health and safety implications notwithstanding) might be expected to cause an experiment to fail, however the control system ought to recover gracefully when it is plugged back in. Similarly, bugs in closed source drivers and libraries are points of failure outside of a users control.

To make our system robust against such hardware and software failures, **BLACS** implements a multiprocess architecture similar to the sandboxed tabs of the Google Chrome web browser [82]. For each device in **BLACS**, a *worker process* is spawned, which communicates with the hardware device. This makes **BLACS** robust against crashes: if one device has a problem it will not affect others. If a hardware device becomes unresponsive, or the device driver encounters a serious error, its isolation in a separate process prevents the GUI and other devices from suffering the same fate.

Should a worker process crash, the user is presented with the option of restarting the process, which will reload any device libraries it uses. It is worth noting that systems implemented in LabVIEW cannot force libraries to reload, so errors leading to an undefined

```
rev:      25 (79ce3eb23e1f)
author:   Chris Billington <chrisjbillington@gmail.com>
date:    Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

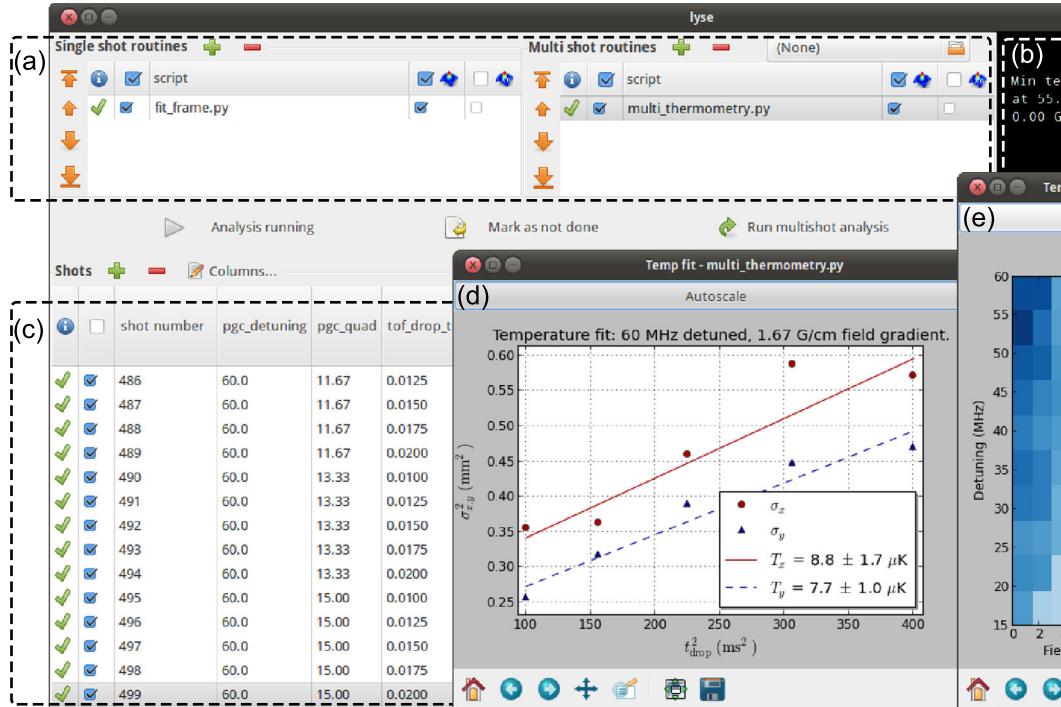


Figure 5.7: The `lyse` interface. (a) Routines can be selected to analyze single or multiple shots. (b) Terminal output from the analysis routines in (a). (c) Table of shots; columns show globals and analysis results. A small subset of columns is displayed here. (d) A fit yielding the temperature of laser cooled atoms prepared at a particular field gradient and detuning. (e) The results of the analysis in (d) repeated at each point in the parameter space.

state would only be remedied by restarting the entire control system.

The initialization of hardware in preparation for a shot is an important part of an experiment, and can significantly contribute to the experiment cycle time. The multiprocess architecture naturally provides for simultaneous programming of hardware devices, resulting in an increased experiment duty cycle. We have implemented a *smart programming* feature on many of our devices, further decreasing programming time, reprogramming them only if their instructions have changed since the previous shot (on a per-instruction basis when possible). Devices with large buffers and slow communication (such as the Novatech DDS9m rf synthesizer) benefit greatly from this technique.

5.7 Image acquisition—BIAS

Using secondary control programs to communicate with specific devices is desirable when software to do so exists and has been debugged, particularly software written in another programming language. `BLACS` integrates such programs into the control flow by sending them HDF files containing hardware instructions to program devices for execution upon a hardware trigger. `BLACS` notifies secondary control programs that the shot has completed, at which point they write any acquired data to the HDF file.

Our camera control and image acquisition system, `BIAS`, is one such program. `BIAS` is a LabVIEW application that operates scientific cameras, captures image sequences, and performs image processing tasks such as background subtraction, saturation correction, optical depth calculation, and simple 2D fitting.

```
rev:    25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date:   Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

Multiple instances of `BIAS` can be run simultaneously to control multiple cameras in one experiment. `BIAS` can also run as a stand-alone program for quick visualization of previously captured data or acquire images manually. Hardware communication in `BIAS` is abstracted through LabVIEW’s object hierarchy, allowing a camera class to be written for any vendor library.

LabVIEW provides convenient components for creating graphical interfaces, and `BIAS` displays raw and computed images as they become available (Fig. 5.6). Fit results such as atom cloud shape and atom number are prominently displayed to detect and diagnose problems as they occur. The camera acquisition area and regions of interest used to inform fits can be interactively adjusted, without needing to interrupt or recreate a currently running sequence of shots. Multiple regions of interest can be selected and their coordinates saved to the HDF file, enabling further analysis.

5.8 Analysis—LYSE

Analysis is a critical part of an autonomous control system. Automated analysis—performed immediately after every shot—is often restricted to routines that change infrequently and are applied uniformly once per shot. Ideally analysis should be flexible as well as autonomous; these can be conflicting goals without a unifying analysis framework. Our analysis system `lyse` accommodates collective analysis of a group of shots and trivial re-analysis upon changing or adding routines.

`lyse` is a scheduler for user-written analysis routines, which are ordinary Python scripts. It provides functions for extracting the experiment data and metadata from the HDF files and saving analysis results to these files. Multiple analysis routines added to `lyse` execute one after the other when a new HDF file is received over the network, or on command through the GUI. Plots produced by the user’s code are updated following every shot as new data comes in from the experiment.

There are two types of routine that `lyse` can run: single-shot, which are run on every shot, and multi-shot, which analyze a group of shots together. Analysis of the thermometry example in Sec. 5.5 is shown in Fig. 5.7. A single-shot routine computes the size of an atom cloud after a fixed expansion time, and a multi-shot routine uses these results to determine the expansion rate and thus the temperature. The multi-shot routine then plots this temperature as a function of laser detuning and magnetic field strength.

Splitting, sorting, plotting, and exploring large multidimensional datasets are cumbersome when directly accessing a set of files. In addition to direct access to the HDF files, `lyse` provides a tabular data structure—a `pandas` [83] DataFrame—for multi-shot routines, containing all globals as set by `runmanager`, and all single-shot analysis results. With `pandas` and the standard Python scientific stack of `numpy` [84], `scipy` [85], and `matplotlib` [86], `lyse` provides a powerful environment for analysis [87].

Analysis routines can be run independently of `lyse` if desired. This allows the same framework and analysis code to be used for publication preparation.

5.9 Optimization—MISE

Marrying powerful Python tools to shot-based analysis permits extensibility of the control system, such as closed loop optimization of measured quantities. One often performs parameter space scans for optimization, requiring many shots. This may be tuning a parameter of an apparatus to enhance its performance, finding a resonance of some transition, or some other feature of interest. The quantity being optimized is often the result of some analysis, e.g., the temperature of ultracold atoms (mentioned in Secs. 5.5 and 5.8). We have created `mise`, a program that performs automatic optimization of analysis results using a genetic algorithm [88]. A user specifies one or more parameters to optimize

```
rev:    25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date:   Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

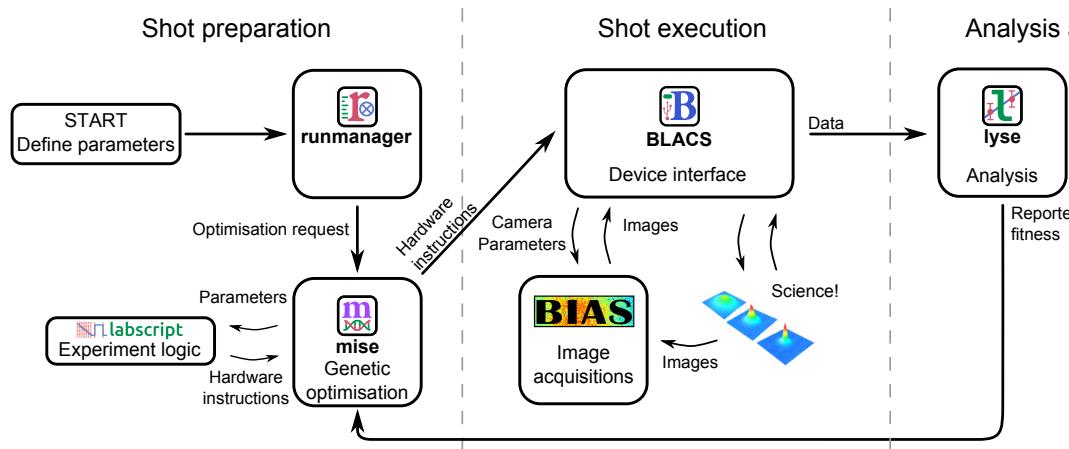


Figure 5.8: The data flow for closed loop optimization. In contrast to Fig. 5.1, analysis results are used to determine future shots automatically. The optimizer `mise` varies parameters, directly calling `labscript` to compile new experiment shots. Parameters to be optimized are selected by the user in `runmanager`. `lyse` reports fitness to `mise` which is used to create the next generation of shots.

against a predefined figure of merit. Genetic algorithms are resistant to noise, making them particularly useful for optimizing experimental results.

The data flow of the optimization process follows Fig. 5.8, modifying that shown in Fig. 5.1. The user specifies in `runmanager` one or more parameters to optimize, with upper and lower limits for each. An analysis routine in `lyse` reports optimality to `mise`, which creates shots with modified parameters and submits them to `BLACS`.

For each parameter being optimized the user also specifies a *mutation rate*. This determines how much the parameter is varied per generation of the genetic algorithm: the larger the mutation rate, the faster `mise` will move towards the optimum. However, a large mutation rate limits the precision to which the optimal parameters can be determined.

With this specification of parameters, `mise` creates a population of *individuals*. Each individual comprises values from one point in the optimization parameter space, initially chosen at random. An individual may be a single experiment shot, or—when optimizing the result of a multi-shot analysis—a sequence of shots. Once the shots comprising an individual have executed, the user’s analysis routine computes a *fitness*, which may be derived from any measured quantity. `mise` uses the reported fitness in the genetic algorithm to optimize the specified parameters. The genetic algorithm used by `mise`³ is a variation on pointed directed mutation [89], in which mutations are biased in directions previously shown to be successful.

The user can specify when to stop the optimization, either by manual intervention or by a convergence condition written into their analysis script. They may also “guide” the evolution by adding and deleting individuals from the gene pool at any time.

An example of automated optimization using `mise` is shown in Fig. 5.9. By preferentially exploring the more interesting regions of parameter space, autonomous optimization allows optima to be found in fewer shots.

`mise` uses the `labscript` software library to create HDF shot files and submit them to `BLACS`. Additional user-written components could similarly submit shots to `BLACS` if more complex programmatic generation of shots is required.

