IN SEARCH OF OPTIMALITY

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by
ILIA ZINTCHENKO
MSc. in Micro and Nanosystems ETH Zurich
born on 10.08.1989
citizen of
Russian Federation

accepted on the recommendation of
Prof. Dr. Matthias Troyer, examiner
Prof. Dr. Helmut Katzgraber, co-examiner

2016
## Contents

<table>
<thead>
<tr>
<th>Sections</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>vii</td>
</tr>
<tr>
<td>Riassunto</td>
<td>ix</td>
</tr>
<tr>
<td>I  Quantum gases</td>
<td>1</td>
</tr>
<tr>
<td>1  Introduction</td>
<td>3</td>
</tr>
<tr>
<td>2  Density functional theory</td>
<td>7</td>
</tr>
<tr>
<td>2.1 Hohenberg-Kohn</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Kohn-Sham</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Finite temperature</td>
<td>10</td>
</tr>
<tr>
<td>2.4 Dynamics</td>
<td>10</td>
</tr>
<tr>
<td>3  Unitary Fermi gas at a finite temperature</td>
<td>13</td>
</tr>
<tr>
<td>3.1 Exact conditions</td>
<td>14</td>
</tr>
<tr>
<td>3.2 Energy functionals</td>
<td>15</td>
</tr>
<tr>
<td>3.3 Applications</td>
<td>16</td>
</tr>
<tr>
<td>3.4 Conclusion</td>
<td>22</td>
</tr>
<tr>
<td>4  Ferromagnetism of the repulsive atomic Fermi gas</td>
<td>23</td>
</tr>
<tr>
<td>4.1 Quantum Monte-Carlo study</td>
<td>24</td>
</tr>
<tr>
<td>4.2 Dynamics and three body recombination</td>
<td>26</td>
</tr>
<tr>
<td>4.3 Conclusion</td>
<td>31</td>
</tr>
<tr>
<td>II Quantum computing</td>
<td>35</td>
</tr>
<tr>
<td>5  Introduction</td>
<td>37</td>
</tr>
</tbody>
</table>
6 Randomized gap and amplitude estimation
   6.1 Bayesian inference ........................................ 40
   6.2 Randomized gap estimation ................................. 42
       6.2.1 Numerical tests ...................................... 45
   6.3 Implementing random unitaries ............................. 48
   6.4 Adiabatic elimination of eigenvalues ..................... 50
   6.5 Application to amplitude estimation ....................... 52
   6.6 Application to control map learning ...................... 53
       6.6.1 Two level example ................................... 55
       6.6.2 Learning $2 \times 2$ single qubit control maps ....... 57
   6.7 Conclusion .................................................. 59

III Optimisation

7 Introduction ......................................................... 63

8 Optimised simulated annealing for Ising spin glasses ......... 67
   8.1 Optimisations ............................................... 69
       8.1.1 Forward computation of $\Delta E$ .................... 69
       8.1.2 Fixed loop lengths and unrolling .................... 69
       8.1.3 Fast random number generators ...................... 69
       8.1.4 Deterministic traversal over lattice sites .......... 70
       8.1.5 Precomputing random numbers ....................... 70
       8.1.6 Optimisations for bipartite graphs ................ 70
       8.1.7 Multi-spin coding .................................... 71
   8.2 Optimising annealing strategies ............................. 75
       8.2.1 Optimising the schedules ............................ 75
       8.2.2 Optimising annealing times .......................... 75

9 From local to global ground states in Ising spin glasses ..... 79
   9.1 Boundary condition dependence in frustrated spin systems .... 80
       9.1.1 Review ............................................... 81
       9.1.2 Weak Field Scaling in $d = 2$ ...................... 82
       9.1.3 Boundary Condition Dependence ...................... 85
   9.2 Finding the exact global ground state ..................... 87
       9.2.1 Single spin reduction ................................ 87
       9.2.2 Patch reduction ...................................... 89
       9.2.3 Improved patch reduction ............................ 89
       9.2.4 Empirical scaling .................................... 91
   9.3 Hierarchical search ......................................... 91
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.3.1</td>
<td>Optimization of groups</td>
<td>92</td>
</tr>
<tr>
<td>9.3.2</td>
<td>Hierarchical recursive algorithm</td>
<td>92</td>
</tr>
<tr>
<td>9.3.3</td>
<td>Selecting groups</td>
<td>94</td>
</tr>
<tr>
<td>9.3.4</td>
<td>Results</td>
<td>94</td>
</tr>
<tr>
<td>9.4</td>
<td>Conclusion</td>
<td>99</td>
</tr>
<tr>
<td>10</td>
<td>Partial reinitialisation for optimisers</td>
<td>105</td>
</tr>
<tr>
<td>10.1</td>
<td>Algorithm</td>
<td>106</td>
</tr>
<tr>
<td>10.1.1</td>
<td>Choosing $M_\ell$</td>
<td>109</td>
</tr>
<tr>
<td>10.2</td>
<td>Benchmarks</td>
<td>110</td>
</tr>
<tr>
<td>10.2.1</td>
<td>Training hidden Markov models</td>
<td>111</td>
</tr>
<tr>
<td>10.2.2</td>
<td>Clustering with k-means</td>
<td>112</td>
</tr>
<tr>
<td>10.2.3</td>
<td>Clustering with k-medoids</td>
<td>114</td>
</tr>
<tr>
<td>10.2.4</td>
<td>Training Boltzmann machines</td>
<td>115</td>
</tr>
<tr>
<td>10.3</td>
<td>Conclusion</td>
<td>118</td>
</tr>
<tr>
<td>IV</td>
<td>Scientific computing framework</td>
<td>119</td>
</tr>
<tr>
<td>11</td>
<td>Introduction</td>
<td>121</td>
</tr>
<tr>
<td>12</td>
<td>Project Whiplash</td>
<td>123</td>
</tr>
<tr>
<td>12.1</td>
<td>Overview</td>
<td>123</td>
</tr>
<tr>
<td>12.1.1</td>
<td>Usability</td>
<td>124</td>
</tr>
<tr>
<td>12.1.2</td>
<td>Security</td>
<td>125</td>
</tr>
<tr>
<td>12.1.3</td>
<td>Portability</td>
<td>125</td>
</tr>
<tr>
<td>12.1.4</td>
<td>Fault tolerance</td>
<td>125</td>
</tr>
<tr>
<td>12.1.5</td>
<td>Collaboration</td>
<td>126</td>
</tr>
<tr>
<td>12.1.6</td>
<td>Memoisation</td>
<td>127</td>
</tr>
<tr>
<td>12.1.7</td>
<td>MapReduce</td>
<td>128</td>
</tr>
<tr>
<td>12.1.8</td>
<td>Scheduling</td>
<td>128</td>
</tr>
<tr>
<td>12.1.9</td>
<td>Query optimisation</td>
<td>128</td>
</tr>
<tr>
<td>12.1.10</td>
<td>Database</td>
<td>129</td>
</tr>
<tr>
<td>12.2</td>
<td>Conclusion</td>
<td>129</td>
</tr>
<tr>
<td>Afterword</td>
<td></td>
<td>163</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td></td>
<td>165</td>
</tr>
</tbody>
</table>
Abstract

This thesis consists of four parts: quantum gases, quantum computing, optimisation and computational frameworks. The parts are not connected and can be read separately.

The first part is about density functional theory for cold atomic gases. Density functional theory allows to study significantly larger and more complex systems than, for example, exact diagonalisation, but at the cost of accuracy and reliability. We use it to study static and dynamic properties of Fermi gases at unitarity and in the weakly repulsive regime.

The second part is about quantum computing. We introduce a novel method based on Bayesian inference, which we term randomised phase estimation, to learn the spectral gaps in a quantum device without the need for ancillary qubits or well calibrated gates. Bayesian inference is ubiquitous in classical machine learning and allows to efficiently gain information about unknown parameters of a system based on randomised sampling.

The third part is about optimisation. With the task of finding ground states of Ising spin glasses as a benchmark, we introduce a set of tricks for greatly increase the performance of simulated annealing and propose a method to boost the power of an optimiser by using it to solve sub-sets of variables rather than the whole problem at once. This method can also be used to extend the advantage of a small device to solve large optimisation problems. On a different problem, namely fitting models to data in various machine learning applications, we propose another way, termed partial reinitialisation, to significantly increase performance. The core idea is to reinitialise only sub-sets of variables rather than the whole system after each call to the optimiser.

The fourth part is about high-performance computing. We introduce a collaborative, distributed framework which makes using a cluster not any harder than running a script on a laptop. Aimed at computational scientists, the goal of Project Whiplash is to maximise the time spent on actual research and to become an industry standard for general purpose, high-performance scientific computing.
Questo lavoro di testi consta di quattro parti: gas quantistici, informatica quantistica, ottimizzazione e strutture computazionali. Le diverse parti non sono interconnesse e possono leggersi separatamente.

La prima parte riguarda la teoria del funzionale densità per gas di atomi freddi. La teoria del funzionale densità permette di studiare dei sistemi ben più grandi e complessi rispetto, per esempio, alla diagonalizzazione esatta, ma al prezzo di minore precisione ed affidabilità. Utilizziamo questo approccio per studiare proprietà statiche e dinamiche di gas di Fermi nel regime unitario e per deboli interazioni repulsive.

La seconda parte riguarda l’informatica quantistica. Qui introduciamo un nuovo metodo basato sull’inferenza Bayesiana, da noi denominato stima aleatoria della fase, che permette di apprendere le gap spettrali in un dispositivo quantistico senza la necessità di qubit ausiliari o di porte quantistiche ben calibrate. L’inferenza Bayesiana è assai diffusa nei metodi classici di apprendimento automatico e permette di ottenere efficacemente informazioni sui parametri incogniti di un sistema basato sul campionamento aleatorio.

La terza parte riguarda l’ottimizzazione. Fissandoci come obiettivo di riferimento quello di trovare gli stati fondamentali nei modelli di vetri di spin, introduciamo una serie di trucchi che migliorano sensibilmente il rendimento del “simulated annealing”. Proponiamo inoltre un metodo per migliorare il rendimento di un ottimizzatore, utilizzandolo per risolvere delle sotto classi di variabili piuttosto che per l’intero problema. Questo metodo può essere inoltre utilizzato per estendere i vantaggi dei piccoli dispositivi alla risoluzione di problemi di ottimizzazioni su larga scala. Per un problema differente, cioè per trovare il migliore modello che riproduca i dati in diverse applicazioni dell’apprendimento automatico, proponiamo un’altra strategia, denominata reinizializzazione parziale, che migliora sensibilmente le prestazioni. L’idea di base è di reinizializzare solamente dei sottoinsiemi di variabili piuttosto che l’intero sistema, dopo ciascuna chiamata all’ottimizzatore.

La quarta parte riguarda il calcolo ad elevate prestazioni. Introduciamo una struttura di calcolo, collaborativa e distribuita, che rende l’uso di
un supercalcolatore semplice quanto lanciare un piccolo programma su un computer portatile. Rivolto ai fisici computazionali, obiettivo del progetto "Whiplash" è di massimizzare il tempo passato sulla vera ricerca nonché di divenire uno standard industriale per il calcolo scientifico multiuso ed ad alte prestazioni.
Part I

Quantum gases
Chapter 1

Introduction

This part is based on the papers [238, 230, 237].

Feynman envisioned a quantum simulator [1] almost four decades ago. A quantum simulator allows studying the properties of interacting quantum systems and gives access to many important observables, which can be used to reconstruct the equation of state and gain a microscopic understanding of many-body physics. Cold atoms are one realisation of such a simulator.

Cold atoms are maintained at temperatures close to absolute zero and are millions of times less dense than air at atmospheric pressure to maximise the observable effects of quantum mechanics [2]. Theoretical predictions can be directly probed in experiments with a wide range of tunable parameters including the interaction strength, temperature, external potential and population imbalance, seldom accessible in solid state materials [3] and allowing the realisation of model Hamiltonians relevant for condensed matter physics [4].

In recent years, ultracold atoms have emerged as the ideal experimental system to investigate intriguing quantum phenomena caused by strong correlations and gained widespread international interest in the physics community after the observation of Bose-Einstein condensation at MIT in 1995 [5]. Since then the field has rapidly grown to become a leading tool for groundbreaking observations at temperatures close to absolute zero such as quantised vertices [6], the Josephson effect [7], bosonic superfluidity [8] and Efimov states [9].

In a real material, the nuclei of atoms impose a periodic potential on the electrons. Such a system can be modelled with cold atoms by using an optical lattice generated by interfering lasers from multiple directions, which impose a periodic potential on the atoms through the optical dipole force.
CHAPTER 1. INTRODUCTION

Effectively any lattice geometry can be engineered this way. The kinetic energy can also be tuned by varying the lattice strength \[10\]. If the lattice is strong enough, such a system represents the Hubbard model and becomes a Mott insulator for very large ratios of interaction to kinetic energy. The effective interaction energy can be tuned across several orders of magnitude through Feshbach resonances using external magnetic fields \[11\].

Although in the early days most interest was in bosonic systems, attention has recently shifted towards fermions with the creation of the first ultracold Fermi gas in 1999 \[12\], paving the way towards novel physics. We will here focus on two regimes. The weakly repulsive and the unitary Fermi gas.

Recent advancements in quantum Monte Carlo techniques allowed sampling of finite temperature properties of homogeneous correlated systems precisely \[13, 14\]. One example is the ultra-cold unitary Fermi gas (UFG), where theory and experiment show great agreement \[15, 16\]. A UFG can be realised experimentally by tuning the scattering length of atoms to infinity through Feshbach resonances \[11\]. Without a length scale for interaction, thermodynamics are universal and in many ways similar to the ideal Fermi gas. The UFG can, for example, be used to study the neutron matter in the interior of a neutron stars.

We use data from bold diagrammatic Monte Carlo \[15\] to obtain a finite temperature exchange-correlation energy functional for a UFG and study static and dynamic properties in a harmonic trap using both the Kohn-Sham and Thomas-Fermi local density approximations. We find that the Thomas-Fermi approximation breaks down for highly anisotropic traps with mesoscopic number of atoms. Also, we investigate a set of exact conditions for the energy functional, fulfillment of which naturally leads to the virial theorem and undamped breathing modes of a UFG in a harmonic trap.

Although attractive fermions have been widely studied, strongly repulsive interactions are much less investigated and have so far had no solid experimental verification mainly due to rapid three-body recombination at large scattering lengths. Such a system has previously been suggested as a model to study itinerant ferromagnetism with two-component phases. Itinerant ferromagnetism, which occurs in transition metals like nickel, cobalt and iron, is an intriguing quantum mechanical phenomenon due to strong correlations between delocalised electrons.

