State Dependent Forces in Cold Quantum Systems

Chris Billington

Supervisors: Prof. Kristian Helmerson, Dr. Lincoln Turner and Dr. Russell Anderson

April 28, 2016

This page intentionally left blank

Contents

Contents i

1 Introduction 1

2 Atomic physics: Experimental techniques and theory 3
- 2.1 Doppler cooling . . . 3
- 2.2 Magneto-optical trapping . . . 3
- 2.3 Optical dipole trapping . . . 3
- 2.4 Two-body scattering and Feshbach resonances . . . 3
- 2.5 Spin, fine structure, and hyperfine structure . . . 3
- 2.6 Equations of motion for two level atom with hyperfine structure . . . 3
- 2.7 The Monte-Carlo wavefunction method . . . 3
- 2.8 Mean field theory for Bose–Einstein condensates . . . 3
  - 2.8.1 Superfluid velocity . . . 3
  - 2.8.2 Vortices . . . 3

3 Quantum mechanics on a computer 5
- 3.1 The interaction picture . . . 5
- 3.2 Discrete degrees of freedom . . . 6
  - 3.2.1 Unitary integration . . . 6
    - 3.2.1.1 Direct exponentiation via diagonalisation of Hamiltonian . . . 6
    - 3.2.1.2 Approximate exponentiation by operator product 6
- 3.3 Continuous degrees of freedom . . . 6
  - 3.3.1 Finite differences . . . 6
  - 3.3.2 The Fourier basis . . . 7
    - 3.3.2.1 Split operator method . . . 7
  - 3.3.3 Harmonic oscillator basis functions . . . 7
- 3.4 Finding groundstates . . . 7
  - 3.4.1 Imaginary time evolution . . . 7
  - 3.4.2 Successive over-relaxation . . . 7
  - 3.4.3 Generalisation to excited states via Gram–Schmidt orthonormalisation . . . 7
- 3.5 The finite-element discrete variable representation . . . 8
- 3.6 Fourth order Runge-Kutta in an instantaneous local interaction picture 8
  - 3.6.1 Algorithm . . . 10
  - 3.6.2 Results . . . 12

4 Development of a cold atom physics experiment 15
- 4.1 Dual species apparatus . . . 15
  - 4.1.1 Vacuum chamber and bakeout . . . 15
- 4.2 Research visit report . . . 15

4.2.1 Optics setup 15

5 Software for experiment control and analysis 17

5.1 labscript 17

5.2 rumananger 17

5.3 BLACS 17

5.4 lyse 17

6 Particle velocimetry of vortices in Bose–Einstein condensates 19

6.1 Semiclassical simulations 19

6.2 Sisyphus cooling in a magnetic field 19

7 Wave mixing in Bose–Einstein condensates 21

7.1 Off-resonant four wave mixing 21

7.2 Spin wave mixing 21

8 Hidden variables for semiclassical models with state-dependent forces 23

Chapter 1

Introduction

[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]

- Brief historical overview of the discovery of Bose–Einstein condensation, laser cooling, turbulence and superfluids and vorticity.

- 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.

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.

2.1 Doppler cooling

2.2 Magneto-optical trapping

2.3 Optical dipole trapping

2.4 Two-body scattering and Feshbach resonances

2.5 Spin, fine structure, and hyperfine structure

2.6 Equations of motion for two level atom with hyperfine structure

2.7 The Monte-Carlo wavefunction method

2.8 Mean field theory for Bose–Einstein condensates

2.8.1 Superfluid velocity

2.8.2 Vortices

This page intentionally left blank

Chapter 3

Quantum mechanics on a computer

- Descriptions of how to perform simulations of atomic systems and Bose–Einstein condensates. Include numerical methods, interaction pictures, Monte–Carlo wavefunction methods and other acquired wisdom that would be relevant to anyone implementing similar simulations.

- Fourier method: usefulness, use with RK4IP, limitations (i.e. first derivs no good near vortices)

Operators and their matrix representation The general case: Need to integrate the equations of motion. What does this mean in a computer? Formal solution of the Schrodinger equation. Is not so formal - can be typed into a programming language thusly: [example code - time ordered product of matrix exponentials].

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.

3.1 The interaction picture

- 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.