```

rev:      25 (79ce3eb23e1f)
author:   Chris Billington <chrisjbillington@gmail.com>
date:     Tue Sep 06 23:02:15 2016 +1000
summary:  More work on continuous DOF part of numerics chapter

```

³See supplementary material at [URL will be inserted by AIP] for implementation details of the genetic algorithm used by `mise`

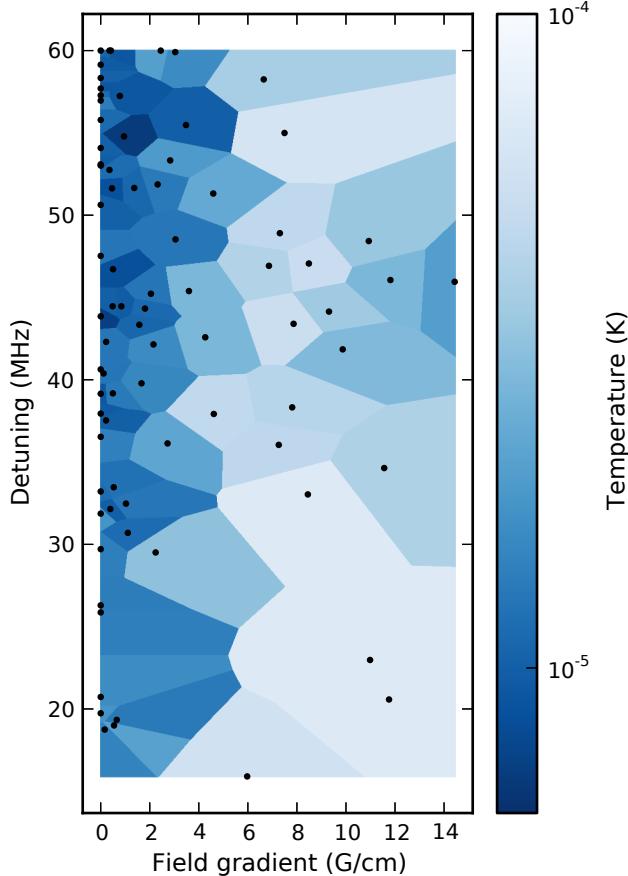


Figure 5.9: A proof-of-principle optimization using `mise`. `mise` scanned the parameter space described in Sec. 5.5, searching for the coldest point. Each black point represents a temperature measurement at a specific field gradient and detuning, with the surrounding shading indicating the temperature. Eighty points were taken, corresponding to 400 shots. The colder region of parameter space is sampled more densely than the uniformly-sampled scan shown in Fig. 5.7(e), with 500 shots.

5.10 Portability and Extensibility

Our software runs on Windows, Linux, and OS X, although `BLACS` and `BIAS` compatibility is subject to the availability of appropriate hardware drivers. If particular devices must be interfaced with a specific computer, operating system, or programming language, a secondary control program (such as `BIAS`, Sec. 5.7) can be used. The components of the labscript suite communicate with each other via data in HDF files, and over the network with ZeroMQ sockets. The widespread support of these technologies across many platforms⁴ ensures users are not bound to any one operating system or programming language. The modular nature of our system allows users to replace or supplement any of our programs in their choice of language.

The programs themselves are also written with extensibility in mind. Adding new hardware support to the labscript suite entails writing a new device class for `labscript`, and a GUI tab for `BLACS`,⁵ or a camera class for `BIAS`. Adding analysis routines to `lyse` amounts to writing a Python script to process experiment data. Existing library functions

⁴ HDF bindings include C/C++, MATLAB, Python, LabVIEW and Mathematica. ZeroMQ support includes C/C++, Python, LabVIEW, Java and many more. See http://www.hdfgroup.org/products/hdf5_tools/ and http://www.zeromq.org/bindings:_start/ for more complete lists.

⁵BLACS communicates with hardware devices through user-written interface code. Devices communicating over standard buses (RS232, USB, Ethernet) are easily interfaced using standard Python libraries for these buses. Devices with proprietary interfaces can be programmed by calls to vendor-supplied libraries through Python’s sophisticated foreign-function interface.

and base classes assist such development. The suite has already proved useful in a setting distinct from quantum science experiments, automating the prototyping of an objective lens, in which the image of a pinhole was acquired and analyzed at 3600 points in a plane to determine the field of view [90].

The labscript suite is open-source and freely available online [91]. We encourage readers to contact us if they are interested in implementing the suite in their laboratory.

```
rev:    25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date:   Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

Particle velocimetry of vortices in Bose–Einstein condensates

- Semiclassical simulations: Provide details of how the simulations were performed and the results they gave. Describe method of computing collisions between the classically modelled particles and the condensate, and how the fluorescence imaging was simulated. Present results and discuss experimental feasibility.
- Sisyphus cooling in a magnetic field Describe the scheme for Sisyphus cooling I developed, and show 1D simulation results.

6.0.1 Method

The main experimental aim of this project is the use of tracer particles to track vortex cores in a BEC in real time. The tracer particles will be ^{87}Rb atoms and the BEC made of ^{41}K . This choice is due to the strong interspecies repulsion between these atoms, which gives rise to the trapping of atoms in the vortex cores. In the limit of low densities and temperatures, such that three body collisions are suppressed and s-wave scattering dominates the interspecies interactions [92, p 120], The rubidium atoms experience a potential due to the potassium:

$$V(\mathbf{r}) = \frac{2\pi\hbar^2 a_s}{m_r} \rho_{\text{K}}(\mathbf{r}), \quad (6.1)$$

where $\rho_{\text{K}}(\mathbf{r})$ is the spatially varying atom density of the potassium condensate, a_s is the interspecies s-wave scattering length, and $m_r = \frac{m_{\text{K}} m_{\text{Rb}}}{m_{\text{K}} + m_{\text{Rb}}}$ is the reduced mass of the scattering pair.

Vortex cores thus create potential wells for other atoms, since they are regions of low condensate density in a background of high density.

The basic setup of this experiment is shown in Figure 6.1. A potassium condensate will be formed, and then have cold rubidium atoms introduced to it (likely magnetically transported from a MOT). Both species will then be trapped at the focus of a high power 1064nm laser, using the dipole force.

Various methods will be used to create vortices in the condensate. These include bluff-body flow, where a repulsive potential is dragged through the condensate, and inducing a turbulent state by applying off resonant laser speckle. The rubidium atoms are then expected to become trapped in the low density vortex cores.

The atoms will be imaged with resonant or near-resonant laser light, depending on the exact scheme employed. We will attempt several imaging schemes, some of which involve the laser light also cooling the atoms to keep them trapped in the vortex cores.

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

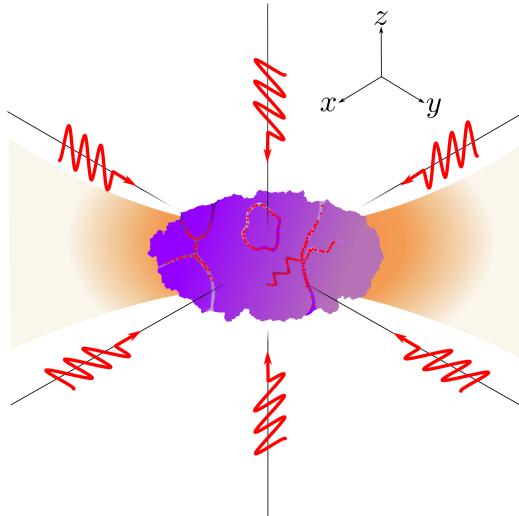


Figure 6.1: The simplest scheme for cooling and imaging the tracer particles with the same light is polarisation gradient cooling, involving six slightly off resonant beams (red), with each counterpropagating pair having opposite linear polarisations. This will scatter some light off the tracer atoms, as well as cool them to sub-Doppler temperatures. If the cooling is sufficient, it should encourage the atoms into the vortex cores where their energy is lower, if they aren't already there. Both the rubidium tracers and the potassium BEC will be trapped with approximately the same trapping potential by a strong, far off-resonant laser (orange), via the dipole force. Magnetic trapping cannot be used, as polarisation gradient cooling does not work in the presence of a magnetic field.

The simplest imaging scheme involves only imaging a small amount of the tracer atoms at a time, by transferring population slowly from the $|1, 1\rangle$ groundstate into an $F = 2$ state where a resonant laser images them and likely removes them from the trap.

The simplest scheme which attempts to cool the rubidium atoms is ordinary polarisation gradient cooling, where the same light is used for imaging and cooling the atoms (Figure 6.1). This method precludes the use of a magnetic trap or large bias field, since either would destroy the cooling effect.

If this does not cool the atoms enough to keep them within the vortex cores, then a Feshbach resonance will be used to increase the interspecies scattering length, and alternative cooling schemes—which work in the presence of a magnetic field—will be investigated (see section 6.1).

We aim to be able to scatter 10^5 photons per second off each rubidium atom without it escaping its vortex core trap, and without causing so much heating as to destroy the condensate on a reasonable experimental timescale.

A high resolution, low aberration lens (numerical aperture ≈ 0.5) will focus the scattered light onto a fast capture, high quantum efficiency camera to produce images of vortex motion.