Two fundamental models were originally introduced to explain itinerant ferromagnetism in transition metals. The first is the Stoner model \[17\], whose Hamiltonian represents a Fermi gas in a continuum with short-range repulsive interactions originally treated at the mean-field level. The second is the Hubbard model \[18\], which describes electrons hopping between sites of a discrete lattice with on-site repulsion. Despite the simplicity of these
models, their zero-temperature ferromagnetic behaviour is still uncertain.

Although there has recently been some experimental indication of itinerant ferromagnetism in a gas of $^6$Li atoms [19], there have been a number of refutations of these results, including suggestions that a strongly correlated state can be more energetically feasible [20] and that ferromagnetism may not be achievable at all [21] due to high pair formation rates causing localised heating. Unambiguous experimental proof of ferromagnetic behaviour in atomic gases is hence still missing.

Using density functional theory, with an exchange-correlation functional obtained from continuous-space quantum Monte-Carlo, we show that three-body recombination is small enough to permit experimental detection of the phase and that in such a setup ferromagnetic domains emerge rapidly from a paramagnetic initial state.
Chapter 2

Density functional theory

A non-relativistic $N$-particle system can be described by the Hamiltonian

$$\hat{H} = -\sum_{i=1}^{N} \frac{\Delta^2}{2} + \sum_{i<j}^{N} V(r_{ij}),$$  \hspace{1cm} (2.1)

where we set $\hbar = m = k_B = 1$ and $V(r)$ includes the external potential and inter-particle interactions. The complexity of finding the ground state wavefunction $\Psi(r_1, r_2, \ldots, r_N)$ of this system, which satisfies the Schrodinger equation $\hat{H}\Psi = E\Psi$ scales exponentially with the number of particles and we therefore resort to various approximations for large systems.

One such approximation is density functional theory (DFT) [22, 23], which is currently the most important tool for ab-initio simulations of the complex electronic structure of solid state systems and is widely popular in the materials theory and quantum chemistry community. DFT gives systematically reliable results for simple metals and semiconductors, but extension to strongly correlated materials still represents an outstanding open challenge [24]. It has also only recently gained popularity with cold atoms.

As pointed out in the landmark work of Hohenberg-Kohn [23], the ground state energy of a many-particle system is given by

$$E[\rho] = \int d\mathbf{r} \rho(\mathbf{r}) V_{\text{ext}}(\mathbf{r}) + F[\rho]$$  \hspace{1cm} (2.2)

$$F[\rho] = T[\rho] + E_{\text{xc}}[\rho]$$  \hspace{1cm} (2.3)

where $\rho(\mathbf{r})$ is the density, $V_{\text{ext}}(\mathbf{r})$ the external potential, $T[\rho]$ is the kinetic energy and $E_{\text{xc}}[\rho]$ the exchange correlation energy due to inter-particle interactions. We will here use the local spin density approximation (LSDA) [23] by assuming homogeneity at each position and write the exchange-correlation
energy in the form

\[ E_{xc} = \int d\mathbf{r} \rho(\mathbf{r}) e_{xc}[\rho] \]  

(2.4)

where \( e_{xc}[\rho] \) is the equation of state often obtained from quantum Monte-Carlo calculations for uniform systems. Hence, LSDA is exact only if the potential is uniform. Even for non-uniform potentials, however, it often gives surprisingly accurate results and allows to effectively extend quantum Monte-Carlo calculations to larger systems and more complex potential profiles.

We will here consider two different forms for the kinetic energy represented by the Hohenberg-Kohn [23] and Kohn-Sham [25] formalisms.

### 2.1 Hohenberg-Kohn

Within Hohenberg-Kohn, the kinetic term is in the Thomas-Fermi form

\[ T[\rho] = \int d\mathbf{r} \rho(\mathbf{r}) t[\rho], \]  

(2.5)

where \( t[\rho] = \frac{3}{5} E_F \), \( E_F = k_F^2/2 \) is the Fermi energy and \( k_F = (3\pi^2\rho)^{1/3} \) is the Fermi wave-vector. Given a total number of particles \( N \)

\[ N = \int d\mathbf{r} \rho(\mathbf{r}), \]  

(2.6)

the ground state density is determined by the condition

\[ \delta \left\{ E[\rho] - \mu \int d\mathbf{r} \rho(\mathbf{r}) \right\} = 0; \]  

(2.7)

where the chemical potential \( \mu \) acts as a Lagrange multiplier. The density is then a solution of the Euler-Lagrange equation

\[ V_{\text{ext}}(\mathbf{r}) - \mu + \frac{\delta F[\rho]}{\delta \rho} = 0 \]  

(2.8)

However, as the kinetic energy is that of non-interacting fermions, the accuracy of this method is limited.

### 2.2 Kohn-Sham

In the more rigorous Kohn-Sham formalism, the kinetic energy is implicitly found by self-consistently solving \( N \) one particle equations with a combined
2.2. KOHN-SHAM

ground-state wavefunction represented by a single Slater determinant [26]

$$\Psi = \frac{1}{\sqrt{N!}} \det[\psi_1 \psi_2 \ldots \psi_N]$$ (2.9)

The one-particle states $\psi_i$ solve

$$\left[ -\frac{\nabla^2}{2} + V_{\text{eff}}(r) \right] \psi_i = \epsilon_i \psi_i$$ (2.10)

$$\rho(r) = \sum_i |\psi_i(r)|^2$$ (2.11)

$$V_{\text{eff}}(r) = V_{\text{ext}}(r) + \frac{\delta E_{\text{xc}}}{\delta \rho(r)}$$ (2.12)

where the difference between the exact and non-interacting kinetic energies is baked into the exchange-correlation energy $E_{\text{xc}}$. Given the density is $v$-representable [27], the total energy is minimised by the $N$ lowest lying orbitals. The density is $v$-representable if it is uniquely defined as a non-degenerate ground-state density corresponding to the interacting Hamiltonian with an arbitrary external potential. If, however, $v$-representability is not fulfilled, $N$ orbitals would need to be chosen from a much larger set such as to minimise the total energy $E[\rho]$. We will here assume $v$-representability is fulfilled.

From the eigenvalues

$$\epsilon_i = \left\langle \psi_i \left| -\frac{\nabla^2}{2} + V_{\text{ext}} + V_{\text{xc}} \right| \psi_i \right\rangle$$ (2.13)

and density profiles, the energies are computed as follows

$$E_{\text{tot}} = \sum_i \epsilon_i - \langle \rho | V_{\text{xc}} \rangle + \langle \rho | e_{\text{xc}} \rangle =$$

$$= \sum_i \left\langle \psi_i \left| -\frac{\nabla^2}{2} \right| \psi_i \right\rangle + \langle \rho | V_{\text{ext}} \rangle + \langle \rho | V_{\text{xc}} \rangle - \langle \rho | V_{\text{xc}} \rangle + \langle \rho | e_{\text{xc}} \rangle =$$

$$= E_{\text{kin}} + E_{\text{pot}} + E_{\text{xc}}$$ (2.14)

$$E_{\text{kin}} = E_{\text{tot}} - E_{\text{pot}} - E_{\text{xc}}$$ (2.15)

where $\langle f | g \rangle \equiv \int dr f \cdot g$ and $E_{\text{tot}}$, $E_{\text{kin}}$ and $E_{\text{pot}}$ are the total, kinetic and potential energies respectively. The equation of state $e_{\text{xc}}[\rho]$ is obtained from quantum Monte-Carlo calculations of interacting fermions in homogeneous systems.
CHAPTER 2. DENSITY FUNCTIONAL THEORY

2.3 Finite temperature

At finite temperature, the grand-canonical potential is written as a functional of real-space density

\[ \Omega[\rho] = F[\rho] + \int d\mathbf{r} \rho(\mathbf{r})(V_{\text{ext}}(\mathbf{r}) - \mu) \]  
\[ F[\rho] = F_S + F_{\text{hxc}} \]  

(2.16)  
(2.17)

where \( \mu \) is the chemical potential and \( F[\rho] \) is a universal free-energy functional which is independent of external potential [28]. \( F_S \) is the kentropy [29] of an ideal Fermi gas with density \( \rho(\mathbf{r}) \) and \( F_{\text{hxc}} \) is the Hartree-exchange-correlation free-energy whose exact form is unknown.

Eq.(2.17) is then minimized by solving the single particle Kohn-Sham equations as above, except the exchange-correlation potential is now \( V_{\text{xc}} = \frac{\delta F_{\text{hxc}}}{\delta \rho} \) and the density is computed as

\[ \rho = \sum_i w_i |\psi_j|^2 \]  

(2.18)

where \( w_i = [1 + e^{(\epsilon_i - \mu)/T}]^{-1} \) is the Fermi-Dirac distribution at temperature \( T \).

2.4 Dynamics

Time-dependent density functional theory was first introduced by Runge and Gross in the form of the Runge-Gross theorem [30], which proved the one-to-one mapping between the time-varying external potential and the density of the system. The time-dependent Schrödinger equation for Kohn-Sham orbitals is

\[ i \frac{\partial}{\partial t} \psi_i(\mathbf{r}, t) = \left[ -\frac{\Delta^2}{2} + V_{\text{eff}}(\mathbf{r}, t) \right] \psi_i(\mathbf{r}, t) \]  

(2.19)

and the density is like in the static case

\[ \rho(\mathbf{r}, t) = \sum_i N |\psi_i(\mathbf{r}, t)|^2. \]  

(2.20)

Similarly, the effective Kohn-Sham potential contains the external and exchange correlation terms

\[ V_{\text{eff}}(\mathbf{r}, t) = V_{\text{ext}}(\mathbf{r}, t) + V_{\text{xc}}(\mathbf{r}, t), \]  

(2.21)
where
\[ V_{xc}(\mathbf{r}, t) = \frac{\delta A_{xc}}{\delta \rho(\mathbf{r}, t)}, \] (2.22)
and \( A_{xc} \) is the action-functional [31].

The simplest and most popular approximation for \( V_{xc}(\mathbf{r}, t) \), which we will use here, is the Adiabatic Local Density Approximation (ALDA) [30]. Within this approximation, the system is assumed to evolve slowly enough to stay close to equilibrium at all times. We can hence use same exchange-correlation potential as for a static system
\[ V_{xc}(\mathbf{r}, t) = V_{xc}[\rho] \big|_{\rho=\rho(\mathbf{r}, t)} \] (2.23)
Chapter 3

Unitary Fermi gas at a finite temperature

Recent advancements in quantum Monte Carlo techniques allowed sampling the finite temperature properties of homogeneous correlated systems precisely [13, 14]. One example is the ultra-cold unitary Fermi gas (UFG), where theory and experiment show great agreement [15, 16]. A UFG can be realised experimentally by tuning the scattering length of atoms to infinity through Feshbach resonances [11]. Without a length scale for interaction, thermodynamics are universal and in many ways similar to the ideal Fermi gas.

We use the bold diagrammatic Monte Carlo (BDMC) results in [15] to obtain a finite temperature exchange-correlation energy functional for a UFG and study static and dynamic properties of a UFG in a harmonic trap using the Kohn-Sham (KS-LDA) and Thomas-Fermi (TFA) local density approximations. We show that for highly anisotropic traps and mesoscopic numbers of particles the TFA breaks down relative to the more accurate KS-LDA.

Various exact conditions for energy functionals used in DFT calculations are usually derived by coordinate scaling [32] and adiabatic connection [33] techniques. Due to the absence of an interaction length scale in a UFG, these exact conditions are greatly simplified. We derive two of them and show their connection to the virial theorem [34, 35, 36] and undamped breathing modes for a UFG. This provides a guideline for exploring better approximations beyond the LDA.
CHAPTER 3. UNITARY FERMI GAS AT A FINITE TEMPERATURE

3.1 Exact conditions

The virial theorem relates the kinetic energy and density distribution of a quantum system. For the unitary Fermi gas and non-interacting KS system it reads respectively [34, 35, 36]:

\[ K = \frac{1}{2} \langle r \cdot \nabla V_{\text{ext}} \rangle \]

(3.1)

\[ K_S = \frac{1}{2} \langle r \cdot \nabla V_S \rangle, \]

(3.2)

where \( K \) and \( K_S \) are the kinetic energies of the UFG and KS system and we define \( \langle O \rangle = \int d^3r \rho(r)O(r) \).

For a UFG, since the interaction terms are missing, all correlation effects are contained in kinetic energies,

\[ E_{\text{HXC}} = K - K_S \]

(3.3)

Putting together Eq. 2.12 and 3.3 we obtain,

\[ 2E_{\text{HXC}} = -\langle r \cdot \nabla V_{\text{HXC}} \rangle \]

(3.4)

This is the exact condition for a finite-temperature exchange-correlation potential at unitarity. Eq.(3.4) has similar form as its zero temperature counterpart [37], but is simpler due to the symmetry properties at unitarity.

To show that the LDA exchange-correlation potential fulfils Eq.(3.4), we rescale the coordinates as \( r \rightarrow \lambda r \), but keep the Hamiltonian operator unchanged. The density now scales as \( \rho(\lambda r) = \lambda^3 \rho(\lambda r) \). As

\[ \langle r \cdot \nabla V_{\text{HXC}} \rangle = -\frac{\partial F_{\text{HXC}}[\rho_{\lambda}(r)]}{\partial \lambda} \bigg|_{\lambda=1} \]

(3.5)

we have

\[ 2E_{\text{HXC}} = \frac{\partial F_{\text{HXC}}[\rho_{\lambda}(r)]}{\partial \lambda} \bigg|_{\lambda=1} \]

(3.6)

which is equivalent to Eq.(3.4) expressed as a scaling derivative. In a DFT calculation, any approximation fulfilling Eq.(3.4) or Eq.(3.6) restores the virial theorem in Eq.(3.1). Under LDA we have \( F_{\text{HXC}}[\rho(\lambda r)] \approx \int d^3r f_{\text{HXC}}(\rho_{\lambda}) \) and \( E_{\text{HXC}} \approx \int d^3r e_{\text{HXC}} \), where small capitals denote the (free-)energy per volume. Note that \( e_{\text{HXC}} \) scales as \( \lambda^2 \) while the entropy is scale invariant [29]. Thus \( \frac{\partial F_{\text{HXC}}[\rho_{\lambda}(r)]}{\partial \lambda} \bigg|_{\lambda=1} = 2 \int d^3r e_{\text{HXC}} = 2E_{\text{HXC}} \). Hence, LDA fulfils Eq.(3.6) and the virial theorem at unitarity Eq.(3.1) is restored.
3.2. ENERGY FUNCTIONALS

Figure 3.1: Virial expansion fit of the Bold-Diagrammatic Monte Carlo equation of state (EOS) [15]. The EOS of an ideal Fermi gas is also plotted. For a given density the difference between the two equations of state gives the exchange-correlation potential (see Eq.(3.8)). The interpolation is done with a cubic spline within $-1.5 < \mu/T < 2.25$ to 500 points.