3.2 Discrete degrees of freedom

3.2.1 Unitary integration

3.2.1.1 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)

3.2.1.2 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 RKILIP does in the spatial basis. Interaction pictures are really useful!]

3.3 Continuous degrees of freedom

- 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*? Probably not.). Fourier method assume the Fourier series of the wavefunction at the given points can be used to interpolate between points (or the wavefuntion 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.

3.3.1 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.

3.3.2 The Fourier basis

Because of properties of Fourier transforms, derivitives 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 $O(n^2)$)

3.3.2.1 Split operator method

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

3.3.3 Harmonic oscillator basis functions

3.4 Finding groundstates

3.4.1 Imaginary time evolution

3.4.2 Successive over-relaxation

3.4.3 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 groundstate using one of the above techniques, and then repeat the process, subtracting off the wavefunction's projection onto the already found groundstate 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.

3.5 The finite-element discrete variable representation

[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 od 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: pre-conditioning the potential to be an approximation better representable in the DVR basis.]

3.6 Fourth order Runge-Kutta in an instantaneous local interaction picture

TODO:

- Cite RSP stuff, talk about how this is ideal for FEDVR
- talk about how this solves the last remaining reason why RK4 can't reliably be used over those difficult to implement unitary methods (well, they're difficult when the Hamiltonian isn't super simple)

- spruik benefits of RK4 “workhorse” etc, no special transformations of DE required, good accuracy per step etc.

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:

$$\frac{d}{dt}\langle n|\psi(t)\rangle = -\frac{i}{\hbar}\sum_m \langle n|\hat{H}(t)|m\rangle \langle m|\psi(t)\rangle\,, \tag{3.1}$$

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. Often, and if the basis is chosen well, this type of evolution is dominated by simple dynamical phase evolution, that is, $\hat{H}(t)$ is almost diagonal in the $|n\rangle$ basis. A transformation into an *interaction picture* (IP) 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 take 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[1], and transforming between bases may be computationally expensive (involving large matrix-vector multiplications). Furthermore, the Hamiltonian may change sufficiently from one part of the computation to another that the original time-independent Hamiltonian to 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 not have to take unreasonably small timesteps.

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

$$\hat{H}(t) = \hat{H}_{\rm diag}(t) + \hat{H}_{\rm nondiag}(t) \tag{3.2}$$

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

$$\hat{H}_0^{t'} = \hat{H}_{\rm diag}(t'), \tag{3.3}$$

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

$$|\psi_I^{t'}(t)\rangle = e^{\frac{i}{\hbar}\hat{H}_0^{t'}(t-t')}|\psi(t)\rangle\,, \tag{3.4}$$

which obeys the differential equation:

$$\frac{d}{dt}|\psi_I^{t'}(t)\rangle = e^{\frac{i}{\hbar}\hat{H}_0^{t'}(t-t')}\frac{d}{dt}|\psi(t)\rangle + \frac{i}{\hbar}\hat{H}_0^{t'}|\psi_I^{t'}(t)\rangle\,, \tag{3.5}$$

where:

$$|\psi(t)\rangle = e^{-\frac{i}{\hbar}\hat{H}_0^{t'}(t-t')}\left|\psi_I^{t'}(t)\right\rangle \tag{3.6}$$

[1] such as a spatial basis which allows for partitioning the integration region over multiple nodes on a cluster or cores on a GPU

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, then solving the differential equation for $|\psi_I^{t'}(t)\rangle$ should allow one to take larger integration timesteps than solving (3.1) directly. And if not, then it should do no harm other than the (small) computational costs of computing some extra exponentials.

Equation (3.4) 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 (3.4), (3.5) and (3.6) become:

$$\langle n|\psi_I^{t'}(t)\rangle = e^{i\omega_n^{t'}(t-t')}\langle n|\psi(t)\rangle\,, \tag{3.7}$$

$$\frac{d}{dt}\langle n|\psi_I^{t'}(t)\rangle = e^{i\omega_n^{t'}(t-t')}\frac{d}{dt}\langle n|\psi(t)\rangle + i\omega_n^{t'}\langle n|\psi_I^{t'}(t)\rangle\,, \tag{3.8}$$

and:

$$\langle n|\psi(t)\rangle = e^{-\omega_n^{t'}(t-t')}\langle n|\psi_I^{t'}(t)\rangle\,, \tag{3.9}$$