We will attempt the experiment first in a two-dimensional BEC, and attempt to observe an inverse cascade, which is predicted to occur if a small repulsive potential hill is added to the middle of the trapping potential of the condensate. Tracking vortex motion in two dimensions should also be much simpler due to the fact that vortices in 2D are pointlike rather than extended.

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

6.1 Simulations of Sisyphus cooling

ONE OF THE PROBLEMS with polarisation gradient cooling is that it doesn't work in a magnetic field. Usually this isn't an issue for the cooling stage used en-route to BEC; the magnetic field is simply temporarily switched off. Our imaging method would greatly benefit from a cooling method that did work in a magnetic field, since the repulsive interactions between ^{87}Rb and ^{41}K can be greatly enhanced via a Feshbach resonance at 34 gauss [47]. This would make the potential wells that the rubidium atoms see deeper, trapping them more strongly. However if the magnetic field destroys our cooling method then the atoms won't stay trapped for long.

This Feshbach resonance only occurs if both species are in their respective $|F = 1, m_F = 1\rangle$ spin state¹, so we would like a cooling method that has the rubidium atoms spending a significant amount of time in this state. Polarisation gradient cooling isn't particularly efficient when using this groundstate, due to the high probability of atoms in the $F = 2$ excited state decaying to the $F = 2$ groundstate, requiring repumping.

One possibility is to develop a sub-Doppler cooling scheme that works in a 34 G magnetic field. The basic Sisyphus mechanism—of atoms moving alternately between spin states which see different potentials—should be possible to find in many multi-level systems of sufficient complexity². Below I describe one that uses four lasers to cool ^{87}Rb in a 34 G field, with the atoms spending approximately half their time in the $|1, 1\rangle$ state.

Incidentally, my initial misunderstanding of polarisation gradient cooling coupled with a misplaced negative sign in my calculations—which hid that misunderstanding—led me to construct a scheme qualitatively different from conventional PGC. This scheme uses blue detuned light, and the main cooling force comes from atoms ascending repulsive optical potential hills, rather than climbing out of attractive potential wells as in ordinary PGC.

In section 6.1.4, I describe another cooling scheme, recently suggested by Kris Helmer-son, which uses the vortex cores themselves as the potential hills in a Sisyphus mechanism.

¹ F is not a good quantum number in a nonzero magnetic field, so what we really mean writing this is the state that one would get if starting in an F state and adiabatically ramping the field up (with no coupling between the states—so that crossings are not avoided).

²And indeed, many other Sisyphus cooling mechanisms exists other than polarisation gradient cooling [33, p 116].

6.1.1 Description of cooling scheme

My scheme involves four lasers, two for cooling and two for repumping. First let's focus on the cooling lasers only. Looking at Figure 6.2, imagine that we have a rubidium atom at $z = 0$ and in the $|1, 1\rangle$ hyperfine groundstate. Here our atom sees no light, as the intensity of the cooling laser labeled ω_1 is zero, and it is in the wrong state to be pumped by the ω_2 laser (which is not resonant with any transitions from the $|1, 1\rangle$ groundstate).

As our atom moves rightward however, it will have to climb the repulsive potential hill formed by the the ω_1 laser. As it does so, its $|1, 1\rangle$ excited state probability will increase, and along with it, the probability of spontaneous emission. Spontaneous emission will be most likely to occur near the top of the potential hill where the laser intensity—and hence the excited state probability—is greatest.

The most likely groundstate for the atoms to decay to from the $|1, 1\rangle$ excited state is the $|2, 2\rangle$ groundstate, and this is most likely to occur near $z = \frac{\lambda}{4}$. If this occurs, we now have an atom in the $|2, 2\rangle$ groundstate at $z = \frac{\lambda}{4}$, a situation similar to that in which it started. Again, our atom now sees no light, but which laser has zero intensity and which targets the wrong transition are swapped.

As our atom continues rightward, it now has to contend with the potential hill formed by the ω_2 laser, and is most likely to undergo spontaneous emission from the $|2, 2\rangle$ excited state near the top of the potential hill. This time emission is most likely to put the atom into the $|1, 1\rangle$ groundstate.

This process repeats, with atoms repeatedly climbing potential hills and being cooled. They spend approximately half their time in the $|1, 1\rangle$ groundstate, allowing us to take

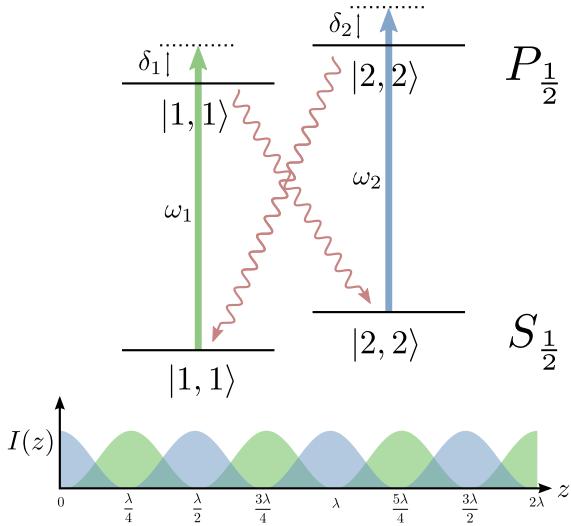


Figure 6.2: An idealised depiction of the cooling scheme, with repump lasers and undesired states not shown. Two lasers on the D_1 line are used for cooling, both linearly polarised, and arranged so as to form two interleaved standing waves. Both are blue detuned from the transitions they target, and they differ by about 6.8 GHz. This difference means that the alignment of the two standing waves can only be maintained over a distance of about a centimetre.

advantage of the strong interspecies repulsion that that state entails for our two atomic species.

Of course, as is always the case, things aren't that simple. Whilst the two spontaneous decays mentioned above are the most likely, they are by no means the only possibilities. Some spontaneous decays will put the atoms back into the groundstate from which they came, with no harm done except a little extra heating from the photon recoil. Other decays however will put our atom into states that are not involved in the cooling scheme, where they will remain with no further cooling unless we do something about it. For this we need repump lasers (Figure 6.3).

There are three states that the atom might end up in as a result of decay from the two excited states involved in the cooling process, and two repump lasers are used to excite them to three $P_{\frac{1}{2}}$ states. Two of these transitions are similar enough that they can be addressed with the same laser.

6.1.2 Methods

³One complex number for each state in Fig. 6.3.

⁴Using the dipole approximation and the rotating wave approximation, following the derivation in [59, p 9].

⁵Each energy eigenstate at nonzero field is a superposition of exactly two of the zero field eigenstates.

This scheme was simulated for the case of a single atom, with the internal state of the atom modelled with the Schrödinger equation in the spin basis, the state vector being a complex 32-vector³. The coupling terms between each pair of states were computed by solving the eigenvalue problem in the spin basis, with the Hamiltonian including Zeeman terms, and projecting the resulting eigenvectors onto the zero field eigenvectors. The zero field eigenvectors have easily computed coupling constants⁴, a weighted sum of which gives the coupling constants for the states at higher field⁵. Since the coupling constants are dependent on the laser intensity, they were re-computed constantly as the atom moved through different intensities of the cooling beams. This process produced a set of 32 coupled differential equations for the complex amplitudes of each state [33, p 4],

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

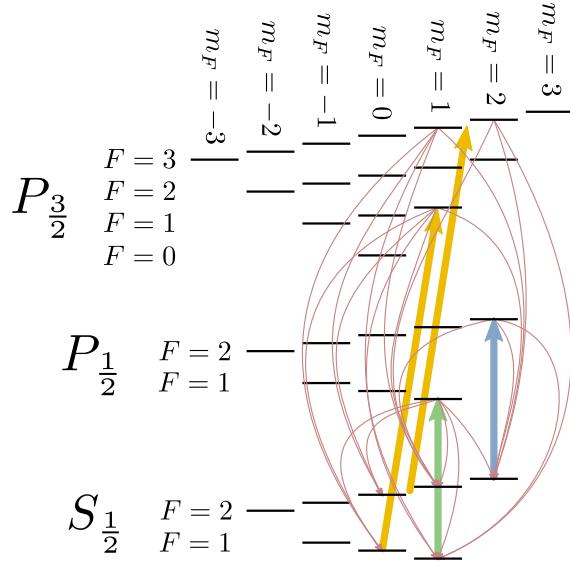


Figure 6.3: The full cooling scheme, including repump lasers (yellow), cooling lasers (blue and green), and all possible decay paths (red). The repump beam which is drawn in between two ground and excited states has a frequency equal to the average of those two transitions.

of the form :

$$i\hbar \frac{dc_e(t)}{dt} = -\frac{1}{2}e \sum_{g, n} E_n \langle g|q_n|e\rangle c_g(t) e^{-i\delta_{nge}t}, \quad (6.2)$$

and

$$i\hbar \frac{dc_g(t)}{dt} = -\frac{1}{2}e \sum_{e, n} E_n \langle g|q_n|e\rangle c_e(t) e^{i\delta_{nge}t}, \quad (6.3)$$

where each $c(t)$ is the complex amplitude of one state; the e indices are over the excited states and the g indices over the groundstates⁶; the n indices are over the lasers, with E_n being the amplitude of the n^{th} laser's electric field, δ_{nge} the detuning of the n^{th} laser from the transition between the g^{th} ground and e^{th} excited states, and $\langle g|q_n|e\rangle$ the dipole moment between the g^{th} ground and e^{th} excited states for the polarisation of the n^{th} laser.

⁶not to be confused with the electron charge or base of the natural logarithm, also used but not as indices.

The external motion of the atom was modelled classically, with the atom having a definite position and velocity in one dimension. The force on the atom was computed using the gradient of the light shift that the two groundstates experience due to the standing waves formed by the cooling beams [33, eqn 3.16, p 33]. The atom's state was projected onto the two groundstates which see the cooling lasers. The resulting force on the atom was used in the classical equation of motion.

Spontaneous emission was handled by at each integration timestep, summing up all the population in $P_{1/2}$ and $P_{3/2}$ excited states, weighted by their decay rates (equal to the natural linewidth). This gave the probability of decay per unit time. Multiplying by the duration of one timestep, and comparing with a random number then determined whether a decay was to occur.