3.2 Energy functionals

Virial expansions are commonly used to investigate high temperature properties of correlated systems, which are encoded in the virial coefficients [38, 39]. The density of a uniform system is given by:

$$\rho = \frac{2}{\lambda_T^2} \sum_{n=1}^{\infty} nb_n z^n$$  \hspace{1cm} (3.7)

where $\lambda_T = [2\pi \hbar^2/(mk_B T)]^{1/2}$ is the thermal wavelength, $z = e^{\mu/T}$ is the fugacity and $b_n$ are the virial coefficients. For an ideal Fermi gas, Eq.(3.7) reduces to $\rho = \frac{2}{\lambda_T^2} f_{3/2}(z_0)$, where $z_0 = e^{\mu_0/T}$ and $f_\nu(z) = \frac{1}{\Gamma(\nu)} \int_0^\infty x^{\nu-1} e^{-x} \frac{1}{z+x} dx$ is the Fermi-Dirac function.

For a unitary Fermi gas, previous theoretical studies found $b_1 = 1$, $b_2 = 3\sqrt{2}/8$ and $b_3 = -0.29095295$ [38, 39]. We fit the equation of state from BDMC calculations in Ref. [15] against Eq.(3.7) up to $b_9$ (see Fig.3.1 and
CHAPTER 3. UNITARY FERMI GAS AT A FINITE TEMPERATURE

Table 3.1: Virial coefficients. $b_1 \sim b_3$ are known from [38, 39], $b_4$ to $b_9$ are obtained by fitting against the BDMC data [15]. Error bars to last digits appear in bracket.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$b_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>$3\sqrt{2}/8$</td>
</tr>
<tr>
<td>3</td>
<td>$-0.29095295$</td>
</tr>
<tr>
<td>4</td>
<td>0.08890(18)</td>
</tr>
<tr>
<td>5</td>
<td>$-0.017099(17)$</td>
</tr>
<tr>
<td>6</td>
<td>0.0020862(14)</td>
</tr>
<tr>
<td>7</td>
<td>$-1.5632(14) \times 10^{-4}$</td>
</tr>
<tr>
<td>8</td>
<td>$6.553(13) \times 10^{-6}$</td>
</tr>
<tr>
<td>9</td>
<td>$-1.175(12) \times 10^{-7}$</td>
</tr>
</tbody>
</table>

Table (3.1)). Nine is the minimal order to give a goodness of fit of 0.23.

In a KS-DFT calculation the KS system is an inhomogeneous ideal Fermi gas which reproduces the density distribution of an interacting system [40]. Under LDA, $F_{\text{HXC}} \approx \int \text{d}r f_{\text{HXC}}(\rho(r))$ and

$$V_{\text{HXC}} = \frac{\delta F_{\text{HXC}}}{\delta \rho} \approx \frac{\partial f_{\text{HXC}}}{\partial \rho}$$

$$= \mu - \mu_0$$

$$= T[\ln z(\rho) - \ln z_0(\rho)] \quad (3.8)$$

With Eq.(3.8), we solve the Kohn-Sham equations Eq.(2.10,2.12,2.11) self-consistently. In each step, with a given density $\rho$, we invert Eq.(3.7) to get $z(\rho)$ and $z_0(\rho)$ and insert them into Eq.(3.8) to get the HXC potential. The process is shown schematically in Figure 3.1. Note that since the thermal wavelength and temperature are involved in this inversion, $V_{\text{HXC}}$ in general depends on the temperature. It is always negative because of the attractive nature of interactions in a UFG. We then update $V_S$ according to Eq.(2.12), solve Eq.(2.10) and calculate the new density using Eq.(2.11).

3.3 Applications

To study the properties of a UFG in experimental conditions, commonly involving harmonic traps, we use the energy-functional obtained above in a KS-LDA framework. We consider a 3D harmonic trap $V_{\text{ext}} = \omega_{xy}^2 (x^2 + y^2)/2 + \omega_z^2 z^2/2$ and calculate various thermodynamic quantities and the density profile of the system.
In a finite-temperature Kohn-Sham calculation the grand-canonical potential is minimised and the virial theorem \[34, 36, 35\] is satisfied after convergence (see Fig.3.3). Thermodynamics quantities are calculated as

\[
E = E_S - \int \text{d} \mathbf{r} (\rho V_{\text{HXC}} - e_{\text{HXC}}) \\
F = F_S - \int \text{d} \mathbf{r} (\rho V_{\text{HXC}} - f_{\text{HXC}}) \\
\Omega = F - \mu N \\
S = (E - F)/T
\]
Their form explicitly fulfils the scaling condition proposed in [29].

As shown in Figure 3.2, the effects of temperature are significantly larger than the exchange-correlation potential. Figure 3.4 shows a comparison of KS-LDA with the exact virial expansion in a harmonic trap [39] and TFA.
3.3. APPLICATIONS

Figure 3.4: Chemical potential vs temperature in an isotropic oscillator with fixed $N = 4$. KS-LDA shows good agreement with the 3rd-order virial expansion in a harmonic trap [39], which is exact in this parameter regime. Inset shows radial densities of an ideal Fermi gas and UFG from KS and TFA calculations.

calculations with fixed $N = 4$. KS-LDA shows good agreement with exact results, while TFA predicts systematically smaller chemical potentials. The radial density of the UFG (see inset of Fig. 3.4) is significantly squeezed towards the center of the trap relative to the ideal Fermi gas. TFA overestimate this effect and predicts an even higher density in the center than KS-LDA.

The density difference $\Delta = \frac{1}{N} \int d\mathbf{r} |\rho_{KS}(\mathbf{r}) - \rho_{TFA}(\mathbf{r})|$ between the two approaches is larger at lower temperatures and smaller particle numbers (see Fig. 3.5). This is because the correlation length increases at lower temperatures and gradient corrections become more significant. For small particle numbers the difference is mainly due to the shell effect [42]. The inset of Fig. 3.5 shows that the density difference becomes larger as the trap becomes more anisotropic. It approaches one for highly anisotropic traps since only one quantum level is filled in the strongly confined direction.

Finally we consider breathing modes of the UFG in an isotropic harmonic
Figure 3.5: Normalised density difference $\Delta = \frac{1}{N} \int dr |\rho_{KS}(r) - \rho_{TFA}(r)|$ between KS-LDA and TFA calculations vs temperature. Main panel for an isotropic trap with atom numbers $N = 100, 300, 1000$. Inset shows difference as a function of aspect ratio $\alpha = \omega_z/\omega_{xy}$ (keep $\bar{\omega} = \sqrt{\omega_{xy}^2/\omega_z}$ unchanged) of the trap for $N = 10$ and 100 atoms at $T/EF = 0.32$. The system is quasi-1d (cigar) for $\alpha \ll 1$ and quasi-2d (pancake) for $\alpha \gg 1$.

The system is quasi-1d (cigar) for $\alpha \ll 1$ and quasi-2d (pancake) for $\alpha \gg 1$.
Combining Eq.(3.11) and Eq.(3.12), we have

\[ \frac{d^2}{dt^2} \langle r^2 \rangle + 4\omega^2 \langle r^2 \rangle = C \]  

where \( C \) is a time-independent constant. The physical solution of Eq.(3.13) is undamped oscillation with frequency \( 2\omega \), independent of temperature and particle number. This provides a proof of undamped breathing modes in DFT language and poses an exact condition for TDDFT energy functionals.

To visualise the above derivations, we excite a breathing motion and follow its time evolution in a harmonic trap. First we calculate the equilibrium state with trapping frequency \( \omega \), then change the frequency by \( \delta\omega \) for a short duration \( \tau \) and tune it back to \( \omega \) [45]. The breathing motion can be observed through the mean squared cloud radius \( \langle r^2(t) \rangle \). As the bulk viscosity of a unitary Fermi gas is zero [46, 47], the breathing modes are undamped independent of particle number, temperature and \( \delta\omega \). As shown in Fig.3.6, its
frequency is indeed fixed at $2\omega$ \cite{47}, while for a weakly repulsive Fermi gas the breathing modes are damped and the frequency increases with time \cite{48, 49}.

3.4 Conclusion

Based on the recent bold diagrammatic Monte Carlo results \cite{15}, we obtained a new energy functional for a unitary Fermi gas at finite-temperature in the normal phase and studied statics and dynamics of harmonically confined fermions at unitarity within the finite-temperature Kohn-Sham LDA formalism.

Our results show that the commonly used TFA for the UFG breaks down relative to the more accurate KS-LDA for aspect ratios below 0.01 and system sizes below 100 atoms. Caution must be taken in such highly anisotropic traps with mesoscopic numbers of atoms \cite{50}. Unlike the Thomas-Fermi formalism, the KS-DFT approach is useful for studying the collective motion of a unitary Fermi gas.

We also derived a few exact relations for exchange-correlation functionals of a unitary Fermi gas at finite temperature. Energy functionals fulfilling these conditions restore the conventional virial theorem and preserve undamped breathing modes in harmonic confinement.
Chapter 4

Ferromagnetism of the repulsive atomic Fermi gas

The ground-state of a repulsive gas of fermions with contact interactions was first predicted by Stoner [17] in 1933 to be ferromagnetic and a precise value for the critical interaction strength in a homogeneous system was recently obtained with diffusion Monte Carlo simulations [230, 41, 51]. While ultracold fermionic gases should provide a controlled environment to study this phenomenon, the instability of a strongly repulsive gas towards molecule formation has so far prevented experimental realization of this phase. Repulsive fermions on the positive side of the Feshbach resonance live on the meta-stable repulsive branch [52]. When three atoms, one with the opposite spin of the other two, come close to each other, two atoms with opposite spin will form bosonic molecules and the other one carries the binding energy away. The rate of such a process increases rapidly with scattering length. The lifetime of the gas is therefore largely governed by the interaction strength and the spatial overlap between the two spin species.

A recent experiment [19] presented evidence for a possible ferromagnetic state formed after a rapid increase of the scattering length. The nature of this phase has, however, been questioned [20, 21, 53] as the peaks in kinetic energy, cloud size and loss rate observed in [19] are only indirect evidence for ferromagnetic domains [20], and it has been shown that molecule formation is dominant at large interaction strengths [53, 21]. There has been several proposals to reduce the recombination rate by using a polar molecular gas with dipole interactions and positive scattering length [54], narrower Feshbach resonances [55, 56, 53, 21, 57] and fermions with unequal mass [58, 59]. Although these approaches might prove promising in experiments, as has recently been shown in [60], they all change the microscopic physics of the Stoner model.
Here we suggest new strategies which achieve the same goal of stabilising the ferromagnetic phase, yet preserve the microscopic physics and thus pave the way towards experimental realization of Stoner ferromagnetism. Firstly, the lifetime of the system can be increased by reducing the overlap volume between polarized domains with different spins. This can be achieved using elongated traps with larger aspect ratios, but turns out to be insufficient to stabilise the phase by itself. A more effective way is to reduce the critical scattering length, at which ferromagnetism sets in, by imposing a shallow optical lattice. The ferromagnetic transition is determined by a competition between the loss of kinetic energy and gain of interaction energy. Since the optical lattice reduces the kinetic energy scale, the ferromagnetic state is stabilised. A similar effect is found in flat band ferromagnetism [61, 62, 63]

For an intermediate density below half-filling, we first locate the transitions between the paramagnetic, the partially and the fully ferromagnetic phases for the zero-temperature ferromagnetic behavior of a two-component repulsive Fermi gas under the influence of periodic potentials that describe the effect of a simple-cubic optical lattice using continuous-space quantum Monte Carlo. We then verify density functional theory against these results, using an exchange-correlation potential obtained from quantum Monte-Carlo simulations of a uniform Fermi gas. Finally, using density functional theory, we make the setting more realistic by significantly increasing the system size, adding a harmonic trap on top of the optical lattice and including effects of three-body recombination. Our results suggest that three-body recombination is small enough to permit experimental detection of the phase. We also show, using time-dependent density functional theory, that in such a setup ferromagnetic domains emerge rapidly from a paramagnetic initial state.

4.1 Quantum Monte-Carlo study

Performing simulations in continuous space with an external periodic potential, rather than employing single-band discrete lattice models (valid only in deep lattices), allows us to also address the regime of small $V_0$ and to determine the shift of the ferromagnetic transition with respect to the homogeneous gas (corresponding to $V_0 = 0$) [64, 41, 51, 65]. We consider weak and moderately intense optical lattices, where the noninteracting band-gap is small or zero.

We find that the critical interaction strength for the transition between the paramagnetic and the partially ferromagnetic phases (blue circles in Fig. 4.1), as well as the boundary between the partially and fully polarized ferromagnetic phases (black squares), rapidly decreases when $V_0$ increases.
Figure 4.1: Zero-temperature phase diagram at density $nd^3 = 0.75$, as a function of the interactions strength $a/d$ and the optical lattice intensity $V_0/E_R$. The blue circles separate the region of stability of the paramagnetic phase (green) from the partially polarized ferromagnetic phase (yellow). The black squares separate the partially polarized from the fully polarized ferromagnetic phase (red). The violet triangles and the brown dashed line are the corresponding DFT results. Black and blue lines are guides to the eye.
These results strongly support the idea of observing itinerant ferromagnetism in experiments with repulsive gases in shallow optical lattices [237]. At large lattice depths and interaction strengths, however, we observe quantitative discrepancies between QMC and DFT due to the strong correlations which are only approximately taken into account in DFT methods. This regime therefore represents an ideal testbed to develop more accurate exchange-correlation functionals for strongly correlated materials.

This scenario appears to be in contrast with the findings obtained for the single-band Hubbard model, valid for deep lattices and weak interactions, where QMC simulations indicate that the ground-state is paramagnetic [66] (at least up to filling factor 1/4) and stable ferromagnetism has been found only in the case of infinite on-site repulsion [67, 68, 69].

Since for a large optical lattice intensity and weak interactions our results agree with Hubbard model simulations, these findings concerning the ferromagnetic transition indicate that the Hubbard model is not an appropriate description for the strongly repulsive Fermi gas in moderately deep optical lattices and that terms beyond on-site repulsion and nearest neighbor hopping play an essential role. It also suggests that the possibility of independently tuning interparticle interactions and spatial inhomogeneity, offered by our continuous-space Hamiltonian, is an important ingredient to explain itinerant ferromagnetism.

4.2 Dynamics and three body recombination

To study the system in a more experimentally relevant setting, we use Kohn-Sham density functional theory [23, 25], where the exchange-correlation energy is treated within a local spin density approximation (LSDA). LSDA has been widely used for materials simulations and more recently for ultracold atomic gases [70, 71, 72, 73]. The LSDA exchange-correlation functional is obtained by solving a uniform system at zero temperature with diffusion Monte-Carlo [41, 51], where interactions between fermionic atoms with opposite spin are modelled by a hard-sphere potential with scattering length $a_s$.