where we have defined:

$$\omega_n^{t'} = \frac{1}{\hbar}\hat{H}_0^{t'} \tag{3.10}$$

This is in contrast to RK4IP [cite], in which the interaction picture uses the Fourier basis and transforming to and from it involves fast Fourier transforms. RK4IP was developed to augment computations in which FFTs were already in use for evaluating spatial derivatives, and so its use of FFTs imposed no additional cost. Even if one is using FFTs already, our method is still useful for avoiding fast oscillating dynamical phases that are not due to the kinetic term (which is the term of the Hamiltonian that RK4IP takes as its H_0). We compare results of the two methods below [TODO].

3.6.1 Algorithm

Our idea is to use (3.4) 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 in any way compared to if ordinary RK4 were being used, and the only modification to calling code required is for a function to compute and return $\omega_n^{t'}$ be provided.

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, the components $\psi_0^{(n)}$ of the state vector at the start of the timestep, a function $F(t, \psi^{(n)})$

which takes a time and (the components of) a state vector and returns the time derivative for each component, and a function $G(t, \psi^{(n)})$ which takes the same inputs and returns the interaction picture oscillation frequency ω_t for each component at a particular time.

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

$$F(t, \psi(\vec{r}, t)) = -\frac{i}{\hbar}\left[-\frac{\hbar^2}{2m}\nabla^2 + V(\vec{r}, t) + |\psi(\vec{r}, t)|^2\right]\psi(\vec{r}, t), \tag{3.11}$$

and

$$G(t, \psi(\vec{r}, t)) = V(\vec{r}, t) + |\psi(\vec{r}, t)|^2. \tag{3.12}$$

Note that each symbol with a superscript $^{(n)}$ denotes the entire set over all n, subscripts denote the different stages of RK4, and all operations are elementwise,. The only opportunity for non-elementwise operations may occur within F, which contains the details of any couplings between basis states for whatever system of equations is being solved, for example, using Fast Fourier transforms or finite differences to evaluate the Laplacian in (3.11).

```
function RK4ILIP($t_0$, $\Delta t$, $\psi_0^{(n)}$, $F$)
    $f_1^{(n)} \leftarrow F(t_0, \psi_0^{(n)})$        ▷ First evaluation of Schrödinger picture DE
    $\omega^{(n)} \leftarrow G(t_0, \psi_0^{(n)})$        ▷ Oscillation frequencies: $\hbar\omega^{(n)} = \langle n|\hat{H}_{\text{diag}}(t_0)|n\rangle$
    $k_1^{(n)} \leftarrow f_1^{(n)} + i\omega^{(n)}\psi_0^{(n)}$        ▷ Evaluate (3.8) with $t - t' = 0$
    $\varphi_1^{(n)} \leftarrow \psi_0^{(n)} + k_1^{(n)}\frac{\Delta t}{2}$        ▷ First RK4 estimate of IP state vector, at $t = t_0 + \frac{\Delta t}{2}$
    $\psi_1^{(n)} \leftarrow e^{-i\omega^{(n)}\frac{\Delta t}{2}}\varphi_1^{(n)}$        ▷ Convert first estimate back to SP
    $f_2^{(n)} \leftarrow F(t_0 + \frac{\Delta t}{2}, \psi_1^{(n)})$        ▷ Second evaluation of Schrödinger picture DE
    $k_2^{(n)} \leftarrow e^{i\omega^{(n)}\frac{\Delta t}{2}}f_2^{(n)} + i\omega^{(n)}\varphi_1^{(n)}$        ▷ Evaluate (3.8) with $t - t' = \frac{\Delta t}{2}$
    $\varphi_2^{(n)} \leftarrow \psi_0^{(n)} + k_2^{(n)}\frac{\Delta t}{2}$        ▷ Second RK4 estimate of IP state vector, at $t = t_0 + \frac{\Delta t}{2}$
    $\psi_2^{(n)} \leftarrow e^{-i\omega^{(n)}\frac{\Delta t}{2}}\varphi_2^{(n)}$        ▷ Convert second estimate back to SP
    $f_3^{(n)} \leftarrow F(t_0 + \frac{\Delta t}{2}, \psi_2^{(n)})$        ▷ Third evaluation of Schrödinger picture DE
    $k_3^{(n)} \leftarrow e^{i\omega^{(n)}\frac{\Delta t}{2}}f_3^{(n)} + i\omega^{(n)}\varphi_2^{(n)}$        ▷ Evaluate (3.8) with $t - t' = \frac{\Delta t}{2}$
    $\varphi_3^{(n)} \leftarrow \psi_0^{(n)} + k_3^{(n)}\Delta t$        ▷ Third RK4 estimate of IP state vector, at $t = t_0 + \Delta t$
    $\psi_3^{(n)} \leftarrow e^{-i\omega^{(n)}\Delta t}\varphi_3^{(n)}$        ▷ Convert third estimate back to SP
    $f_4^{(n)} \leftarrow F(t_0 + \Delta t, \psi_3^{(n)})$        ▷ Fourth evaluation of Schrödinger picture DE
    $k_4^{(n)} \leftarrow e^{i\omega^{(n)}\Delta t}f_4^{(n)} + i\omega^{(n)}\varphi_3^{(n)}$        ▷ Evaluate (3.8) with $t - t' = \Delta t$
    $\varphi_4 \leftarrow \psi_0 + \frac{\Delta t}{6}\left(k_1^{(n)} + 2k_2^{(n)} + 2k_3^{(n)} + k_4^{(n)}\right)$        ▷ Fourth RK4 estimate, at
$t = t_0 + \Delta t$
    $\psi_4^{(n)} \leftarrow e^{-i\omega^{(n)}\Delta t}\varphi_4^{(n)}$        ▷ Convert fourth estimate back to SP
    return $\psi_4^{(n)}$        ▷ Return the computed SP state vector at $t = t_0 + \Delta t$
end function
```