In the event of a decay, one excited state was randomly chosen, weighted by their populations, and then one groundstate, weighted by the transition strengths from the excited state. All population was then put into that groundstate and the simulation

Type	Transition(s) targeted	Detuning	Intensity (per beam)	Polarisation
cooling (standing wave)	$ S_{\frac{1}{2}}, 2, 2\rangle \rightarrow P_{\frac{1}{2}}, 2, 2\rangle$	+ 66.6 MHz	5.0 mW cm ⁻²	π
cooling (standing wave)	$ S_{\frac{1}{2}}, 1, 1\rangle \rightarrow P_{\frac{1}{2}}, 1, 1\rangle$	+ 31.9 MHz	5.0 mW cm ⁻²	π
repump (single beam)	$ S_{\frac{1}{2}}, 2, 1\rangle \rightarrow P_{\frac{3}{2}}, 2, 2\rangle$ $ S_{\frac{1}{2}}, 2, 0\rangle \rightarrow P_{\frac{3}{2}}, 2, 1\rangle$	Midway between	50.0 mW cm ⁻²	σ^+
repump (single beam)	$ S_{\frac{1}{2}}, 1, 0\rangle \rightarrow P_{\frac{3}{2}}, 1, 1\rangle$	0	10.0 mW cm ⁻²	σ^+

Table 6.1: The parameters used in the laser cooling simulations. There are four lasers, each with a specified polarisation, intensity, and detuning from the transition it targets.

continued, with one photon's worth of momentum in a random direction added to the atom's external state to account for photon recoil.

The equations of motion were solved using fourth order Runge–Kutta integration, with the error monitored by verifying that the overall probability summed over all states remained close to unity.

6.1.3 Results

The laser parameters used in the simulation are shown in Table 6.1. The magnetic field strength used was 34 G.

⁷Which is about ten timesteps per oscillation of the fastest oscillating terms, which oscillate at a rate equal to approximately half the 6.8GHz hyperfine splitting of the rubidium groundstates.

The simulation was run for 715 million integration timesteps of 20 picoseconds each⁷, for a total of 14.3 milliseconds of simulation time. This took 14 days of computer time. In that time, the atom moved a maximum distance of 26 micrometres from its starting position, and its final position was 790 nanometres from its starting position. The atom's initial velocity was 195 millimetres per second, and during the simulation it reversed the direction of its velocity 2226 times. 4103 photons were emitted, for an average scattering rate of 2.87×10^5 photons per second.

Computing the time averaged energy of the atom over the whole simulation using:

$$\langle E \rangle = \frac{1}{2} m_{\text{Rb}} \langle v^2 \rangle, \quad (6.4)$$

and converting to temperature units with $k_B T = \langle E \rangle$ gives a temperature of 8.1 μK . Since this is only a one-dimensional simulation, a temperature approximately three times higher would be expected in three dimensions, as the atom would have approximately the same amount of energy in each spatial degree of freedom.

A histogram of what fraction of the time the atom spent at different velocities is show in Figure 6.4.

This one-dimensional temperature corresponds to approximately 4.4 recoil energies, and if extrapolated to three dimensions, about 132 recoils. Given that the potassium vortex potentials are at most about 15 recoils deep without a Feshbach resonance, and only about 8 recoils when you consider that the rubidium is not in the ideal state half of the time, this result will only be able to keep rubidium atoms trapped in vortex cores if we can get a factor of 20 or so increase in the interspecies repulsion via a Feshbach resonance.

This simulation has not, however, been optimised. Whilst some parameters were computed from others based on assumptions about optimal scattering rates and the like, no attempt has been made to scan over parameter space to see if the temperature can be made lower. I plan on using a genetic algorithm to optimise the parameters by managing a population of simulations running on one of the university clusters. A significant speed

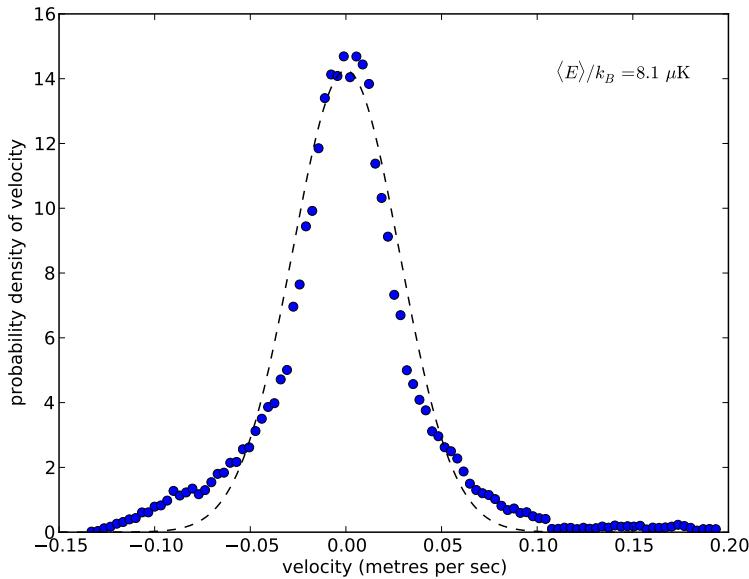


Figure 6.4: Histogram of atom velocity over time, normalised such that it can be interpreted as a probability density. A best-fit Maxwell-Boltzmann distribution is shown as the dotted line. It is no surprise that it is not a good fit—there is no thermalisation happening since we have only one atom and no collisions. The average energy is more informative than the fit parameters for determining the temperature that an ensemble of such atoms would have if they were allowed to thermalise. This is because the average energy would stay constant throughout thermalisation, whereas the fit parameters would not. Only for a fully thermal distribution would the two methods agree. The long tail visible to the right is the atom's initial slowdown from its starting velocity.

up should be possible by excluding from the simulation the atomic states that were shown in the first run never to become occupied. This will eliminate approximately two thirds of the states, and since the simulation is quadratic in the number of states, this should provide an approximately 10× increase in simulation speed.

The simulation will also require repetition to verify that the results still hold when a significant error, recently discovered, is corrected. The error is that the Zeeman sublevels used in the simulation were all incorrect by a sign. There is a large degree of symmetry with this change, and the scattering rates between all involved states are almost identical, so I am confident that this will only slightly change the results.

Ultimately only experimentation will show whether this method is viable, and what the optimal parameters are, but since it requires a large number of lasers, it is likely worth further theoretical investigation before attempting to implement it.

6.1.4 Vortex-assisted Sisyphus cooling

Another idea for a cooling scheme is to use the vortex potential itself as a spatial discriminator for transferring atoms between states. Similar to how a MOT traps atoms by bringing them into resonance with optical pumping only when they are some distance from the trap's centre, we could use the shape of the vortex potential to bring an RF or microwave transition into resonance only when trapped tracer particles are some way up the side of a vortex core.

```
rev:    25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date:   Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter
```

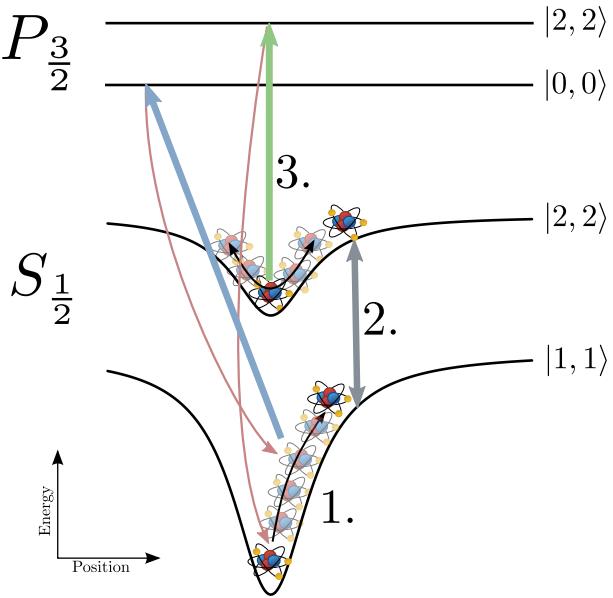


Figure 6.5: A basic description of the vortex-assisted cooling scheme.

1. The rubidium atom in its $|1, 1\rangle$ groundstate repeatedly scatters photons from the laser marked with the blue arrow, climbing the vortex potential as it does so. The optical transition's linewidth is large enough that the energy shift due to the vortex potential does not move it off resonance.
2. An RF or microwave transition however, has an extremely narrow linewidth; its effective linewidth is dependent only on the RF/microwave power. A microwave transition (grey arrow) comes into resonance only when the atom moves sufficiently far from the vortex core's centre, and coherently transfers population into the $|2, 2\rangle$ groundstate.
3. The atom oscillates back and forth in the much shallower vortex potential that its $|2, 2\rangle$ groundstate experiences. It is pumped weakly by the laser marked with the green arrow, and after a random time delay (and hence at a random position) spontaneously decays back to the $|1, 1\rangle$ groundstate.

The basic idea is outlined in Figure 6.5. In the presence of the Feshbach resonance, atoms in the $|1, 1\rangle$ state will scatter some tens of photons, using whichever transition is most likely to have them decay to the same groundstate with minimal repumping (transitioning to the $|0, 0\rangle$ excited state on the D_2 line looks to be the best choice). As the atom scatters photons, it climbs the side of the vortex potential, converting its new found kinetic energy (from photon recoil) into potential energy.

Due to the state-dependence of the interspecies scattering length, the vortex potentials for different states have different depths. This means that the RF or microwave frequency required to transition between the different hyperfine states and Zeeman sublevels varies as a function of space, and can be tuned so as to only be resonant with atoms which have nearly escaped the vortex core.

The atom is then transferred into a different hyperfine or Zeeman state (the $|2, 2\rangle$ groundstate should suit) and the hope is that it then lacks the kinetic energy to escape the (shallower) vortex potential it now finds itself in. Rather, it will oscillate back and forth in the well until a weak laser pumps it back into the $|1, 1\rangle$ groundstate via spontaneous emission from some excited state (again chosen to maximise the decay probability to $|1, 1\rangle$; the $|2, 2\rangle P_{\frac{3}{2}}$ excited state looks to be a good choice.)

If this goes to plan, statistically the atom will be closer to the center of the $|1, 1\rangle$ vortex potential than when it left. Provided its corresponding drop in potential energy makes up for all the photon scattering (which provides fluorescence imaging), then we have a

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

cooling scheme. It is yet another Sisyphus effect, with the atom climbing steep vortex potential hills and descending shallower ones.

This scheme will be simulated to determine its viability; preliminary calculations haven't turned up any problems yet.