Before discussing our proposal to stabilise the ferromagnetic phase we investigate whether thermal fluctuations, which can significantly affect the stability of the ferromagnetic phase [74], might be responsible for the absence of a stable ferromagnetic phase in experiments. To quantify the effect of non-zero temperature we employ finite-temperature density functional theory with a zero temperature exchange-correlation correction [75, 76, 77, 78]. The resulting phase diagram, presented in Fig. 4.2, is in general agreement
4.2. DYNAMICS AND THREE BODY RECOMBINATION

Figure 4.2: Phase diagram of the homogeneous repulsive Fermi gas as a function of temperature $T/T_F$ and interaction strength $k_F a_s$. The white dashed line indicates the paramagnet-ferromagnetic phase transition and the colour scale the polarization $P = (n_{\uparrow} - n_{\downarrow})/(n_{\uparrow} + n_{\downarrow})$.

with previous results [79, 80]. We observe almost full polarization in the currently experimentally accessible temperature regime $T \sim 0.25 T_F$ [57, 21] and thermal fluctuations are thus not the dominant mechanism destabilising the ferromagnetic phase.

An equally important question is that of the time scale over which ferromagnetic domains form from an initially paramagnetic state [81], which should be within the capabilities of current experimental setups for the observation of ferromagnetic domains to remain plausible. We address this issue within the time-dependent Kohn-Sham formalism [30] ignoring the effect of three-body recombination. The system is evolved in pancake shaped harmonic confinement in the presence of thermal noise after a quench of the interaction strength $k_F^0 a_s$ to 1.2. Already after $t \sim 250 t_F$ ferromagnetic domains with a feature size $\sim 10k_F^{-1}$ form (Fig. 4.3), which is within the resolution currently achievable in the lab. The ferromagnetic phase is also significantly more stable with a total recombination rate reduced by a factor $\sim 3$ relative to the initial paramagnetic gas. In the experiment [21], more than half of the particles remain in the meta-stable repulsive branch after $250 t_F$. However, ferromagnetic domains were not observed indicating that three-body recombination has more significant effects beyond reducing the
fraction of meta-stable fermions in the system. We thus focus on three-body recombination and determine its effect on the stability of the ferromagnetic phase. The loss rate $\Gamma = -\partial_t N/N$ ($N$ is the total number of particles in the system) is inversely proportional to the lifetime of the system and can be computed as [82]

$$\Gamma \sim a^3 \sum_\sigma \int dr \int_{|r'-r|<a_s} dr' \varepsilon_F(r) g_{\sigma\bar{\sigma}\sigma}(r, r, r')$$

(4.1)

where $\varepsilon_F(r) = \hbar^2 (3\pi^2 (n_\uparrow + n_\downarrow))^{2/3}/2m$ is the local Fermi energy, $n_\sigma(r)$ is the local density of each spin, $m$ the atom mass and $g_{\sigma\bar{\sigma}\sigma}(r, r, r') = \langle \hat{\psi}_\sigma(r) \hat{\psi}_\sigma^\dagger(r') \hat{\psi}_{\bar{\sigma}}(r') \hat{\psi}_{\bar{\sigma}}(r) \hat{\psi}_\sigma(r) \rangle$ is the three-body correlation function which gives the probability of finding three particles with spin $\bar{\sigma}\sigma\sigma$ [83] at locations $rrr'$ respectively. Here $\bar{\sigma}$ refers to the opposite spin of $\sigma$. In this way the total loss rate is in units of energy and the microscopic contribution, represented by the pre-factor $a^3_s$ together with the integration over $r'$, is decoupled from the many-body background given by the correlation function. The three-particle correlation function is commonly [79, 84, 85] treated phenomenologically as $g^0_{\bar{\sigma}\sigma\sigma}(r, r, r') = n_{\bar{\sigma}}(r)n_\sigma(r)n_\sigma(r')$, where the
4.2. DYNAMICS AND THREE BODY RECOMBINATION

(a) Total density $n_\uparrow + n_\downarrow$

(b) Magnetisation $n_\uparrow - n_\downarrow$

(c) Loss rate $\Gamma$

Figure 4.4: Harmonic confinement with aspect ratio $\lambda = 5$, $k_0^2a_s \sim 1.4$ and $N_\uparrow = N_\downarrow = 969$. The $z$ and $r$ axes are along the horizontal and vertical directions respectively.

density is assumed to be constant within the integral over $r'$. This form disregards quantum mechanical corrections due to exchange effects and violates the Pauli principle. Within the Kohn-Sham formalism a more accurate representation is

$$g_{\sigma\sigma\sigma}(\mathbf{r}, \mathbf{r}, \mathbf{r}) = g_{\sigma\sigma\sigma}^0(\mathbf{r}, \mathbf{r}, \mathbf{r}) - n_\sigma(\mathbf{r})|\rho_\sigma(\mathbf{r}, \mathbf{r}')|^2$$  \hspace{1cm} (4.2)

where $\rho_\sigma(\mathbf{r}, \mathbf{r}')$ is the one-body density matrix. The Pauli principle is restored in this formulation as $g_{\sigma\sigma\sigma}(\mathbf{r}, \mathbf{r}, \mathbf{r}) \equiv 0$, thus leading to a better estimate of three-body losses. As molecule formation in the gas is a three-body process and requires both species of fermions to be close to each other, the recombination rate (4.1) is determined by the total volume where the two spin species are mixed. In a polarised system this amounts to the overlap between ferromagnetic domains at their boundaries, whose structure strongly depends on the external confining potential. To further investigate this effect, we impose a cigar shaped harmonic potential, which has been used in a number of recent experiments [21, 57]. With cylindrical symmetry around the $z$-axis
30

CHAPTER 4. FERROMAGNETISM OF THE REPULSIVE ATOMIC FERMI GAS

Figure 4.5: Normalised recombination rate $\Gamma/\Gamma_c$ in harmonic confinement for aspect ratios $\lambda = \{2, 5, 10, 14\}$ and $N_\uparrow = N_\downarrow = 560$ at different interaction strengths. *Inset:* dependence of $\Gamma/\Gamma_c$ on the aspect ratio for $k_F^0a_s \sim 1.6$. From simulations of different system sizes we checked that the results are not sensitive to the number of atoms in the system.

the potential is in the form $V_{HO}(r, z) = 1/2(\omega_r^2r^2 + \omega_z^2z^2)$ with aspect ratio $\lambda = \omega_r/\omega_z$. We will consider the spin-balanced case $N_\uparrow = N_\downarrow = N/2$ and normalise the loss rate $\Gamma$ by the critical recombination rate $\Gamma_c$, above which the system is unstable. This value can be estimated from the experimental observation of a maximum tolerable $k_Fa_s \sim 0.4$ for $^6$Li atoms in an optical dipole trap [57].

With equal number of spin-up and down particles, the ferromagnetic ground state exhibits a domain wall across the middle of the trap, as shown in Fig. 4.4b. Due to repulsion between the two polarized domains there is a drop in total density along this region with a width $\sim 0.5k_F^{-1}$ (see Fig. 4.4a), which is, however, below the resolution of current experiments [21]. As shown in Fig. 4.4c), particle loss occurs predominantly at the domain wall in the center of the trap and in the low-density halo on the surface of the clouds where the gas remains unpolarised due to low densities. One can thus expect that larger aspect ratios may reduce losses due to a smaller area of the domain wall. To investigate this quantitatively we present the results of our calculations for the recombination rate in Fig. 4.5 as a function of interaction strength $k_Fa_n$. Starting from a non-interacting gas in the param-
4.3. CONCLUSION

In conclusion, while in shallow lattices there is good agreement between QMC and Kohn-Sham LSDA, the regime of deep lattices and strong interactions represents a new test bed to develop more accurate exchange-correlation

agnetic ground state, the recombination rate increases until phase separation due to ferromagnetism sets in and a maximum is reached at \( k_F a_s \sim 1.1 \), after which total losses are again reduced. A similar dependence of the loss rate on scattering length was observed in the experiment [19] and theoretical studies [85, 86]. As the trap is compressed around the \( z \)-axis at constant volume \( \tilde{\omega} = (\omega_x^2 \omega_z)^{1/3} \) and system size \( N \) (see inset of Fig. 4.5), the aspect ratio increases and the recombination rate is indeed significantly reduced. However, with a six times lower recombination rate at \( \lambda = 14 \) compared to \( \lambda = 2 \), the loss rate is still above the critical value \( \Gamma_c \).

We now turn to the most promising way of stabilising the ferromagnetic phase, by reducing the critical scattering length of the ferromagnetic phase transition. This can be achieved by imposing a shallow optical lattice [73], which reduces the kinetic energy loss at the phase transition and thus favors the ferromagnetic state. Figure 4.6 shows the phase diagram at quarter filling \( \bar{n} = 0.5 \) in a shallow 3D cubic optical lattice \( V_{OL}(r) = V_0 \sum_{i=1}^{3} \sin^2(\pi r_i/d) \), where \( V_0 \) is in units of recoil energy \( E_R = \frac{\hbar^2 \pi^2}{2md^2} \) and \( d \) is the lattice constant. At quarter filling \( \bar{n} = 0.5 \), the maximum tolerable \( k_F a_s \sim 0.4 \) corresponds to \( a_s = 0.16d \), which is nearly half the critical scattering length predicted by diffusion Monte Carlo [41, 51].

We see that a ferromagnetic phase exists at this scattering length for a lattice depth of about \( V_0 \sim 2.5 E_R \). However, the loss rate is likely significantly affected by the presence of a periodic potential. To address this issue we calculate the recombination rate of the paramagnetic state in an optical lattice by integrating over the unit cell Eq.(4.1). The loss rate initially grows as the lattice is ramped up since the density increases at the center of the unit cell (inset of Fig. 4.6). In very deep lattices the loss rate is reduced when one approaches the regime where the physics is well described by the single band Hubbard model and triple occupation of a lattice site is reduced. In Fig. 4.6 we show contour lines of constant loss rate \( \Gamma \) and indicate by grey shading the regime where we expect three body losses to be above the critical experimental value. We find that despite the increased three-body losses in a shallow lattice, the ferromagnetic phase is stable in a wide region of the phase diagram. Our calculations have not taken into account the quantum Zeno effect [87] which will further suppress three body recombination.

4.3 Conclusion

In conclusion, while in shallow lattices there is good agreement between QMC and Kohn-Sham LSDA, the regime of deep lattices and strong interactions represents a new test bed to develop more accurate exchange-correlation
Figure 4.6: Contours of constant normalised loss rate $\Gamma/\Gamma_c$ at quarter filling $\bar{n} = 0.5$ in an optical lattice. In the shaded region $\Gamma > \Gamma_c$. Red (yellow) region denotes the fully (partially) polarized ferromagnetic phase. **Inset:** Normalised loss rates $\Gamma/\Gamma_c$ for $a_s = \{0.12, 0.14, 0.16\}$ at different lattice depths. The dotted line is $\Gamma_c$. From simulations at different fillings below $\bar{n} = 1.0$, we checked that the physics remains qualitatively the same.
4.3. CONCLUSION

functionals, which is an outstanding open challenge in material science [24]. Furthermore, our results show that moderately intense optical lattices are favorable for experimental realization of ferromagnetism, also due to a faster thermalization rate compared to very deep lattices.

Although larger aspect ratios in harmonic confinement can significantly reduce the total recombination rate, the gas remains unstable. However, in an optical lattice the ferromagnetic phase extends down to small enough scattering lengths where three-body recombination is below the critical value. The idea of stabilising the ferromagnetic phase in experiments is also of fundamental significance: by introducing an optical lattice we increase the effective mass of the atoms. This reduces their kinetic energy and hence allows the ferromagnetic phase to be stabilised at much smaller interaction strengths, where three body losses are less significant. Experimental verification of our results will be solid confirmation of the Stoner model of itinerant ferromagnetism.
Part II

Quantum computing
Chapter 5

Introduction

The work in this part was done as part of an internship at QuArC, Microsoft Research and is based on the paper [234] and patent [239].

Although the theoretical foundations of quantum computing have been around for decades, only in the last few years have the first working prototypes become available and interest in the field has increased rapidly, both from industry and research groups around the world.

The fundamental building blocks of a quantum computer are qubits, which hold the quantum state. A universal quantum computer, that is, a quantum Touring machine, can be used to do anything a classical Touring machine can do and vice versa. However, by making use of quantum entanglement and superposition, a quantum computer could do some things much more efficiently, even exponentially faster than its classical counterpart.

However, despite significant theoretical advantages over classical computers, two fundamental challenges remain. The first challenge is finding useful applications and corresponding algorithms which can make use of the quantum advantage. Once such algorithms have been identified, the second challenge is to build the hardware.

Perhaps the most promising application of quantum computers is simulating quantum systems. As Feynman noted in his seminal paper [1], this requires exponential resources on a classical computer, but scales only polynomially on a quantum machine. For example, simulating Fermions and Bosons [88], chemical reactions [89] and TQFTs [90].

Another fundamental application is running quantum algorithms. A quantum algorithm is a quantum circuit which operates on an input state and generally ends with a measurement. This can be done on an analog or a digital machine.
Analog quantum computing has been popularised by the adiabatic quantum annealing algorithm [91], where the idea is to obtain the ground state of a target Hamiltonian by starting in the ground state of a known Hamiltonian and slowly transitioning towards that target. As of today, however, such annealers have not shown any advantage [92].

The digital, or quantum circuit, model of computation is a more promising candidate to beat classical computers. Although such a computer has not yet been realised in hardware beyond a few qubits in a lab [93], a range of quantum algorithms have been invented which could significantly outperform their classical counterparts. Some notable examples are Grover’s algorithm for quadratically faster database search [94], amplitude amplification [95], which can be viewed as a generalisation of Grover’s algorithm, Shor’s algorithm for exponentially faster integer factorisation [96], Simon’s [97] and Deutsch-Jozsa [98] algorithms which have an exponential advantage on more artificial problems.

One of the fundamental tools which serves a basis for many quantum algorithms is the phase estimation algorithm, which computes the eigenvalue of a given eigenvector of a unitary gate. A range of different approaches have been developed for phase estimation in recent years [99, 100, 101, 102]. All of them, however, require at least one calibrated ancillary qubit. Keeping such a qubit calibrated remains a significant challenge, despite lots of efforts dedicated to isolate the system from the environment and to develop error correcting codes.

We introduce a novel method which can estimate the spectral gaps and generally the full eigenspectrum of a quantum system. At its core, it uses Bayesian inference coupled with rejection sampling to efficiently infer information about the system. Unlike traditional phase estimation, our approach does not require ancillary qubits nor well characterised gates. Instead, it only requires the ability to perform approximate Haar–random unitary operations, applying the unitary whose eigenspectrum is sought out and performing measurements in the computational basis. We also discuss application of these ideas to in-place amplitude estimation, quantum device calibration and bootstrapping a quantum system from a completely uncalibrated state.
Chapter 6

Randomized gap and amplitude estimation

In recent years a host of methods have been developed for performing phase estimation in quantum systems [99, 100, 101, 102]. These methods, driven by demand in quantum algorithms and metrology, have provided ever more efficient means of learning the eigenvalues of a quantum system. Through the use of sophisticated ideas from optimization and machine learning, recent phase estimation methods come close to saturating the Heisenberg bound [103, 104, 102, 101], which is the ultimate performance limit for any phase estimation algorithm.