3.6.2 Results

- compare with ordinary RK4, RK4IP and split-step for the 2D harmonically trapped turbulent BEC example. Increase timestep until they all break and make a plot of energy error over the 100ms simulation as a function of timestep. Expect all methods to be about the same, with RK4IP being slightly lower error but not much. RK4IP will not blow up at large timesteps, but that doesn't mean it's accurate. Comment on the fact that larger timesteps with RK4IP or split step might be stable in that they don't cause the simulation to blow up, but that timesteps larger than that dictated by the dispersion relation necessarily miss some of the nonlinear dynamics, leading to that inaccuracy. This is ok if you are close to a groundstate such that the dynamics are all pretty linear, but it's not OK for vortices and the like. To demonstrate this, perhaps repeat the comparison with a BEC full of sound waves but no vortices.

- compare the three methods for the case of an anharmonic trap, increasing the degree of the polynomial $V = r^\alpha$. Show that as soon as a potential higher than a few times the chemical potential exists anywhere in the simulation region, ordinary RK4 requires much smaller timesteps (than that give by the dispersion relation) to not blow up. Split step would probably still do fine.
- but what about when the equation is more complicated? There is no split step method that can do a rotating frame without colossal matrix-vector products, since the rotation terms are not diagonal in either the spatial or the Fourier basis. This is where RK4ILIP should shine, because it doesn't care how complicated your Hamiltonian is. So compare the RK4 methods for the case of an anharmonic trap in a rotating frame. (include a harmonic trap that exactly cancels the centrifugal potential, in addition to the anharmonic trap, as per Tapio's suggestion).