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

This page intentionally left blank

Wave mixing in Bose–Einstein condensates

7.1 Off-resonant four wave mixing

Describe the wave mixing experiments and their results, compare with the simulations I performed and provide the simplified three-level model explaining the results.

7.2 Spin wave mixing

Show results of spin wave mixing simulations, including scaling behaviour with increasing c_2 .

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

This page intentionally left blank

CHAPTER 8

Hidden variables for semiclassical models with state-dependent forces

TODO:

- Define/describe what a hidden variable theory is, drawing heavily on Aaronson's [93] explanations. Give examples, argue why the Schrodinger theory is appealing.
- Motivate with Stern-Gerlach experiment, and derive the method, show what sorts of problems it solves and where it disagrees with other models, provide simulation results. Limitations: no time dependent potentials, no 3D.
- Possibly include speculation about these:
 - Maybe include a test to see whether it actually does work in 3D as-is, since we haven't actually checked, we just haven't been able to show on paper that current behavior is correct in 3D (also haven't shown it's incorrect).
 - Time dependent potentials could potentially be handled by approximating unitary as product of part due to spatial variation in H, and part due to time variation in H. compute transition probs for both such that a transition can be attributed to one or the other - only do velocity jumps to conserve potential if due to spatial motion, as time dependent potential can exchange energy with particle.
 - Matrix scaling for Schrodinger theory, discuss methods: Sinkhorn-Knopp, Lineal, and my one. Compare time complexity of algorithms so as to define the computational complexity of the hidden variables semiclassical method. Method is of course parallelisable on GPU or similar so is fast on parallel machines even if matrix scaling is slow.
 - Discuss how it would make sense for the systems to behave in the presence of collisions w.r.t collapse of state vectors.
 - Be sure to include picture from KOALA talk of atoms going in multiple directions

8.1 Approximate Markovian decoherence rate for separating wavepackets

Positional separation of two different internal states of an atom leads to decoherence of those states, with a decoherence factor $r_{ij}(t)$ equal to the overlap of the spatial wavefunc-

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

tions of the two components in question a time t after they began separating. Approximating both wavepackets as initially overlapping Gaussians of width σ , ignoring dispersion, and assuming they separate with constant relative acceleration a_{ij} , the decoherence factor is

$$r_{ij}(t) = \langle \psi_i(t) | \psi_j(t) \rangle \quad (8.1)$$

$$= C \int_{-\infty}^{\infty} e^{-\frac{x^2}{4\sigma^2}} e^{-\frac{(x-x_{\text{rel}})^2}{4\sigma^2} + ik_{\text{rel}}x} dx, \quad (8.2)$$

where

$$x_{\text{rel}}(t) = \frac{1}{2} a_{ij} t^2 \quad (8.3)$$

and

$$k_{\text{rel}}(t) = \frac{m}{\hbar} a_{ij} t \quad (8.4)$$

¹ a_{ij} , x_{rel} and k_{rel} are the acceleration, position, and wavenumber of the j^{th} component with respect to the i^{th} component, that is, $a_{ij} = a_j - a_i$, etc.

are the wavepackets' relative¹ position and wavenumber due to acceleration for a time t starting from zero relative velocity, and

$$C^{-1} = \int_{-\infty}^{\infty} e^{-\frac{x^2}{2\sigma^2}} dx \quad (8.5)$$

is a normalisation constant [TODO CHECK IF NEEDS TO BE SQUARED]. Note that this expression holds for any number of dimensions—relative motion is only along one axis so the integrals in all other directions equal one.

Evaluating the Gaussian integral (8.2) gives the following expression for the decoherence factor $r_{ij}(t)$:

$$r_{ij}(t) = e^{-\left[\frac{1}{8\sigma^2} x_{\text{rel}}^2 + \frac{i}{2} x_{\text{rel}} k_{\text{rel}} + \frac{\sigma^2}{2} k_{\text{rel}}^2\right]}. \quad (8.6)$$

This is a decoherence *factor*; it is the factor by which the (i,j) off-diagonal of the reduced density matrix for the atom's internal state will be reduced at time t . The corresponding decoherence *rate* is given by the logarithmic derivative of (8.6):

$$\Gamma_{ij}(t) = -\frac{1}{r_{ij}(t)} \frac{d}{dt} r_{ij}(t). \quad (8.7)$$

The fact that (8.6) does not describe a constant decoherence rate (i.e., it does not have the functional form of exponential decay) means that the back-action on the atom's internal state caused by measurements of its motional state will be different depending on the interval of time between measurements.

For example, the logarithmic derivative of (8.6) approaches zero as t goes to zero. This means that in the limit of infinitely frequent measurements, no decoherence occurs at all in between measurements, and the motional state is reset after each measurement such that the wavepackets never separate at all. This is the quantum Zeno effect, and its appearance in models of open quantum systems is usually treated as a reminder that the assumption of infinitely frequent strong measurements is unphysical [CITE].

Since experimentally we are not measuring atoms' motional states so frequently, we ought to wait until the wavepackets are completely separated before performing a projective measurement. As in quantum optics models of open quantum systems, in

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

which the measurement interval “should be large enough to allow the photons to get away from the atom” [CITE The Quantum Jump Approach and Quantum Trajectories Gerhard C. Hegerfeldt], ours should be large enough for the atomic states to get away from each other.

If at large enough times, a decoherence rate is independent of time, that decoherence is called Markovian at that timescale. A Markovian environment is one that has no memory of the decoherence process—it “forgets” any information caused by past interaction with the system. Even though at short times, all decoherence rates in quantum mechanics tend to zero [CITE], if they become Markovian on a timescale shorter than other timescales of interest, the Markov approximation can be used and a constant decoherence rate used at all times. In quantum optics, the decoherence factor for the internal state of an atom due to photon emission indeed tends to exponential decay on timescales that are still much shorter than that of the system evolution, and thus the Markov approximation is accurate.

Unlike quantum optics models, our decoherence factor does not describe Markovian decoherence on any timescale. In the limit of large t , its functional form is e^{-t^4} , not the exponential decay required to treat the decoherence as Markovian [CITE] at that timescale. Nonetheless, if we wish to write a time-local differential equation for the internal state of the atom, Markovian decoherence is the only kind we can include [CITE].

To that end, we will now construct a “time ignorant” version of $r_{ij}(t)$ that answers the question “What is the expected decoherence factor at all future times, if you don’t know how long it has been since the two wavepackets began separating?” In this way we can compute an *average* decoherence rate Γ_{ij} described by our decoherence factor, even though $r_{ij}(t)$ does not have a constant decoherence rate at large times. This essentially amounts to finding the best fitting exponential to $r_{ij}(t)$. Whilst this approximation is crude, it is nonetheless an improvement over the Ehrenfest model, which has no decoherence at all (i.e. it has a decoherence rate that is also constant like ours—but equal to zero).

We define the time-ignorant decoherence factor $\tilde{r}_{ij}(t)$ as the overlap of the wavefunction of the i^{th} internal state with a superposition of wavepackets of the j^{th} internal state, with the superposition being over all times in the past the wavepackets began separating:

$$\tilde{r}_{ij}(t) = \langle \psi_i(t) | A \int_{-\infty}^0 |\psi_j(t-t')\rangle dt', \quad (8.8)$$

where A is a normalisation constant such that $\tilde{r}_{ij}(0) = 1$. Since $|\psi_i(t)\rangle$ is time independent (rather, since we can perform our calculations in the frame of reference in which it is stationary), this is:

$$\tilde{r}_{ij}(t) = A \int_{-\infty}^0 r_{ij}(t-t') dt', \quad (8.9)$$

which is simply the convolution of our decoherence factor with a step function which is nonzero at all negative times. Our average decoherence rate Γ_{ij} is then given by the logarithmic derivative of $\tilde{r}_{ij}(t)$ at $t = 0$:

$$\Gamma_{ij} = -\frac{\tilde{r}'_{ij}(0)}{\tilde{r}_{ij}(0)} \quad (8.10)$$

$$= -\frac{\int_0^\infty \tilde{r}'_{ij}(t) dt}{\int_0^\infty \tilde{r}_{ij}(t) dt} \quad (8.11)$$

$$\Rightarrow \Gamma_{ij}^{-1} = \int_0^\infty e^{-\left[\frac{1}{8\sigma^2}x_{\text{rel}}^2 + \frac{i}{2}x_{\text{rel}}k_{\text{rel}} + \frac{\sigma^2}{2}k_{\text{rel}}^2\right]} dt. \quad (8.12)$$

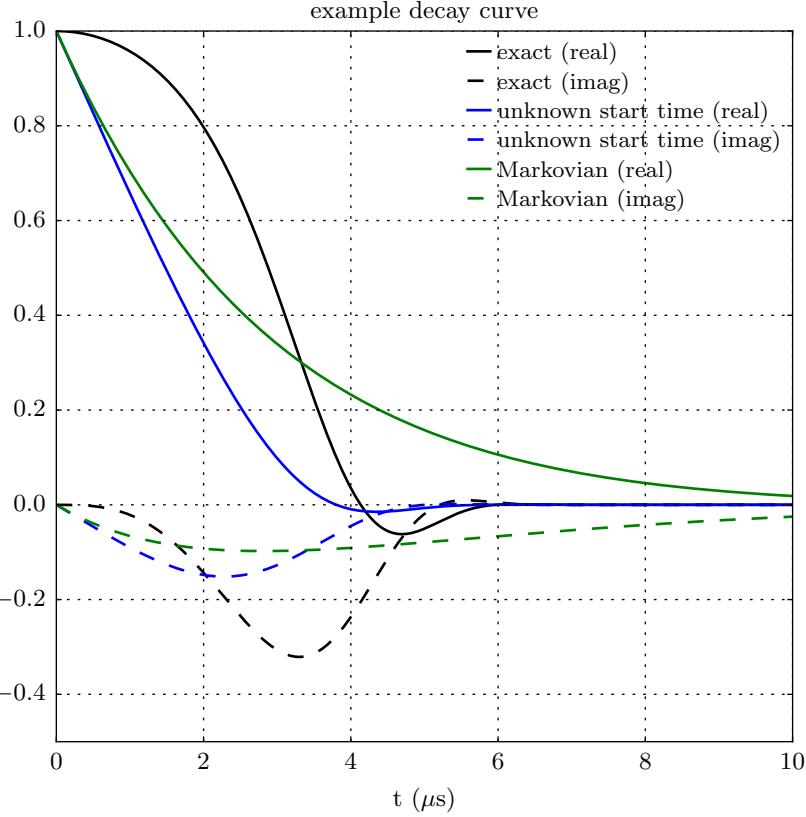


Figure 8.1: Caption.