Despite the near-optimality of existing methods, there are a number of avenues of inquiry that still remain open surrounding phase estimation. Importantly, recent work has begun to look at operational restrictions on phase estimation including limited visibility in measurements, de-coherence and time-dependent drift. Such generalisations are especially important as we push towards building a scalable quantum computer and are faced with the challenge of characterising the gates in a massive quantum system that is imperfectly calibrated.

One important restriction that has not received as much attention is the issue that traditional phase estimation algorithms require entangling the quantum device with one (or more) ancilla qubits. This need to couple the system with an external qubit precludes ideas from phase estimation to be applied to single qubit devices. Even if ancilla qubits are available, such characterisation methods would entail the use of entangling gates which are very costly in many quantum computing platforms.

Our work provides a way to circumvent this problem for small quantum devices. It allows an experimentalist to learn the spectral gaps in an uncharacterised device with an amount of experimental time that is proportional to
that of phase estimation without requiring any ancillary qubits. The idea of our approach is reminiscent of randomised benchmarking [105, 106, 107, 108] in that we use random operations to extract information about the underlying dynamics of the system.

We consider three applications for this method, which we call randomised gap estimation:

1. Eigenspectrum estimation for small quantum systems that lack well characterised gates.


3. Control map learning for small quantum devices.

This last application is interesting in the context of quantum bootstrapping [109] because it gives an inexpensive way of calibrating a set of one and two qubit gates at the beginning of a bootstrapping protocol. This is significant because an important caveat in bootstrapping is that while a trusted simulator can be used to quickly learn how to control an uncalibrated system, calibrating that trusted simulator requires exponentially more measurements and polynomially more experimental time than using traditional methods. Thus our techniques can, under some circumstances, dramatically reduce the total cost of a bootstrapping protocol.

Our paper is laid out as follows. We first provide a review of Bayesian inference in Section 6.1 and discuss the method for approximate Bayesian inference that we use in our numerical studies. We then introduce randomised gap estimation in Section 6.2 and discuss the challenges faced when applying the method to high-dimensional systems. We then show how the adiabatic theorem can be used to eliminate this curse of dimensionality for certain systems in Section 6.4. We then apply randomised gap estimation to the problem of amplitude estimation in Section 6.5 and also to learning a map between experimental controls and the system Hamiltonian for a quantum device Section 6.6 before concluding.

6.1 Bayesian inference

Bayesian inference is a widely used method to extract information from a system. The goal of Bayesian inference is to compute the probability that a hypothesis is true given evidence $E$ and a set of prior beliefs about the hypotheses. These prior beliefs are represented as a probability distribution known as a prior. This prior, along with evidence $E$, can be thought of as an
6.1. BAYESIAN INFERENCE

input to the inference algorithm. The output of the algorithm is the posterior distribution which is given by Bayes’ theorem as

$$P(x|E) = \frac{P(E|x)P(x)}{\int P(E|x)P(x)dx},$$  \hspace{1cm} (6.1)

where $P(E|x)$ is known as the likelihood function. The posterior distribution output by this process is then used as a prior distribution in online inference algorithms and this process of generating a new prior distribution using Bayes’ rule is known as a Bayesian update.

As an example of how Bayesian inference plays a role in eigenvalue estimation, consider the traditional approach used for iterative phase estimation. In iterative phase estimation, one wishes to learn the eigenvalues of a unitary operation $U$ using the following circuit:

$$|0\rangle \xrightarrow{H} e^{-iM\theta Z} \xrightarrow{H} E$$

where $M$ and $\theta$ are user–specifiable experimental parameters, $H$ is the Hadamard matrix and $Z$ is the Pauli–$Z$ operator. Let $|\phi\rangle$ be an eigenstate such that $U|\phi\rangle = e^{i\lambda}|\phi\rangle$. The likelihood function for this circuit is then

$$P(E = 0|\lambda; \theta, M) = \cos^2(M(\theta - \lambda)).$$  \hspace{1cm} (6.2)

The problem of phase estimation is thus the problem estimating $\lambda$ given a sequence of different experimental outcomes. This can be done using (6.1) starting from an initial prior distribution for $\lambda$ that is uniform over $[0, 2\pi)$. A further benefit is that the posterior variance provides an estimate of the uncertainty in the inferred phase as well as an estimate of the most likely value of $\lambda$. This approach is considered in [100, 101] wherein exact Bayesian inference is found to perform extremely well both in settings where $\theta$ and $M$ are chosen adaptively as well as non–adaptively.

If $P(x)$ has support over an infinite number of points then exact Bayesian inference is usually intractable and discretisations are often employed to address this problem. Such discretisations include particle filter methods and sequential Monte Carlo methods [110, 111, 112]. Here we employ a newly developed approach known as rejection filter inference [101, 113].

The idea behind Rejection filtering is to use rejection sampling to convert an ensemble of samples from the prior distribution to a smaller set of samples from the posterior distribution. Specifically, if evidence $E$ is observed and we draw a sample from the prior distribution and accept it with probability equal
to \( P(E|x) \) then the probability that hypothesis \( x \) is accepted as a sample is from Bayes’ theorem
\[
P(E|x)P(x) \propto P(x|E)
\] (6.3)
Therefore, the samples that pass through the rejection filter are distributed according to the posterior distribution.

Although this process allows us to sample from the posterior distribution, it is not efficient. This is because every time we perform an update, we will, on average, reject a constant fraction of samples. This means that the number of samples kept will shrink exponentially with the number of updates. We can, however, make this process efficient by fitting these samples to a family of distributions and then draw a new set of samples from this model distribution for the next update. This ability to regenerate samples allows rejection filtering inference to avoid this catastrophic loss of support for the approximation to the posterior.

There are a number of models for the prior and posterior that can be considered. Here we use a unimodal Gaussian distribution. Gaussian models for the posterior distribution provide a number of advantages. First, they are parametrised by the posterior mean and covariance matrix which give an estimate of the true hypothesis and the uncertainty in it. Furthermore, these quantities are easy to estimate from the accepted samples and can be computed incrementally, which allows our algorithm to be executed using near-constant memory.

There are several advantages to using rejection filtering for approximate inference. Specifically, it is very fast, easy to parallelise, can be implemented using far less memory than particle filter or sequential Monte-Carlo methods. Perhaps most importantly, it is also substantially easier to implement than traditional particle filters. Rejection filtering has also been successfully used in phase estimation algorithms [101], where the use of rejection filtering leads to substantial reductions in the experimental time needed to perform the inference relative to Kitaev’s phase estimation algorithm [99] and information theoretic phase estimation [100].

### 6.2 Randomized gap estimation

Traditional approaches for learning the eigenspectrum of a Hamiltonian require ancillary qubits and well characterised gates. Here we present an approach, which we call randomised gap estimation, to efficiently estimate the eigenspectrum of a small system with no ancillary qubits and potentially poorly characterised gates. The idea behind this approach is to use Bayesian
6.2. RANDOMIZED GAP ESTIMATION

Algorithm 1 Bayesian inference algorithm for eigenphases

**Input:** Set of unitary operators \( \{U : j = 1 : K\} \), evolution time \( t \), prior over gaps \( P(\Delta) \).

Prepare state \( |0\rangle \in \mathbb{C}^N \).
Randomly select \( U \) from set of unitary operations.

\[
|0\rangle \leftarrow U^\dagger e^{-iHt}U|0\rangle.
\]

Measure state and \( E \leftarrow 0 \) if the result is '0' otherwise \( E \leftarrow 1 \).

Use approximate Bayesian inference to estimate \( P(\Delta|E) \propto P(\Delta) \cdot \int P(E|\Delta; t, U)\mu(U)\,dU \).

**return** \( P(\Delta|E) \).

inference in concert with random evolutions to infer the gaps between the eigenphases of a unitary operator.

In the first step, the state

\[
|\Psi\rangle := U|0\rangle = \sum_k \beta_k |v_k\rangle
\]

(6.4)
is prepared, where \( |v_k\rangle \) is the eigenvector corresponding to eigenvalue \( \lambda_k \) of \( H \) in an arbitrary ordering. Here \( \beta_k \) are unknown parameters that depend not only on the random unitary chosen, but also the eigenbasis of \( H \). For low-dimensional systems exact methods for drawing \( U \) uniformly according to the Haar measure are known. More generally it, however, suffices to draw \( U \) from a unitary 2–design (see Appendix 6.3). For example, a single qubit system \( U \) has an Euler angle decomposition of

\[
U = R_z(\phi)R_x(\theta)R_z(\psi),
\]

(6.5)
up to an irrelevant global phase.

Next, we evolve the system according to \( e^{-iHt} \) for a controlled time \( t \).
This results in the state

\[
e^{-iHt}|\Psi\rangle = e^{-iHt}U|0\rangle.
\]

(6.6)

Finally, \( U^\dagger \) is applied and a measurement in the computational basis is performed, which returns '0' with probability

\[
P(0|H; t, U) := |\langle 0|U^\dagger e^{-iHt}U|0\rangle|^2
\]
\[
= \left( \sum_k |\beta_k|^2 \cos (\lambda_k t) \right)^2 + \left( \sum_k |\beta_k|^2 \sin (\lambda_k t) \right)^2
\]
\[
= \sum_{ij} \cos ((\lambda_i - \lambda_j)t)|\beta_i|^2|\beta_j|^2 := P(0|\Delta; t, U).
\]

(6.7)
CHAPTER 6. RANDOMIZED GAP AND AMPLITUDE ESTIMATION

Note that the gaps $\Delta_{ij} = \lambda_i - \lambda_j$ cannot be easily learned from this expression because of the unknown $\beta_i$ and $\beta_j$ terms. Haar averaging provides a solution to this problem.

Given an unknown Haar–random unitary $U$, the likelihood of measuring '0' is given by the law of conditional probability

$$P(0|\Delta; t) = \int P(0|\Delta; t, U)\mu(U)dU,$$

(6.8)

where $\mu$ is the Haar–measure over $U(N)$. This probability is distinct from the likelihood that would be used if the user knew, or was capable of computing, $P(0|H; t, U)$ for the particular $U$ that was used in the experiment.

If we define $\Delta$ to be a matrix such that $\Delta_{i,j} := \lambda_i - \lambda_j$, this Haar average evaluates to

$$P(0|\Delta; t) = N\langle|\beta_i|^4\rangle + \langle|\beta_i|^2|\beta_j|^2\rangle\sum_{i\neq j} \cos (\Delta_{ij}t)$$

$$= \frac{2}{N+1}\left(1 + \frac{1}{N}\sum_{i>j} \cos (\Delta_{ij}t)\right).$$

(6.9)

Eq. (6.9) provides a likelihood function that can be used to perform Bayesian inference. In particular, Bayes’ rule states that if a binary experiment is performed wherein the only two outcomes are $|0\rangle$ and $|v\rangle \neq 0\rangle$, the latter occurs with probability $1 - P(0|H; t)$. If we define our prior distribution over eigenvalues to be $P(\lambda)$, then given a measurement of '0' is recorded, Bayes’ rule states that the posterior distribution is

$$P(\lambda|0; t) = \frac{2\left(1 + \frac{1}{N}\sum_{i>j} \cos (\Delta_{ij}t)\right) P(\lambda)}{(N+1)P(0)},$$

(6.10)

where $P(0)$ is a normalisation factor. We outline this approach in Algorithm 1. Thus Bayesian inference allows the gaps to be learned from such experiments. Since the likelihood of observing '0' in (6.10) scales inversely with $N$, we expect that the number of experiments required to learn the spectrum should grow at least linearly with $N$.

The Cramér–Rao bound allows us to formalise this argument by bounding the minimum number of experiments and/or experimental time needed ensure that the variance of $\Delta_{ij}$ is sufficiently small [114]. If we assume that $R$ experiments are performed, each with $D \in O(1)$ outcomes and evolution time at least $t$, then the elements of the Fisher matrix are

$$I_{ij,kl} = \sum_{d_1=0}^{D-1} \cdots \sum_{d_R=0}^{D-1} \partial_{\Delta_{ij}} \prod_{q=1}^{R} P(d_q|\Delta)\partial_{\Delta_{kl}} \prod_{q=1}^{R} P(d_q|\Delta)$$

$$= O\left(\frac{R^2t^2}{N^2}\right).$$

(6.11)
which through the use of the Cramér–Rao bound implies that the variance of any unbiased estimator of $\Delta_{ij}$ after all $R$ experiments scales at least as $\Omega(N^2/R^2t^2)$.

While this shows that the number of experiments needed to learn the gaps grows at least linearly with $N$, the uncertainty in the optimal unbiased estimator also shrinks as $\Omega(1/T)$, where $T$ is the total evolution time. The error scaling is hence quadratically smaller than would be expected from statistical sampling. This opens up the possibility of very high precision frequency estimates for small quantum devices.

### 6.2.1 Numerical tests

While the prior argument suggests that the error should scale as $1/T$ for randomised gap estimation, an important question remains regarding how well it actually scales in practice. We assess this by using rejection filtering to estimate the gaps and use the particle guess heuristic, discussed in [115, 116, 109] and also the appendix. We use this approach because it is fast, accurate and more importantly easier to implement than sequential Monte–Carlo methods [117, 118].

Although performing rejection filtering to infer the gaps may seem straightforward, a complication emerges that makes it conceptually more challenging to apply the technique directly. To see this, consider a Gaussian prior over the eigenvalue gaps for an $N$–dimensional system. If the eigenvalue gaps are drawn independently, the result will with high probability have inconsistent eigenvalue gaps. By inconsistent we mean that if $\Delta_{43} = \lambda_4 - \lambda_3$ and $\Delta_{32} = \lambda_3 - \lambda_2$ are eigenvalue gaps, then $\Delta_{42} = \lambda_4 - \lambda_2 = \Delta_{43} + \Delta_{32}$ must also be one of the eigenvalue gaps for the system. If $\Delta_{43}$ and $\Delta_{32}$ are chosen independently from a Gaussian prior, then it is very unlikely that their sum is also. Hence, gaps chose independently from the prior distribution will not correspond to a feasible set of eigenvalues. There are many ways that this self-consistency constraint could be imposed in the inference step, but perhaps the simplest approach is to use Bayesian inference to learn a set of consistent eigenvalues for the system, rather than gaps directly. We take the lowest eigenvalue to be zero without loss of generality and then define the error in the inference, given that rejection filtering reports a mean of $\mu$ and covariance matrix of $\Sigma$ over the eigenvalues, to be

$$\Delta = \min \left\{ \sum_i |\lambda_i - \mu_i|, \sum_i |\lambda_i - (\max\{\mu_i\}_i - \mu_i)| \right\}$$  \hspace{1cm} (6.12)

Here the minimisation is meant to remove a mirror symmetry in the gaps that naturally arises in this problem. The uncertainty is computed as $\sqrt{\text{Tr}(\Sigma)}$.  