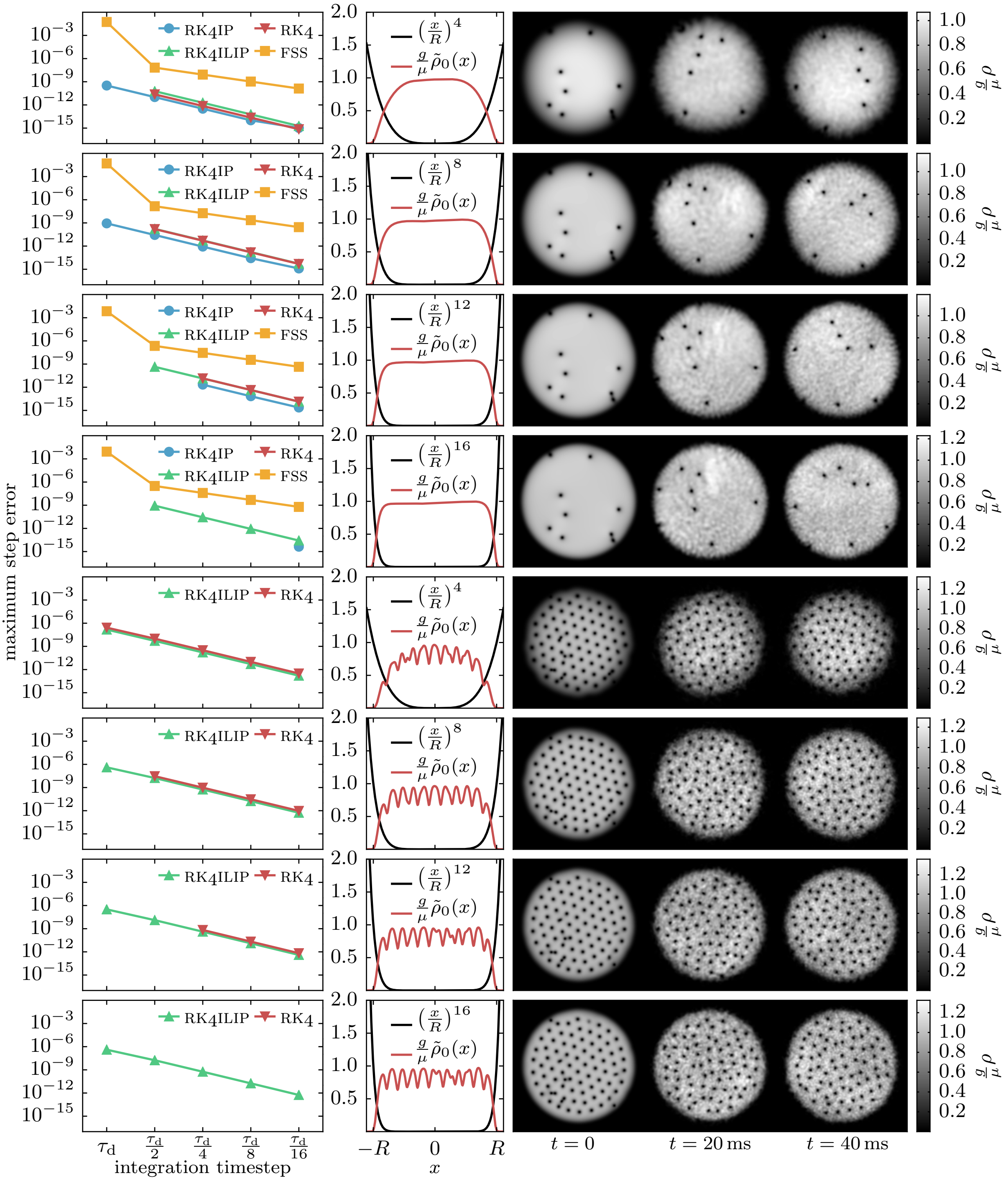

Figure 3.1: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras consequat convallis lobortis. Fusce posuere, nibh non sodales dignissim, neque mauris eleifend augue, et gravida nunc magna quis mi. Phasellus cursus finibus convallis. In tempor purus quis urna posuere efficitur. Aenean posuere aliquet ligula at ultrices. Praesent feugiat lacinia enim, in aliquam turpis vestibulum non. Quisque diam lectus, malesuada vel imperdiet sit amet, posuere pretium ante. Aliquam vestibulum nisi in ipsum posuere pellentesque.

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 appratus itself.

4.1 Dual species apparatus

4.1.1 Vacuum chamber and bakeout

4.2 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.

4.2.1 Optics setup

[include layout diagram of setup I designed]

This page intentionally left blank

Chapter 5

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 runmananger, 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.

5.1 labscript

5.2 rumananger

5.3 BLACS

5.4 lyse

This page intentionally left blank

Chapter 6

Particle velocimetry of vortices in Bose–Einstein condensates

6.1 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.

6.2 Sisyphus cooling in a magnetic field

Describe the scheme for Sisyphus cooling I developed, and show 1D simulation results.

This page intentionally left blank

Chapter 7

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 c2.

This page intentionally left blank

Chapter 8

Hidden variables for semiclassical models with state-dependent forces

Define/describe what a hidden variable theory is, drawing heavily on Aaronson's 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.