As mentioned, Γ_{ij} is the decay constant for the best fitting exponential to our decoherence factor (8.6). Although (8.6) looks nothing like a decaying exponential in time, an exponential approximation to it nonetheless ought to decay to zero on the same timescale. An example of this is shown in Figure 8.1

In order to obtain an approximate analytic expression for this integral, we consider two limiting cases and then stitch them together in the intermediate regime. In the limit of small wavepackets, σ is small and thus the first term in the exponent in (8.12) is largest, and the third term is smallest. In this regime, which describes when positional separation (as opposed to separation in k -space) dominates the decoherence, we'll neglect the third term in the exponent and treat the second term as small relative to the first. This gives us:

$$\Gamma_{ij}^{-1}(\text{pos}) \approx \int_0^\infty e^{-\left[\frac{1}{8\sigma^2}x_{\text{rel}}^2 + \frac{i}{2}x_{\text{rel}}k_{\text{rel}}\right]} dt. \quad (8.13)$$

$$\approx \int_0^\infty e^{-\frac{1}{8\sigma^2}x_{\text{rel}}^2} \left(1 - \frac{i}{2}x_{\text{rel}}k_{\text{rel}}\right) dt. \quad (8.14)$$

$$= 2^{\frac{5}{4}} \Gamma\left(\frac{5}{4}\right) \sqrt{\frac{\sigma}{a_{ij}}} - 2i \frac{m\sigma^2}{\hbar}, \quad (8.15)$$

²This isn't necessary in order to obtain a simple expression for $\Gamma_{ij}(\text{pos})$ —the reciprocal without this approximation is equally simple—but it leaves us with power laws for the real and imaginary parts of $\Gamma_{ij}(\text{pos})$, which are easier to stitch together with those from the large σ regime.

where we used a first-order Taylor expansion of an exponential in (8.14). We similarly use a first order expansion to take the reciprocal of (8.15) (since the second term is much smaller than the first²), and arrive at:

$$\Gamma_{ij}(\text{pos}) \approx \frac{1}{2^{\frac{5}{4}} \Gamma(\frac{5}{4})} \sqrt{\frac{a_{ij}}{\sigma}} + \frac{i}{2\sqrt{2}\Gamma(\frac{5}{4})^2} \frac{m\sigma a_{ij}}{\hbar} \quad (8.16)$$

Similarly for the large σ regime, we neglect the first term in the exponent of (8.12) and consider the second term small relative to the third. This is the regime in which the decrease in overlap of the two wavepackets is dominated by their separation in velocity space. Following the same process as above gives:

$$\Gamma_{ij}^{-1}(\text{vel}) \approx \int_0^\infty e^{-\left[\frac{i}{2}x_{\text{rel}}k_{\text{rel}} + \frac{\sigma^2}{2}k_{\text{rel}}^2\right]} dt. \quad (8.17)$$

$$\approx \int_0^\infty \left(1 - \frac{i}{2}x_{\text{rel}}k_{\text{rel}}\right) e^{-\frac{\sigma^2}{2}k_{\text{rel}}^2} dt \quad (8.18)$$

$$= \sqrt{\frac{\pi}{2}} \frac{\hbar}{m\sigma a_{ij}} - i \frac{\hbar^3}{2m^3\sigma^4 a_{ij}^2} \quad (8.19)$$

$$\Rightarrow \Gamma_{ij}(\text{vel}) \approx \sqrt{\frac{2}{\pi}} \frac{m\sigma a_{ij}}{\hbar} + \frac{i}{\pi} \frac{\hbar}{m\sigma^2} \quad (8.20)$$

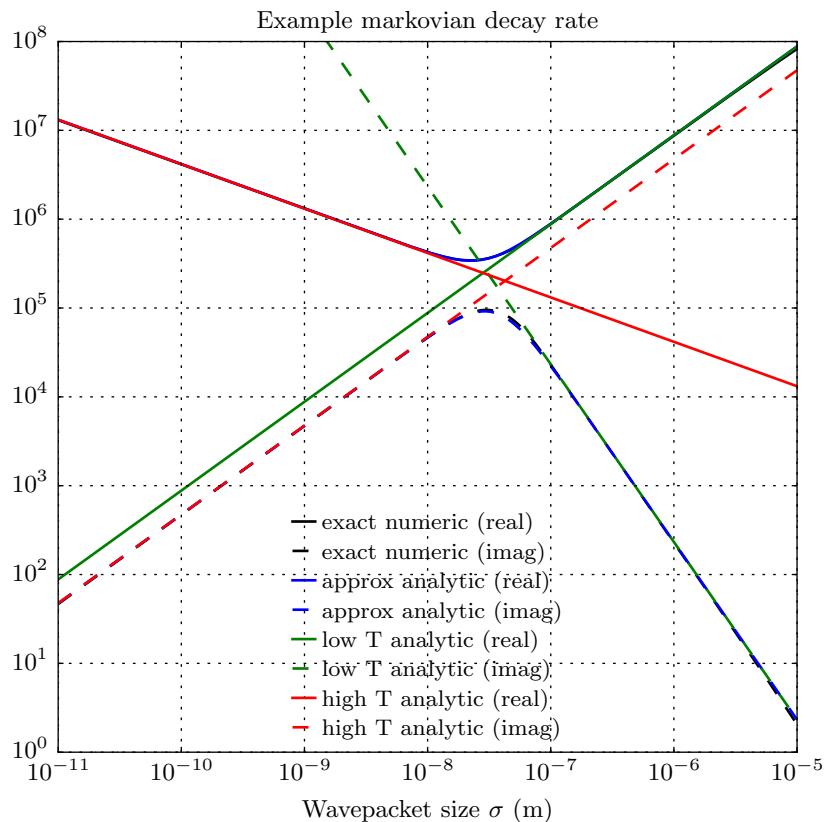
Equations (8.16) and (8.20) are our final expressions for the decoherence rate in the limit of small and large wavepackets respectively. Adding their real parts in quadrature and adding the reciprocals of their imaginary parts then provides a reasonable approximation for Γ_{ij} over all wavepacket sizes:

$$\Gamma_{ij} \approx \left[\text{Re}(\Gamma_{ij}(\text{pos}))^2 + \text{Re}(\Gamma_{ij}(\text{vel}))^2 \right]^{\frac{1}{2}} + i \left[\text{Im}(\Gamma_{ij}(\text{pos}))^{-1} + \text{Im}(\Gamma_{ij}(\text{vel}))^{-1} \right]^{-1}. \quad (8.21)$$

We now have an approximate analytic expression that is computationally inexpensive to evaluate for each atom in an ensemble at every timestep of a differential equation. An example showing the accuracy of (8.21), compared to the exact expression (8.12) for Γ_{ij} over a range of wavepacket sizes is shown in Figure 8.2.

8.2 Choice of wavepacket size

How big is a wavepacket?

**Figure 8.2:** Caption.

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

References

- [1] Satyendra Nath Bose. *Plancks gesetz und lichtquantenhypothese*. Zeitschrift für Physik A Hadrons and Nuclei **26**, 178 (1924). DOI: [10.1007/BF01327326](https://doi.org/10.1007/BF01327326). [p 3]
- [2] A. Einstein. *Quantentheorie des einatomigen idealen gases*. Sitzungsberichte der Preussischen Akademie der Wissenschaften **3** (1925). [p 3]
- [3] M. H Anderson, J. R Ensher, M. R Matthews, C. E Wieman, and E. A Cornell. *Observation of Bose–Einstein condensation in a dilute atomic vapor*. Science **269**, 198 (1995). DOI: [10.1126/science.269.5221.198](https://doi.org/10.1126/science.269.5221.198). [pp 3 and 11]
- [4] K. B. Davis, M. -O. Mewes, M. R. Andrews, N. J. van Druten, D. S. Durfee, D. M. Kurn, and W. Ketterle. *Bose–Einstein condensation in a gas of sodium atoms*. Physical Review Letters **75**, 3969 (1995). DOI: [10.1103/PhysRevLett.75.3969](https://doi.org/10.1103/PhysRevLett.75.3969). [p 3]
- [5] G. Modugno, G. Ferrari, G. Roati, R. J Brecha, A. Simoni, and M. Inguscio. *Bose–Einstein condensation of potassium atoms by sympathetic cooling*. Science **294**, 1320 (2001). DOI: [10.1126/science.1066687](https://doi.org/10.1126/science.1066687). [p 3]
- [6] C. C. Bradley, C. A. Sackett, and R. G. Hulet. *Bose–Einstein condensation of lithium: Observation of limited condensate number*. Physical Review Letters **78**, 985 (1997). DOI: [10.1103/PhysRevLett.78.985](https://doi.org/10.1103/PhysRevLett.78.985). [p 3]
- [7] Tino Weber, Jens Herbig, Michael Mark, Hanns-Christoph Nägerl, and Rudolf Grimm. *Bose–Einstein condensation of cesium*. Science **299**, 232 (2003). DOI: [10.1126/science.1079699](https://doi.org/10.1126/science.1079699). [p 3]
- [8] W. F Vinen. *The detection of single quanta of circulation in liquid helium II*. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences **260**, 218 (1961). DOI: [10.1098/rspa.1961.0029](https://doi.org/10.1098/rspa.1961.0029). [p 4]
- [9] M. R. Matthews, B. P. Anderson, P. C. Haljan, D. S. Hall, C. E. Wieman, and E. A. Cornell. *Vortices in a Bose–Einstein condensate*. Physical Review Letters **83**, 2498 (1999). DOI: [10.1103/PhysRevLett.83.2498](https://doi.org/10.1103/PhysRevLett.83.2498). [p 4]
- [10] Carlo F. Barenghi, Russell J. Donnelly, and W. F. Vinen. *Quantized vortex dynamics and superfluid turbulence*. Springer (2001). [p 4]
- [11] Gregory P. Bewley. *The generation of particles to observe quantized vortex dynamics in superfluid helium*. Cryogenics **49**, 549 (2009). DOI: [10.1016/j.cryogenics.2008.10.018](https://doi.org/10.1016/j.cryogenics.2008.10.018). [p 4]