We find numerically that Bayesian inference can rapidly estimate the eigenvalues of an unknown Hamiltonian, up to a degeneracy, in one and two qubit systems (see Figure 6.1). In particular, we see that the error shrinks exponentially with the number of measurements before being ultimately limited by machine-\(\epsilon\). This illustrates that randomised gap estimation can learn the eigenvalue gaps for low-dimensional systems using exponentially less data than would ordinarily be required by statistical sampling.

Note that the relation between the eigenvalues and gaps is a well known open problem in computer science called the Turnpike Problem [119]. If there are no degenerate gaps, a solution to the turnpike problem is unique up to a reflection of the spectrum about its midpoint. However, in the presence of degeneracies the number of possible solutions could scale slightly faster.
6.2. RANDOMIZED GAP ESTIMATION

than linear in $N$ and the eigenspectrum can in this case only be narrowed down to one of these solutions. Moreover, while a backtracking algorithm can find a solution for the typical case in time $O(N^2 \log(N))$, the worst case complexity is $O(2^N N \log(N))$ [119]. This suggests that Bayesian inference will not always find a solution in time that is polynomial in $N$, which itself is exponential in the number of qubits. However, for the two level spectrum of a single qubit, the phase can be determined unambiguously.

As mentioned above, we use sets of eigenvalues as our hypotheses for Bayesian inference. Without loss of generality we can shift the eigenspectrum such that $\lambda_1 = 0$ and we hence have $N-1$ variables, where $N$ is the number of levels.

For the Bayesian inference step, we used rejection filtering with a threshold of 10000 accepted samples. We found that drawing parameters from the model until 10000 samples are accepted provides sufficient accuracy to represent the likelihood function in these experiments. We use a standard unimodal multivariate Gaussian as a model for the prior distribution on hypothesis space. While this unimodal distribution is incapable of representing a multi-modal distribution (which may occur in cases where degenerate solutions exist), multimodal models can also be used to improve learning. However, for our experiments the mean and covariance of all modes quickly converged to the same value and the accuracy was hence no better than in the unimodal case.

The initial prior distribution is taken to be a Gaussian with mean and standard deviation set to be those appropriate for Gaussian Unitary Ensemble (GUE) Hamiltonians. In particular, we find these values using the gap distribution of GUE Hamiltonians and set the mean eigenvalue to be that found by adding $N-1$ random gaps drawn from the distribution appropriate for GUE Hamiltonians to $\lambda_1 = 0$. Similarly the covariance matrix for the $N-1$ unknown eigenvalues was set to be variances equivalent to those found for each of the eigenvalues of GUE Hamiltonians. While this distribution does not accurately model the true prior distribution of the eigenvalues, it has sufficient overlap to be able to zero in on the correct eigenvalues (up to degeneracies that cannot be resolved by randomised gap estimation alone).

Evolution times are chosen to be $t = 1/(2\Gamma)$, where $\Gamma$ is the uncertainty which we define to be the trace of the inferred covariance matrix. This heuristic was introduced in previous works [115, 116, 109] as the particle guess heuristic and is shown to be nearly optimal for frequency estimation problems. The prefactor of $1/2$ was found to work well empirically, but more optimal choices may exist for this problem.

Since degeneracies in the eigenvalues can emerge in our problem that prevent correct inference of the gaps, we resort to modifying the likelihood
function in order to forbid particular Hamiltonian models. Specifically, we force the eigenvalues to be sorted and positive by introducing an additional sign-term in the likelihood function

\[ \langle L \rangle = \frac{2}{N+1} \left( 1 + \frac{1}{N} \sum_{i>j} \text{sign}(\Delta_{ij}) \cos(\Delta_{ij} t) \right) \]  \hspace{1cm} (6.13)

where the sign-function is +1 if its argument is positive and −1 otherwise and \( \Delta_{ij} = \lambda_i - \lambda_j \). Hence, this likelihood will align with the experimental likelihood at all time-steps \( t \) only if \( \lambda_i > \lambda_j \). As this modified likelihood can for some sets of parameters be larger than unity or smaller than zero, we do an additional truncation step such that \( 0 \leq L \leq 1 \).

The median error and uncertainty reported are computed across 1000 instances drawn from the Gaussian Unitary Ensemble. This number was sufficient to make the sample error in the estimates graphically insignificant.

For higher-dimensional systems achieving the same task is expected to be much more daunting and the error metric in (6.12) should be the minimum over all consistent re-orderings of the spectrum. Such problems can, however, be sidestepped in systems with a gapped adiabatic path connecting the Hamiltonian to a efficiently diagonalizable Hamiltonian, as we show below.

### 6.3 Implementing random unitaries

In low dimensional systems there are a number of methods that can be used to perform random or pseudo-random unitary operations. The most direct approach is to exploit an Euler angle decomposition of \( U(N) \) to generate the unitary. For example, for \( U \in SU(2) \), we can implement such a unitary via

\[ U = R_z(\beta)R_y(\gamma)R_z(\delta). \]  \hspace{1cm} (6.14)

In order to pick such a unitary uniformly according to the Haar measure it suffices to pick \( \theta := \gamma/2 \in [0, \pi/2] \), \( \phi := (\beta+\delta)/2 \) mod \( 2\pi \) and \( \psi := (\beta-\delta)/2 \) mod \( 2\pi \) chosen according to the following probability density function [120]

\[ dP(\phi, \theta, \psi) = \sin(2\theta)d\phi d\theta d\psi. \]  \hspace{1cm} (6.15)

Such constructions also are known for higher-dimensional systems [120].

In practice, Haar-random unitaries are hard to exactly implement on a quantum computer. This task is even more challenging in an uncalibrated device. Fortunately, efficient methods for implementing pseudo-random unitaries are well known. Here we use the definition of \( t \)-design given by Dankert
et al [105], wherein a unitary $t$–design on $N$ dimensions is taken to be a finite set of unitary operations such that the average of any polynomial function of degree at most $t$ in the matrix elements and their complex conjugates agrees with the Haar average. In particular, let $P_{(t,t)}(U_k)$ be such a polynomial function and assume that there are $K$ elements in this design. Then

$$
\frac{1}{K} \sum_{k=1}^{K} P_{(t,t)}(U_k) = \int P_{(t,t)}(U)\mu(U)dU,
$$

(6.16)

where $\mu$ is the Haar–measure.

More generally, one can also consider the notion of an $\epsilon$–approximate $t$–design. This notion differs from a $t$–design in that strict equality is not required and a discrepancy of at most $\epsilon$ in the diamond–distance is permitted. Specifically, let $G_W(\rho) = \sum_i U_i^{\otimes k} \rho(U_i^{\dagger})^{\otimes k}$ be a super-operator corresponding to twirling the input operator over the elements of the design and $G_H(\rho) = \int U^{\otimes k} \rho(U^{\dagger})^{\otimes k} \mu(U)dU$ be the same quantity averaged over the Haar–measure. Then the set of unitary operators forms a $\epsilon$–approximate $k$–design if $\|G_W - G_H\|_\diamond \leq \epsilon$, where the diamond norm is discussed in detail in [121].

This definition implies that the Haar–expectations of the terms in (6.9) can be found using a 2-design since all such terms are quadratic in the matrix elements of $U$ and their complex conjugates. The remaining question is how to form such a design.

Dankert et al. show that random Clifford operations can be used to form exact and $\epsilon$–approximate 2-designs using $O(n^2)$ and $O(n \log(1/\epsilon))$ gates respectively [105]. These results are sufficient in settings where the device has access to well characterised Clifford gates. If the system does not have such gates, the result of Harrow and Low [122] can be applied to show that $\epsilon$–approximate $t$–designs can be formed out of random sequences of gates taken from any universal gate set. The number of gates needed to generate such a pseudo–random unitary is $O(n(n + \log(1/\epsilon)))$ [122] in such cases, where the multiplicative constant depends on the details of the universal gate set used.

The result of Harrow and Low is especially significant here because the gates do not need to be known for it to hold. This means that even if we want to use randomised gap estimation to learn how to control an uncalibrated quantum system, we do not need to explicitly know the gates that are actually applied to the system to sufficiently randomise the measurement results if the underlying gate set is universal and a sufficiently long sequence of random operations is used.
Algorithm 2 Bayesian inference algorithm for eigenphases

Input: Set of unitary operators \{U : j = 1 : K\} such that \(U = I \otimes V_j\) where \(V\) acts on a \(M \leq N\) dimensional subspace of the computational basis, a diagonal Hamiltonian \(H_0\), annealing time \(T\), an interpolation function \(f : [0, 1] \rightarrow [0, 1]\) such that \(f(0) = 1 - f(1) = 0\), evolution time \(t\) and prior over \(M\) smallest gaps \(P(\Delta)\).

Prepare state \(|0\rangle \in \mathbb{C}^N\).

Randomly select \(U\) from set of unitary operations.

\[
|0\rangle \leftarrow U^\dagger \left( Te^{-i \int_0^1 (1 - f(s)) H_0 + f(s) H ds T} \right)^\dagger e^{-i H t} \left( Te^{-i \int_0^1 (1 - f(s)) H_0 + f(s) H ds T} \right) U |0\rangle.
\]

Measure state and \(E \leftarrow 0\) if the result is ’0’ otherwise \(E \leftarrow 1\).

Use approximate Bayesian inference to estimate \(P(\Delta|E) \propto P(\Delta) \cdot \int P(E|\Delta,t,U) \mu(U) dU\).

return \(P(\Delta|E)\).

6.4 Adiabatic elimination of eigenvalues

Most eigenvalue estimation tasks in quantum computing focus on learning only a part of the spectrum rather than all the eigenvalues. We could use randomised gap estimation to learn the eigenvalues in this part of the spectrum if we could apply Haar–random unitary operations in only that subspace. Since the eigenvectors of the Hamiltonian are generally not known, it is difficult to do this directly. Despite this fact, the adiabatic theorem provides a means by which these random unitaries can be applied to the appropriate subspace by employing adiabatic state preparation.

Let us assume we wish to learn the spectrum, up to an additive constant, for a given subspace of a Hamiltonian \(H_p\). In particular let \(S = \text{span}(|\lambda_{j_1}\rangle, \ldots, |\lambda_{j_m}\rangle)\) be a subspace of eigenvectors of \(H_p\) such that the eigenvalues obey \(\lambda_1 \leq \lambda_2 \leq \cdots\) and \(j\) is a monotonically increasing sequence on \(\{1, \ldots, 2^n\}\). Then we define an adiabatic interpolation to be a time–dependent Hamiltonian of the form \(H(s)\) such that \(H(0) = H_0\) and \(H(1) = H_p\) where \(H(s)\) is at least three-times differentiable for all \(s \in (0, 1)\) (where \(s\) is the dimensionless evolution time) and furthermore has a gapped spectrum for all times in this interval. A common example of such a Hamiltonian is

\[
H(s) = (1 - s)H_0 + sH_p. \tag{6.17}
\]

The adiabatic theorem [123, 124, 125] shows that if \(|\lambda_0\rangle\) is an eigenvector of \(H_0\) whose eigenvalue corresponds to that of \(|\lambda_j\rangle\) in the sorted list of eigenvalues,

\[
Te^{-i \int_0^1 H(s) ds T} |\lambda_0\rangle = e^{i \omega_j T} |\lambda_j\rangle + O(1/T), \tag{6.18}
\]
where $T$ is the time-ordering operator. Hence, if we can perform a Haar–random unitary on the subspace of eigenvectors of $H_0$, denoted by $S_0 = \text{span}(|\lambda^0_1\rangle, \ldots, |\lambda^0_m\rangle)$, we can adiabatically transform the resultant state to a Haar–random state in $S$, up to phases on each of the eigenvectors and error $O(1/T)$.

Now let $U$ be a Haar–random unitary acting on $S_0$. Then the new adiabatic protocol for estimating the eigenphase is

$$P(0|H; t, U) = \left| \langle 0 | U^\dagger \left( T e^{i \int_0^T H(s) ds} T \right) e^{-i H_p t} \left( T e^{-i \int_0^T H(s) ds} T \right) U | 0 \rangle \right|^2. \quad (6.19)$$

Given $U | 0 \rangle = \sum_{j:|\lambda_j\rangle \in S_0} \alpha_j |\lambda_j^0\rangle$ and $\tilde{U} | 0 \rangle := \sum_{j:|\lambda_j\rangle \in S} \alpha_j |\lambda_j\rangle$, (6.18) shows that

$$\left( T e^{-i \int_0^T H(s) ds} T \right) U | 0 \rangle = \sum_{j:|\lambda_j\rangle \in S_0} \alpha_j e^{i \omega_j t} |\lambda_j\rangle + O(1/T). \quad (6.20)$$

Then using the fact that $H_p |\lambda_j\rangle = \lambda_j |\lambda_j\rangle$, we see from (6.19) and (6.20) that

$$P(0|H; t, U) = \left( \sum_{k:|\lambda_k\rangle \in S} |\alpha_k|^2 \cos (\lambda_k t) \right)^2 + \left( \sum_{k:|\lambda_k\rangle \in S} |\alpha_k|^2 \sin (\lambda_k t) \right)^2 + O(1/T).$$

$$= \left| \langle 0 | \tilde{U}^\dagger e^{-i H_p t} \tilde{U} | 0 \rangle \right|^2 + O(1/T). \quad (6.21)$$

The adiabatic theorem can therefore be used to allow randomised gap estimation to be performed on a specified subspace of eigenvalues if there exists a gapped adiabatic path between a Hamiltonian that is diagonal in the computational basis and the problem Hamiltonian (see Algorithm 2 for an outline of this approach). This shows that the curse of dimensionality that afflicts this method can be excised in cases where such an adiabatic evolution is possible. Randomized gap estimation can thus, in principle, be used as a fundamental tool to characterise and calibrate untrusted quantum systems. We will examine this further in subsequent sections.

As a final note, the asymptotic scaling of the error that emerges because of diabatic leakage out of the eigenstates, or more generally eigenspaces, can be exponentially improved using boundary cancellation methods [126, 127, 128, 129], which set one or more derivatives of the Hamiltonian to zero at the beginning and end of the evolution. However, while this can substantially reduce the cost of the adiabatic transport, it requires fine control of the system Hamiltonian in order to realise the benefits of the cancellation [128]. As a result, it is not necessarily clear when the higher-order versions of these algorithms will find use in applications outside of fault tolerant quantum computing.
6.5 Application to amplitude estimation

Randomised gap estimation also provides an important simplification for amplitude estimation, which is a quantum algorithm in which the probability of an outcome occurring is estimated by combining ideas from amplitude amplification and phase estimation [95]. The algorithm quadratically reduces the number of queries needed to learn a given probability. The main significance of this is that it quadratically speeds up Monte–Carlo algorithms. Specifically, imagine that you are given an unknown quantum state of the form

\[ |\psi\rangle := A |0\rangle = a |\phi\rangle + \sqrt{1 - |a|^2} |\phi^\perp\rangle, \]

(6.22)

where \(|\phi\rangle \in \mathbb{C}^N\), \(A\) is a unitary operator and \(|a| \in (0,1)\). Furthermore, assume that you are given access to an oracle such that \(\chi |\phi\rangle = -|\phi\rangle\) and for all states \(|v\rangle\) orthogonal to \(|\phi\rangle\), \(\chi |v\rangle = |v\rangle\). Amplitude estimation then allows \(|a|^2\) to be estimated to within error \(\epsilon\) using \(\tilde{O}(1/\epsilon)\) applications of \(\chi\) and \(O(\log(1/\epsilon))\) measurements.

In order to understand how this works, consider the Grover search operator \(Q = -A\chi_0 A^\dagger\chi\), where \(\chi_0\) acts as \(\chi\) would for \(\phi = 0\). For the case of a single marked state we then have that

\[ Q^j |\psi\rangle = \sin((2j+1)\theta_a) |\phi\rangle + \cos((2j+1)\theta_a) |\phi^\perp\rangle, \]

(6.23)

where \(\theta_a := \sin^{-1}(a)\). It is then clear that \(Q\) enacts a rotation in this two-dimensional subspace and has eigenvectors

\[ |\psi_\pm\rangle = \frac{1}{\sqrt{2}} (|\phi\rangle \pm i |\phi^\perp\rangle), \]

(6.24)

where

\[ Q |\psi_\pm\rangle = e^{\pm i 2\theta_a} |\psi_\pm\rangle. \]

(6.25)

All other eigenvectors in the space orthogonal to this have eigenvalue \(\pm 1\).

We can learn these phases using randomised gap estimation. The most significant change is that here \(t\) must be taken to be an integer since fractional applications of the Grover oracle have not been assumed. In this case, the eigenvalues are in the set \(\{-2\theta_a, 0, 2\theta_a, \pi\}\), which implies that \(|\Delta_{i,j}|\) takes the values \(\{0, \pm 2\theta_a, 4\theta_a, \pi \pm 2\theta_a, \pi\}\). This means that the gap estimation process has very different likelihoods if \(t\) is even or odd. Since \(\cos((\pi \pm 2\theta_a)(2p+1)) = -\cos(2\theta_a(2p+1))\), for integer \(p\), many of the terms in (6.9) cancel if \(t\) is odd. As a result, it is better to choose \(t\) to be an even integer for this application, in which case the likelihood function is

\[ P(0|\theta_a; t) = \frac{2}{N+1} \left( 1 + \frac{1}{N} \left( \frac{N-2}{2} \right) + 2(N-2) \cos(2\theta_a t) + \cos(4\theta_a t) \right). \]

(6.26)
6.6. APPLICATION TO CONTROL MAP LEARNING

Another important application of randomised gap estimation is calibration of quantum devices. In order to understand why calibration can be a problem, imagine that you wish to calibrate a qubit so that you can perform an X gate within error $\epsilon = 10^{-5}$. In order to verify that the error is this small, roughly $10^{10}$ experiments are needed. This process would also have to be repeated for every quantum bit in the system, which could easily result in terabytes of data needed to calibrate a small fault tolerant quantum device. Our approach provides a method that can exponentially reduce the number

\[
\begin{align*}
|\psi\rangle & \rightarrow e^{-iH(G(c))t/r} |\psi\rangle \\
\rightarrow e^{iH(G_{\text{trust}}(c_{\text{est}}))t/r} \rightarrow e^{iH(G_{\text{trust}}(c_{\text{est}}))t/r} & \rightarrow D
\end{align*}
\]

Figure 6.2: Quantum bootstrapping experiment to learn a control map using interactions between a trusted simulator and an uncharacterised device. The value $r$ can be increased to combat the spread of non-local correlations and also to reduce ambiguities that can arise in the inference process when the untrusted and trusted Hamiltonians do not commute.

Such experiments cannot yield substantial information as $N \rightarrow \infty$ according to (6.11) and thus different ideas are needed to make large $N$ instances tractable unless we can sample according to the Haar measure only within the relevant subspace using techniques similar to those in Section 6.4.

This method is, however, viable without modification for small $N$. To illustrate this, consider the case where $N = 2$, where the likelihood function reduces to

\[
P(0|\theta_a; t) = \frac{2}{3} \left( \frac{1}{2} + \cos^2(2\theta_a t) \right).
\]

This likelihood function yields only slightly less information than the one for iterative phase estimation, $P(0|\theta_a; t) = \cos^2(2\theta_a t)$, as their derivatives with respect to $\theta_a$ differ by at most a constant factor. This observation, coupled with the fact that an additional qubit is not needed, means that amplitude estimation can be realistically applied in-place in single qubit devices using this method.

6.6 Application to control map learning

Another important application of randomised gap estimation is calibration of quantum devices. In order to understand why calibration can be a problem, imagine that you wish to calibrate a qubit so that you can perform an X gate within error $\epsilon = 10^{-5}$. In order to verify that the error is this small, roughly $10^{10}$ experiments are needed. This process would also have to be repeated for every quantum bit in the system, which could easily result in terabytes of data needed to calibrate a small fault tolerant quantum device. Our approach provides a method that can exponentially reduce the number
of measurements needed to control such a qubit under these circumstances and polynomially reduce the experimental time needed.

In order to abstract the problem of calibrating an unknown quantum device, let us consider the problem of learning a control map. We define the control map to be the mapping between a set of experimental controls (specified by the vector $c$) and the system Hamiltonian. Specifically, if we define $H(x) = \sum_{i=1}^{N} x_i H_i$ for a set of Hamiltonians $H_i$, then $x$ is given by $c$ via

$$x = Gc + x_0,$$  \hspace{1cm} (6.28)

where the matrix $G$ is known as the control map. For simplicity we will only consider the linear case where $x_0 = 0$ below. The general case is handled by first learning $x_0$ and is discussed in [109].

Recently, a method called quantum bootstrapping has been proposed to learn such maps efficiently, wherein a trusted quantum simulator is used to infer the controls of an untrusted device (see Figure 6.2 for a circuit diagram of a bootstrapping experiment). Although this can allow the control map to be learned using a poly-logarithmic number of experiments, it necessitates the use of swap gates that couple the device to the trusted simulator that is used to characterise it. This means that verifying that the trusted simulator is properly working can still require an exponential number of measurements (even under assumptions of locality). This is significant because the error in the trusted simulator places a lower limit on the error that can be attained using the experiments proposed in [109, 115, 116, 117]. Figure 6.3 demonstrates the dependence of the error in the inference of a control map for a single qubit that arises from the use of a miscalibrated trusted simulator that has $G$ drawn from a zero–mean Gaussian with covariance matrix $\delta I$, where $\delta$ is chosen to take values within the interval $[0, 10^{-2}]$. We see there that the error decays until it saturates at a level dictated by the miscalibrations in the trusted Hamiltonian. This illustrates that errors in the trusted simulator can prevent a bootstrapping protocol from perfectly controlling this idealised quantum system.

It is worth noting that although Bayesian inference does not correctly infer the true control map for the system because of these errors in the trusted simulator, it does infer a set of controls that precisely mimics the trusted device. In this sense, the learning task can be thought of as a machine learning task, wherein the untrusted device is trained to maximise the fidelity of its output with that of the “trusted” simulator, wherein the output state can be thought of as the training data for the problem. It is therefore crucial to have a well calibrated trusted device if we wish to leverage the full power of quantum bootstrapping.
Figure 6.3: Median error for a quantum bootstrapping protocol that attempts to learn a control map for a universal qubit using a miscalibrated second qubit to control the measurement basis with miscalibrations of size $\delta$. These results are simulated in the limit as $r \to \infty$ (see Figure 6.2) as rejection sampling filtering has poor success probability for non–commuting models because of (intermediate multi-modality), unlike the sequential Monte-Carlo methods with $r = 1$ studied in [116].

To this end we provide a number of examples below that show how the control maps of single qubit Hamiltonians can be learned using randomised gap estimation. We thereby show that our techniques may be of great value in building a trusted simulator that can be used in bootstrapping protocols.

### 6.6.1 Two level example

In the case for a two–level, single–qubit, system, the Hamiltonian can be written, up to an irrelevant shift in energy, as

$$H([\alpha, \beta, \gamma]) = \alpha X + \beta Y + \gamma Z. \quad (6.29)$$

Since these three operations anti–commute it is evident that the eigenvalues of $H$ are

$$E = \pm \sqrt{\alpha^2 + \beta^2 + \gamma^2}. \quad (6.30)$$
Thus we can infer information about $\alpha$, $\beta$ and $\gamma$, and in turn $G$, from the eigenspectra of different experiments. Specifically,

$$
\begin{bmatrix}
\alpha
\beta
\gamma
\end{bmatrix} =
\begin{bmatrix}
G_{00} & G_{01} & G_{02} \\
G_{10} & G_{11} & G_{12} \\
G_{20} & G_{21} & G_{22}
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
c_3
\end{bmatrix}.
$$

(6.31)

The simplest example of control map learning is the diagonal case, for which

$$
E = \pm \sqrt{G_{00}^2 c_1^2 + G_{11}^2 c_2^2 + G_{22}^2 c_3^2},
$$

(6.32)

and the control map can be learned, up to signs of $G_{ij}$, using a sequence of three randomised phase estimation experiments with $c = [1, 0, 0]$, $[0, 1, 0]$ and $[0, 0, 1]$. The signs can be inexpensively learned using auxiliary experiments because only a single bit of information is required. Thus empirically the number of experiments required to learn $G$ using randomised phase estimation is upper bounded by a constant times $\log(1/\epsilon)$ (see Figure 6.1).

The next simplest case to consider is that of an upper triangular control map in the form

$$
\begin{bmatrix}
\alpha
\beta
\gamma
\end{bmatrix} =
\begin{bmatrix}
G_{00} & G_{01} & G_{02} \\
0 & G_{11} & G_{12} \\
0 & 0 & G_{22}
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
c_3
\end{bmatrix}.
$$

(6.33)

Clearly $G_{00}$ can be learned directly with randomised gap estimation using the prior experiments. However, the remaining elements must be inferred from different different experiments. If $c = [1, 1, 0]$ then

$$
E([1, 1, 0]) = \pm \sqrt{(G_{00} + G_{01})^2 + G_{11}^2}
$$

(6.34)

$$
E([0, 1, 0]) = \pm \sqrt{G_{01}^2 + G_{11}^2}.
$$

(6.35)

After squaring and subtracting both equations we find

$$
E^2([1, 1, 0]) - E^2([0, 1, 0]) = G_{00}^2 + 2G_{00}G_{01},
$$

(6.36)

which can be solved uniquely for $G_{01}$ since $G_{00}$ is known. Once $G_{01}$ is known, $G_{11}$ can be learned unambiguously from (6.35) using the fact that $G_{11} \geq 0$.

After these steps we have inferred the first two columns of $G$. The remaining column can be learned similarly by performing three randomised gap estimation experiments with $c = [0, 0, 1], [1, 0, 1]$ and $[0, 1, 1]$ which yield

$$
E([0, 1, 1]) = \pm \sqrt{(G_{01} + G_{02})^2 + (G_{11} + G_{12})^2 + G_{22}^2}
$$

(6.37)

$$
E([1, 0, 1]) = \pm \sqrt{(G_{00} + G_{02})^2 + G_{12}^2 + G_{22}^2}
$$

(6.38)

$$
E([0, 0, 1]) = \pm \sqrt{G_{02}^2 + G_{12}^2 + G_{22}^2}
$$

(6.39)
Then by subtracting the square of (6.39) from (6.38) we learn $G_{02}$, from which we learn $G_{12}$ from substituting the result into (6.37). $G_{22}$ can then be learned by subtracting the square of (6.38) from (6.37), substituting the value of $G_{02}$ and using $G_{22} \geq 0$.

More generally, this approach provides information about the inner product between any two columns of $G$. No more information about $G$ can be extracted with such measurements. That is, $G^T G$ can be learned, but $G$ itself is only determined up to an orthogonal transformation $G' = QG$ which preserves the inner products of its columns. The matrix $Q$ can further be determined only if additional constraints are imposed on $G$. For example, if $G$ is upper or lower triangular and with a positive diagonal it is unique within such measurements and $Q$ is the identity matrix as discussed above.

### 6.6.2 Learning $2 \times 2$ single qubit control maps

As mentioned above, gap estimation experiments alone are in general not enough to uniquely specify the control map. In fact, amplitudes of the states in the eigenbasis are required. This raises the question of whether amplitude estimation may be used to glean the necessary information from these coefficients using $O(\log(1/\epsilon))$ measurements.

For simplicity let’s consider the Hamiltonian of a single qubit with only $X$ and $Z$ rotations

$$H([\alpha, \gamma]) = \alpha X + \gamma Z. \quad (6.40)$$

and

$$\begin{bmatrix} \alpha \\ \gamma \end{bmatrix} = \begin{bmatrix} G_{00} & G_{01} \\ G_{10} & G_{11} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}. \quad (6.41)$$

The case of a full $3 \times 3$ control map is similar, but we do not discuss it here because it is much more cumbersome owing to the increased number of equations and unknowns.

Randomized gap estimation can provide

$$E([1, 0]) = \pm \sqrt{G_{00}^2 + G_{10}^2}$$

$$E([0, 1]) = \pm \sqrt{G_{01}^2 + G_{11}^2}$$

$$E([1, 1]) = \pm \sqrt{(G_{00} + G_{01})^2 + (G_{10} + G_{11})^2}, \quad (6.42)$$

Since there are four unknowns and three equations, the control map cannot be unambiguously learned from these relations. However, it can be seen that any other such randomised gap estimation provides information that is not
linearly independent of these three equation. We need additional information that cannot directly come from randomised gap estimation to solve the problem. This problem was avoided in the prior example by constraining $G_{10}$ to be zero.