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

- [12] Gregory P. Bewley, Daniel P. Lathrop, and Katepalli R. Sreenivasan. *Superfluid helium: Visualization of quantized vortices.* Nature **441**, 588 (2006). DOI: [10.1038/441588a](https://doi.org/10.1038/441588a). [p 4]
- [13] Richard E. Packard. *Vortex photography in liquid helium.* Physica B+C **109–110**, 1474 (1982). DOI: [10.1016/0378-4363\(82\)90171-1](https://doi.org/10.1016/0378-4363(82)90171-1). [p 4]
- [14] Waseem S. Bakr, Jonathon I. Gillen, Amy Peng, Simon Fölling, and Markus Greiner. *A quantum gas microscope for detecting single atoms in a Hubbard-regime optical lattice.* Nature **462**, 74 (2009). DOI: [10.1038/nature08482](https://doi.org/10.1038/nature08482). [p 4]
- [15] Chris J Billington. *Particle velocimetry of vortices in Bose–Einstein condensates.* Honours thesis, Monash University (2010). [p 4]
- [16] B. P. Anderson, P. C. Haljan, C. A. Regal, D. L. Feder, L. A. Collins, C. W. Clark, and E. A. Cornell. *Watching dark solitons decay into vortex rings in a Bose–Einstein condensate.* Physical Review Letters **86**, 2926 (2001). DOI: [10.1103/PhysRevLett.86.2926](https://doi.org/10.1103/PhysRevLett.86.2926). [p 4]
- [17] V. Bretin, P. Rosenbusch, F. Chevy, G. V. Shlyapnikov, and J. Dalibard. *Quadrupole oscillation of a single-vortex Bose–Einstein condensate: Evidence for Kelvin modes.* Physical Review Letters **90**, 100403 (2003). DOI: [10.1103/PhysRevLett.90.100403](https://doi.org/10.1103/PhysRevLett.90.100403). [p 4]
- [18] M. Leadbeater, T. Winiecki, D. C. Samuels, C. F. Barenghi, and C. S. Adams. *Sound emission due to superfluid vortex reconnections.* Physical Review Letters **86**, 1410 (2001). DOI: [10.1103/PhysRevLett.86.1410](https://doi.org/10.1103/PhysRevLett.86.1410). [p 4]
- [19] Chad N. Weiler, Tyler W. Neely, David R. Scherer, Ashton S. Bradley, Matthew J. Davis, and Brian P. Anderson. *Spontaneous vortices in the formation of Bose–Einstein condensates.* Nature **455**, 948 (2008). DOI: [10.1038/nature07334](https://doi.org/10.1038/nature07334). [p 4]
- [20] D. V Freilich, D. M Bianchi, A. M Kaufman, T. K Langin, and D. S Hall. *Real-time dynamics of single vortex lines and vortex dipoles in a Bose–Einstein condensate.* Science **329**, 1182 (2010). DOI: [10.1126/science.1191224](https://doi.org/10.1126/science.1191224). [p 4]
- [21] Stephen W Hawking. *A Brief History of Time: From the Big Bang to Black Holes.* Bantam Books (1988). [p 5]
- [22] S. Corrsin. *Turbulent flow.* American Scientist **49**, 300 (1961). [p 5]
- [23] Peter Alan Davidson. *Turbulence: An Introduction for Scientists and Engineers.* Oxford University Press (2004). [pp 5 and 6]
- [24] A. Kolmogorov. *The local structure of turbulence in incompressible viscous fluid for very large reynolds' numbers.* Akademiiia Nauk SSSR Doklady **30**, 301 (1941). [p 5]
- [25] D. Brian Spalding. *Kolmogorov's two-equation model of turbulence.* Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences **434**, 211 (1991). DOI: [10.1098/rspa.1991.0089](https://doi.org/10.1098/rspa.1991.0089). [p 5]
- [26] Lewis Fry Richardson. *Weather Prediction by Numerical Process.* Cambridge University Press (2007). (First published 1922). [p 5]
- [27] Makoto Tsubota, Michikazu Kobayashi, M. Tsubota, and W.P. Halperin. *Energy spectra of quantum turbulence.* In *Progress in Low Temperature Physics: Quantum Turbulence*, volume 16, pages 1–43. Elsevier (2009). [p 5]

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

- [28] W F Vinen. *How is turbulent energy dissipated in a superfluid?* Journal of Physics: Condensed Matter **17**, S3231 (2005). DOI: [10.1088/0953-8984/17/45/006](https://doi.org/10.1088/0953-8984/17/45/006). [p 5]
- [29] L. Onsager. *Statistical hydrodynamics.* Il Nuovo Cimento (1943-1954) **6**, 279 (1949). DOI: [10.1007/BF02780991](https://doi.org/10.1007/BF02780991). [p 5]
- [30] Robert H Kraichnan. *Inertial ranges in two-dimensional turbulence.* Physics of Fluids **10**, 1417 (1967). DOI: [doi:10.1063/1.1762301](https://doi.org/10.1063/1.1762301). [p 5]
- [31] A. J. Leggett. *Superfluidity.* Reviews of Modern Physics **71**, S318 (1999). DOI: [10.1103/RevModPhys.71.S318](https://doi.org/10.1103/RevModPhys.71.S318). [p 6]
- [32] D. J. Wineland, R. E. Drullinger, and F. L. Walls. *Radiation-pressure cooling of bound resonant absorbers.* Physical Review Letters **40**, 1639 (1978). DOI: [10.1103/PhysRevLett.40.1639](https://doi.org/10.1103/PhysRevLett.40.1639). [p 10]
- [33] Harold J. Metcalf and Peter Van der Straten. *Laser cooling and trapping.* Springer (1999). [pp 10, 57, 58, and 59]
- [34] E. L. Raab, M. Prentiss, Alex Cable, Steven Chu, and D. E. Pritchard. *Trapping of neutral sodium atoms with radiation pressure.* Physical Review Letters **59**, 2631 (1987). DOI: [10.1103/PhysRevLett.59.2631](https://doi.org/10.1103/PhysRevLett.59.2631). [p 10]
- [35] P. Zeeman. *On the influence of magnetism on the nature of the light emitted by a substance.* Philosophical Magazine Series 5 **43**, 226 (1897). [p 10]
- [36] D. M. Brink and C. V. Sukumar. *Majorana spin-flip transitions in a magnetic trap.* Physical Review A **74**, 035401 (2006). DOI: [10.1103/PhysRevA.74.035401](https://doi.org/10.1103/PhysRevA.74.035401). [pp 10 and 11]
- [37] A. Ashkin. *Acceleration and trapping of particles by radiation pressure.* Physical Review Letters **24**, 156 (1970). DOI: [10.1103/PhysRevLett.24.156](https://doi.org/10.1103/PhysRevLett.24.156). [p 11]
- [38] J. Dalibard and C. Cohen-Tannoudji. *Laser cooling below the Doppler limit by polarization gradients: simple theoretical models.* Journal of the Optical Society of America B **6**, 2023 (1989). DOI: [10.1364/JOSAB.6.002023](https://doi.org/10.1364/JOSAB.6.002023). [p 11]
- [39] P. J. Ungar, D. S. Weiss, E. Riis, and Steven Chu. *Optical molasses and multi-level atoms: theory.* Journal of the Optical Society of America B **6**, 2058 (1989). DOI: [10.1364/JOSAB.6.002058](https://doi.org/10.1364/JOSAB.6.002058). [p 11]
- [40] P. D. Lett, W. D. Phillips, S. L. Rolston, C. E. Tanner, R. N. Watts, and C. I. Westbrook. *Optical molasses.* Journal of the Optical Society of America B **6**, 2084 (1989). DOI: [10.1364/JOSAB.6.002084](https://doi.org/10.1364/JOSAB.6.002084). [p 11]
- [41] C Salomon, J Dalibard, W. D Phillips, A Clairon, and S Guellati. *Laser cooling of cesium atoms below 3 μK.* Europhysics Letters (EPL) **12**, 683 (1990). DOI: [10.1209/0295-5075/12/8/003](https://doi.org/10.1209/0295-5075/12/8/003). [p 11]
- [42] Harald F. Hess. *Evaporative cooling of magnetically trapped and compressed spin-polarized hydrogen.* Physical Review B **34**, 3476 (1986). DOI: [10.1103/PhysRevB.34.3476](https://doi.org/10.1103/PhysRevB.34.3476). [p 11]
- [43] Y.-J. Lin, A. R. Perry, R. L. Compton, I. B. Spielman, and J. V. Porto. *Rapid production of ⁸⁷Rb Bose-Einstein condensates in a combined magnetic and optical potential.* Physical Review A **79**, 063631 (2009). DOI: [10.1103/PhysRevA.79.063631](https://doi.org/10.1103/PhysRevA.79.063631). [p 11]

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

- [44] E. Madelung. *Quantentheorie in hydrodynamischer form*. Zeitschrift für Physik A Hadrons and Nuclei **40**, 322 (1927). DOI: [10.1007/BF01400372](https://doi.org/10.1007/BF01400372). [p 12]
- [45] Cheng Chin, Rudolf Grimm, Paul Julienne, and Eite Tiesinga. *Feshbach resonances in ultracold gases*. Reviews of Modern Physics **82**, 1225 (2010). DOI: [10.1103/RevModPhys.82.1225](https://doi.org/10.1103/RevModPhys.82.1225). [p 12]
- [46] S. Inouye, M. R. Andrews, J. Stenger, H.-J. Miesner, D. M. Stamper-Kurn, and W. Ketterle. *Observation of Feshbach resonances in a Bose-Einstein condensate*. Nature **392**, 151 (1998). DOI: [10.1038/32354](https://doi.org/10.1038/32354). [p 12]
- [47] G. Thalhammer, G. Barontini, L. De Sarlo, J. Catani, F. Minardi, and M. Inguscio. *Double species Bose-Einstein condensate with tunable interspecies interactions*. Physical Review Letters **100**, 210402 (2008). DOI: [10.1103/PhysRevLett.100.210402](https://doi.org/10.1103/PhysRevLett.100.210402). [pp 13 and 57]
- [48] J. J. Sakurai. *Modern quantum mechanics*. Addison-Wesley Pub. Co, Reading, Mass (1994). [p 26]
- [49] Benjamin M. Caradoc Davies. *Vortex dynamics in Bose-Einstein condensates*. PhD thesis, University of Otago, Dunedin, New Zealand (2000). [p 28]
- [50] W.H. Press. *The art of scientific computing*. Cambridge University Press (1992). [p 28]
- [51] C.J. Pethick and H. Smith. *Bose-Einstein condensation in dilute gases*. Cambridge University Press (2002). [p 28]
- [52] M. L. Chiofalo, S. Succi, and M. P. Tosi. *Ground state of trapped interacting Bose-Einstein condensates by an explicit imaginary-time algorithm*. Phys. Rev. E **62**, 7438 (2000). DOI: [10.1103/PhysRevE.62.7438](https://doi.org/10.1103/PhysRevE.62.7438). [pp 29 and 33]
- [53] G.M. Muslu and H.A. Erbay. *Higher-order split-step Fourier schemes for the generalized nonlinear Schrödinger equation*. Mathematics and Computers in Simulation **7**, 581 (2005). DOI: [10.1016/j.matcom.2004.08.002](https://doi.org/10.1016/j.matcom.2004.08.002). [p 30]
- [54] Barry I. Schneider, Lee A. Collins, and S. X. Hu. *Parallel solver for the time-dependent linear and nonlinear Schrödinger equation*. Phys. Rev. E **73**, 036708 (2006). DOI: [10.1103/PhysRevE.73.036708](https://doi.org/10.1103/PhysRevE.73.036708). [pp 30 and 34]
- [55] M.A. Heroux, P. Raghavan, and H.D. Simon. *Parallel processing for scientific computing*. Society for Industrial and Applied Mathematics, Philadelphia, PA (2006). [p 30]
- [56] Anshul Gupta and Vipin Kumar. *The scalability of FFT on parallel computers*. IEEE Transactions on Parallel and Distributed Systems **4**, 922 (1993). [p 30]
- [57] P. Gulshani and D. J. Rowe. *Quantum mechanics in rotating frames. I. The impossibility of rigid flow*. Canadian Journal of Physics **56**, 468 (1978). DOI: [10.1139/p78-060](https://doi.org/10.1139/p78-060). [p 33]
- [58] E. G. M. van Kempen, S. J. J. M. F. Kokkelmans, D. J. Heinzen, and B. J. Verhaar. *Inter-isotope determination of ultracold rubidium interactions from three high-precision experiments*. Phys. Rev. Lett. **88**, 093201 (2002). DOI: [10.1103/PhysRevLett.88.093201](https://doi.org/10.1103/PhysRevLett.88.093201). [p 33]
- [59] Daniel A. Steck. Rubidium 87 D line data. <http://steck.us/alkalidata>, (2010). (revision 2.1.4). [pp 38 and 58]

rev: 25 (79ce3eb23e1f)

author: Chris Billington <chrisjbillington@gmail.com>

date: Tue Sep 06 23:02:15 2016 +1000

summary: More work on continuous DOF part of numerics chapter

- [60] Tobias G. Tiecke. Properties of potassium. <http://tobiastiecke.nl/archive/PotassiumProperties.pdf>, (2011). (version 1.02). [p 38]
- [61] Matthias Weidemüller and Claus Zimmermann. *Cold Atoms and Molecules*. Wiley (2009). [pp 41 and 44]
- [62] N.P. Robins, P.A. Altin, J.E. Debs, and J.D. Close. *Atom lasers: Production, properties and prospects for precision inertial measurement*. Phys. Rep. **529**, 265 (2013). [p 42]
- [63] Alexander D. Cronin, Jörg Schmiedmayer, and David E. Pritchard. *Optics and interferometry with atoms and molecules*. Rev. Mod. Phys. **81**, 1051 (2009). DOI: [10.1103/RevModPhys.81.1051](https://doi.org/10.1103/RevModPhys.81.1051). [p 42]
- [64] Antonio Negretti, Philipp Treutlein, and Tommaso Calarco. *Quantum computing implementations with neutral particles*. Quantum Inf. Process. **10**, 721 (2011). DOI: [10.1007/S11128-011-0291-5](https://doi.org/10.1007/S11128-011-0291-5). [p 42]
- [65] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien. *Quantum computers*. Nature **464**, 45 (2010). DOI: [10.1038/nature08812](https://doi.org/10.1038/nature08812). [p 42]
- [66] Immanuel Bloch, Jean Dalibard, and Sylvain Nascimbène. *Quantum simulations with ultracold quantum gases*. Nat. Phys. **8**, 267 (2012). DOI: [10.1038/nphys2259](https://doi.org/10.1038/nphys2259). [p 42]
- [67] R. Blatt and C. F. Roos. *Quantum simulations with trapped ions*. Nat. Phys. **8**, 277 (2012). DOI: [10.1038/nphys2252](https://doi.org/10.1038/nphys2252). [p 42]
- [68] G. Varoquaux. *Agile computer control of a complex experiment*. Comput. Sci. Eng. **10**, 55 (2008). DOI: [10.1109/MCSE.2008.47](https://doi.org/10.1109/MCSE.2008.47). [pp 42 and 46]
- [69] Peter E. Gaskell, Jeremy J. Thorn, Sequoia Alba, and Daniel A. Steck. *An open-source, extensible system for laboratory timing and control*. Rev. Sci. Instrum. **80**, 115103 (2009). DOI: [doi:10.1063/1.3250825](https://doi.org/10.1063/1.3250825). [p 42]
- [70] R. P. Anderson. PhD thesis, Swinburne University of Technology (2010). [p 42]
- [71] Matthew Beeler. PhD thesis, University of Maryland (2011). [pp 42 and 44]
- [72] P. A. Altin. PhD thesis, Australian National University (2012). [pp 42 and 44]
- [73] Thilo Stöferle. PhD thesis, Swiss Federal Institute of Technology, Zurich (2005). [p 42]
- [74] Aviv Keshet and Wolfgang Ketterle. *A distributed, graphical user interface based, computer control system for atomic physics experiments*. Rev. Sci. Instrum. **84**, 015105 (2013). DOI: [10.1063/1.4773536](https://doi.org/10.1063/1.4773536). [pp 42, 44, and 45]
- [75] Scott F. Owen and David S. Hall. *Fast line-based experiment timing system for LabVIEW*. Rev. Sci. Instrum. **75**, 259 (2004). DOI: [10.1063/1.1630833](https://doi.org/10.1063/1.1630833). [pp 42, 44, and 45]
- [76] Todd Meyrath and Florian Schreck. A laboratory control system for cold atom experiments. <http://iqoqi.at/control>, (2012). [pp 42 and 45]
- [77] Pieter Hintjens. *Code Connected Volume 1: Learning ZeroMQ*. CreateSpace Independent Publishing Platform (2013). [p 42]
- [78] The HDF Group. Hierarchical data format version 5. <http://www.hdfgroup.org/HDF5>, (2000-2010). [p 42]

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

- [79] IEEE Standard Codes, Formats, Protocols, and Command Commands for Use With IEEE Std 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation. IEEE Std 488.2-1992. [p 43]
- [80] Guido van Rossum et al. Python programming language v2.7. <http://docs.python.org/2.7/>, (2010–). [p 43]
- [81] John M. Hughes. *Real World Instrumentation with Python: Automated Data Acquisition and Control Systems*. O'Reilly Media, Inc. (2010). [p 43]
- [82] Adam Barth, Collin Jackson, Charles Reis, and The Google Chrome Team. The security architecture of the chromium browser, (2008). Technical report, Stanford Security Laboratory, 353 Serra Mall, Stanford, California 94305. [p 49]
- [83] W. McKinney. pandas: a Python data analysis library. <http://pandas.pydata.org/>. [p 51]
- [84] T. Oliphant. NumPy: numerical Python. <http://www.numpy.org/>. [p 51]
- [85] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: open source scientific tools for Python. <http://www.scipy.org/>, (2001–). [p 51]
- [86] J.D. Hunter. *Matplotlib: A 2D graphics environment*. Comput. Sci. Eng. **9**, 90 (2007). DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55). [p 51]
- [87] Wes McKinney. *Python for Data Analysis*. O'Reilly Media, Inc. (2012). [p 51]
- [88] Thomas Bäck and Hans-Paul Schwefel. *An overview of evolutionary algorithms for parameter optimization*. Evol. Comput. **1**, 1–23 (1993). DOI: [10.1162/evco.1993.1.1.1](https://doi.org/10.1162/evco.1993.1.1.1). [p 51]
- [89] A Berry and P Vamplew. *PoD can mutate: A simple dynamic directed mutation approach for genetic algorithms*. In *AISAT 2004: The 2nd International Conference on Artificial Intelligence in Science and Technology*, pages 200–205, Hobart, Tasmania, Australia (2004). [p 52]
- [90] L. M. Bennie, P. T. Starkey, M. Jasperse, C. J. Billington, R. P. Anderson, and L. D. Turner. *A versatile high resolution objective for imaging quantum gases*. Opt. Express **21**, 9011 (2013). DOI: [10.1364/OE.21.009011](https://doi.org/10.1364/OE.21.009011). [p 54]
- [91] The labscript suite: an open source experiment control and analysis system. <http://labscriptsuite.org/>. [p 54]
- [92] Anthony J. Leggett. *Quantum liquids: Bose condensation and Cooper pairing in condensed-matter systems*. Oxford University Press (2006). [p 55]
- [93] Scott Aaronson. *Quantum computing and hidden variables*. Phys. Rev. A **71**, 032325 (2005). DOI: [10.1103/PhysRevA.71.032325](https://doi.org/10.1103/PhysRevA.71.032325). [p 67]

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter

Word count

Total

Words in text: 21311
Words in headers: 265
Words outside text (captions, etc.): 3345
Number of headers: 69
Number of floats/tables/figures: 26
Number of math inlines: 402
Number of math displayed: 73
Files: 9

Subcounts:

text+headers+captions (#headers/#floats/#inlines/#displayed)
1753+28+325 (9/2/26/8) File(s) total: atomic_physics.tex
725+17+385 (4/4/7/0) File(s) total: experiment.tex
50+0+0 (0/0/0/0) File(s) total: front_matter.tex
1370+19+65 (3/2/39/13) File(s) total: hidden_variables.tex
2043+7+145 (3/1/4/0) File(s) total: introduction.tex
8182+126+975 (29/2/248/48) File(s) total: numerics.tex
4435+33+760 (11/9/1/0) File(s) total: software.tex
2717+22+690 (7/6/77/4) File(s) total: velocimetry.tex
36+13+0 (3/0/0/0) File(s) total: wave_mixing.tex

rev: 25 (79ce3eb23e1f)
author: Chris Billington <chrisjbillington@gmail.com>
date: Tue Sep 06 23:02:15 2016 +1000
summary: More work on continuous DOF part of numerics chapter