We can learn the required final piece of information by measuring expectation values of the following three operators

\[ A([1, 0]) = e^{-i(G_{00}X+G_{10}Z)\pi/(2E([1,0]))}, \]
\[ A([0, 1]) = e^{-i(G_{01}X+G_{11}Z)\pi/(2E([0,1]))}, \]
\[ A([1, 1]) = e^{-i((G_{00}+G_{01})X+(G_{10}+G_{11})Z)\pi/(2E([1,1]))}. \] (6.43)

These each correspond to free evolution of the system Hamiltonian under particular control settings for fixed durations. We can then use these operators (see eq (6.22)) to learn the following amplitudes while marking the $|0\rangle$ state

\[ |a([1, 0])| = |\langle 0 | A([1, 0]) | 0 \rangle| \right| G_{10} \right| E([1,0]) \right|, \] (6.44)
\[ |a([0, 1])| = |\langle 0 | A([0, 1]) | 0 \rangle| \right| G_{11} \right| E([0,1]) \right|, \] (6.45)
\[ |a([1, 1])| = |\langle 0 | A([1, 1]) | 0 \rangle| \right| G_{10} + G_{11} \right| E([1,1]) \right|. \] (6.46)

If the sign of $G_{11}$ or $G_{10}$ is known, these three quantities are enough to unambiguously solve for $G_{11}$ and $G_{10}$, and can all be learned in-place using randomised amplitude estimation if an accurate $Z$–gate can be performed. The remaining elements of the control map can then be found from (6.42). If the sign of $G_{11}$ or $G_{10}$ is not known, they can still be learned with amplitude estimation if the states $(|0\rangle \pm i|1\rangle)/\sqrt{2}$ can be prepared on the quantum device.

In the absence of a calibrated $Z$ gate, the above strategy of amplitude estimation does not apply. The requisite quantities $|a([0, 1])|$, $|a([1, 0])|$ and $|a([1, 1])|$ can nonetheless be learned by statistical sampling. Doing so requires $O(1/\epsilon^2)$ experiments, which dominates the cost of data acquisition. However, randomised gap estimation efficiently provides $E([0,1]), E([1,0])$ and $E([1,1])$ which allows us to optimally extract these expectation values (for the form of the experiments considered). To see this consider the case of learning $|a([0, 1])|$ for arbitrary $t$. The likelihood function is

\[ P(0|G; t) = \cos^2(E([0, 1])|t| + \sin^2(E([0, 1])|t||a([0, 1])|^2. \] (6.47)
It is then clear that the Fisher information is maximised at $t = \pi/(2E([0, 1]))$. In order to compare, let us consider the case where $E([0, 1])t \mod 2\pi$ is taken from a uniform prior over $0, 2\pi$. Then the marginalised likelihood is

$$\frac{1}{2\pi} \int_0^{2\pi} dx \cos^2(x) + \sin^2(x)|a([0, 1])|^2 = \frac{|a([1, 0])|^2 + 1}{2}. \quad (6.48)$$

thus (6.11) shows that the Fisher information for such experiments is reduced by a factor of 2. This furthermore reduces the inference problem to frequency estimation wherein the non–adaptive strategy of estimating $|a([0, 1])|^2$ using the sample frequency is optimal. Thus, even though randomised gap estimation does not lead to a quadratic reduction in the experimental time needed to learn $G$ here, it dramatically simplifies the inference problem and removes the need to use multiple experimental settings to learn the amplitudes.

Above we only discussed the case of a single qubit Hamiltonian with uncalibrated $X$ and $Z$ interactions. The case of a general (traceless) single qubit Hamiltonian follows from exactly the same reasoning. First, one applies randomised gap estimation to learn the relevant energies corresponding to the 6 linearly independent sets of controls. The remaining 3 equations are found by measuring appropriate expectation values using amplitude estimation if a $Z$ gate is available, and statistical sampling if it is not.

Finally, higher dimensional control maps can also be learned using this approach. The analytic approach given above no longer holds unless the Hamiltonian can be decomposed into a set of anti–commuting Pauli operators. The same process can, however, be applied numerically in such cases. We leave a detailed discussion of the issues that arise when doing so for future work.

### 6.7 Conclusion

We introduced an alternative form of phase estimation that uses Bayesian inference on top of randomised experiments to learn the spectral gaps of a unitary operation without needing external qubits or well calibrated gates. In addition to calibration, this gap estimation procedure allows amplitude estimation to be performed in place for quantum systems without needing auxiliary qubits to control the evolution. These randomised gap and amplitude estimation algorithms could radically reduce the costs of characterising and calibrating small quantum devices. Our work further suggests that they may find use in building the initial trusted simulator that is required to start a quantum bootstrapping protocol [109] as well as allowing amplitude estimation to be done in place in small quantum systems.
An important caveat of our work is that the randomisation procedure that we use to extract information about the gaps leads to an exponential decrease in the signal yielded by experiments on high-dimensional systems. We show that this can be mitigated, for some systems, using adiabatic paths to allow the randomisation to be performed only within a low-dimensional eigenspace of the system. Regardless, this caveat still holds in general and as such our methods are complimentary to existing phase and amplitude estimation methods and do not replace them.

This suggests an important remaining question namely that of whether efficient methods exist for estimating phase and amplitude that do not require entangling the system with ancillary qubits. The development of such methods would constitute a major step forward for quantum metrology. Similarly, proof that such methods are impossible would give us new insights into the efficacy of phase estimation and reveal fundamental trade offs that exist between quantum resources for metrological tasks.
Part III

Optimisation
Chapter 7

Introduction

This part is based on the papers [231, 232, 236] and patents [240, 241].

Multivariate optimisation is of central importance in industry and is used for a range of problems from finding solutions which satisfy a set of constraints to training classifiers in machine learning. Despite the wide range of applications, nearly all of them involve finding an optimal assignment of a set of variables with respect to a cost function. An exact global optimum is, however, rarely required and in general significantly more costly to find than a very good local optimum. Heuristic optimisers are therefore ubiquitous for many industrial application problems [130, 131, 132].

A heuristic optimiser often starts from a random configuration of variables and optimises this configuration until a local optimum is found. For a convex optimisation problem this optimum is also the global optimum and no further effort is required. However, problems which are considered NP-hard have, in general, exponentially many such local optima and hence the probability that a particular optimum is also the global one is exponentially suppressed. Nevertheless, industrial application problems often contain some amount of structure which makes their solution useful. This structure can be used to obtain configurations of a given cost much faster than random guessing. As such, each subsequent query to the cost function makes use of preceding queries to make a more informed guess of an optimal configuration.

Here we study ways to improve upon the current state of the art in heuristic optimisation, effectively making better use of each query to the cost function, for two applications. The first is the tasks of finding ground states of Ising spin glasses. The second is building optimal models in machine learning.

In statistical physics the most basic mathematical model for ferromagnetism is the Ising model [133]. If the couplings are sampled from a random
distribution such that the mixture of ferromagnetic and anti-ferromagnetic couplings is roughly equal \cite{134, 135}, the Ising model becomes a spin glass. A spin glass has a frustrated energy landscape, where it is impossible to minimise the energy of all couplings at the same time, and gives rise to multiple metastable configurations with larger energies than the global minimum.

A spin glass typically has a critical temperature below which the correlation length diverges \cite{136}. This results in the spins freezing into one of the metastable configurations. As the directions of the spins are randomly oriented, the net magnetisation is zero, an it cannot be distinguished from a paramagnet by only measuring the magnetisation. This affinity to freeze into one of the metastable configurations makes it NP-hard to find the ground state configuration of a spin glass \cite{137} and has resulted in immense interest in these systems from the mathematics and computer science community. In this thesis we will focus on the application of Ising spin glasses to solving combinatorial optimisation problems.

First, we present several efficient implementations of the simulated annealing algorithm for Ising spin glasses on sparse graphs, including specialised optimisations for bipartite graphs and highly optimised implementations using multi-spin coding for graphs with small maximum degree and discrete couplings with a finite range. The latter codes achieve up to 50 spin flips per nanosecond on modern Intel CPUs. We also compare the performance of the codes to that of the special purpose Dwave device \cite{138} built for solving such Ising spin glass problems.

Secondly, we address the issue of using special purpose devices to solve large application problems. Specialised hardware generally trades off generality against performance on specific tasks it was programmed to do. An important such limitation is the number of variables it can handle, which is usually very limited and makes it non-trivial to apply such a device to solve large application problems.

Here we introduce a heuristic, general-purpose hierarchical approach to use a small device to solve large problems by mapping sub-sets of variables onto the device, optimising the local cost function and mapping the solution back to the global problem. This method can also be used to boost the performance of an optimiser by using it to solve sub-sets of variables rather than the whole problem at once.

The second application of optimisers we will study is machine learning. Machine learning, together with artificial intelligence, is currently one of the hottest fields in computer science. The rapid increase in the amount of available data and compute resources has given a significant boost to the power of classifiers, planning and learning algorithms.

Only in the last year, a program was trained to spot faces from a wide
range of angles and when partially occluded [139], which was considered an ability unique to humans. For the first time in history another program beat a professional human player in a formal match of Go [140], a task which was conjectured by AI experts to be at least a decade away. Another program learned to play Atari2600 games significantly better than the world's top players just by looking at the screen through a camera and having access to the score and the joystick [141]. Yet another program learned to recognise handwritten characters as fast as a human, from only a single example [142]. These are just a small part of the recent progress in the field.

At the heart of all those achievements are statistical models which are trained to mimic the environment as accurately as possible. The training itself is an optimisation problem, where the cost function is the accuracy on test data and the variables are the parameters defining the model. To be able to accurately represent input data, the model needs to contain non-linear terms with arbitrary complex interactions, which often leads to an exponential growth of the training time with desired accuracy. Hence, similar to heuristic optimisation algorithms, training algorithms often get stuck in local optima where the algorithm is unable to find further improvement. The standard approach to circumvent this problem involves restarting from random initial configurations.

We propose a method of partial reinitialisation, whereby, subsets of variables, rather than the whole configuration, are reinitialised. Much of the information gained from previous runs is hence retained. This leads to significant improvements in the quality of the model found in a given time for a variety of machine learning applications.
Chapter 8

Optimised simulated annealing for Ising spin glasses

First introduced three decades ago [143], simulated annealing is a powerful algorithm commonly used for heuristic optimisation due to its simplicity and effectiveness. Within this approach, variables to be optimised are viewed as the degrees of freedom of a physical system and the cost function of the optimisation problem as the energy. One then performs a Monte Carlo simulation of that system, starting at high temperatures and slowly lowering the temperature during the simulation, so that ultimately the configuration of the system ends up in a local minimum. Annealing slow enough and with multiple repetitions, one can hope to find the global minimum.

In this paper we present highly optimised implementations of simulated annealing for the Ising spin glass problem

\[ H = \sum_{i<j} J_{ij} s_i s_j + \sum_i h_i s_i \]  (8.1)

where \( s_i = \pm 1 \). The couplings \( J_{ij} \) induce a graph structure with the spins represented as vertices and with edges between all neighbour pairs \( i \) and \( j \) for which \( J_{ij} \neq 0 \).

The broad interest in the Ising spin glass comes from the fact that finding the ground state is non-deterministic polynomial (NP) hard [137]. This means that many other interesting problems, including constraint satisfaction and the travelling salesman problem, can be mapped to such Ising spin glasses in polynomial time.

The complexity of solving this problem is also the motivation behind special purpose devices built by the company D-Wave systems, which can to date solve Ising spin glass problems with up to \( N = 512 \) spins on the so-called “chimera graph”. Chimera graphs are lattices with bipartite fully
connected unit cells consisting of $2c$ vertices (denoted by $K_{c,c}$) distributed on a $L \times L$ grid (see figure 8.1). D-Wave Two implements a transverse field Ising model on an $8 \times 8 K_{4,4}$ graph, where every node represents a spin and every edge represents a coupling.

The existence of the D-Wave devices has triggered increased efforts into effectively mapping non-linear combinatorial optimisation problems from application domains, such as image recognition, to Ising spin glass problems [144, 145]. A recent study [146] has compared the performance of a D-Wave device against three general purpose classical optimisation algorithms and concluded that the D-Wave device tested was 3600 times faster than the fastest of these codes. In their conclusions, the authors qualify their result with the statement “It would of course be interesting to see if highly tuned implementations of simulated annealing could compete”. The optimised simulated annealing codes for Ising spin glasses presented in this publication can also be applied to the chimera graph of the D-Wave devices and provide such competitive highly tuned implementations. In fact, all the benchmarks in this work are performed on chimera graphs.

Another state-of-the-art approach for finding ground states of spin glasses is parallel tempering [147, 148]. This method can be more efficient than simulated annealing in some cases [149, 150], e.g. for Ising spin glasses with Gaussian distribution of couplings. Most of the optimisation techniques presented in this paper can also be applied to parallel tempering.
8.1 Optimisations

Simulated annealing is simple and can be implemented in a short time for the Ising spin glass. However, a range of optimisations can improve its performance by orders of magnitude. In this work we discuss many of these optimisations and present efficient implementations for modern CPUs in a freely available software package.

8.1.1 Forward computation of $\Delta E$

Performing simulated annealing using the Metropolis algorithm requires calculating the acceptance ratio $\exp(-\beta \Delta E_i)$, where $\beta$ is the inverse temperature and $\Delta E_i$ the energy change upon flipping the $i$'th spin. The value of $\Delta E_i$ is typically computed by traversing the neighbours of spin $i$ and takes up most of the time required for each spin update. However, as for typical annealing schedules the average acceptance rate is only around 15%, it is much more efficient to calculate and store $\Delta E_i$ for every spin and only update this value if spin $i$, or one of its neighbours, is flipped. This way the number of operations if a spin-flip is accepted is the same with an additional array access. On the other hand, if a flip is not accepted, $\Delta E_i$ does not have to be computed, but simply retrieved from an array.

8.1.2 Fixed loop lengths and unrolling

One has to loop over neighbours to compute the energy change $\Delta E_i$ when flipping the $i$-th spin. This loop can be optimised using fixed loop lengths by specifying the maximum number of neighbours at compile time. In this case, the compiler can unroll the loop more efficiently. This approach is advantageous when the distribution of the number of neighbours is narrow. For instance, for perfect chimera graphs with five and six neighbours (there might be a few sites with four neighbours in depleted graphs) the code with fixed loop length is 20% faster. However, using the fixed loop length codes might be disadvantageous when the distribution of the number of neighbours is wide, say, for graphs with the majority of sites having three neighbours and a few sites having ten neighbours.

8.1.3 Fast random number generators

For a simple model, like the Ising model, generation of random numbers can take up a substantial fraction of the computational effort. Unlike simulations aiming at high accuracy results for physical properties, in optimisation
algorithms such as simulated annealing the quality of the random number generator is not very critical and thus fast generators, such as the linear-congruential, can be used.

8.1.4 Deterministic traversal over lattice sites

Lattice sites can be picked up for an update sequentially in some specified order instead of picking them up in random order. This approach decreases the amount of generated random numbers and leads to faster codes. Even though the detailed balance condition is violated, it typically yields better success rates.

8.1.5 Precomputing random numbers

Random numbers can also be reused across multiple repetitions of the annealing, as long as they start from different initial configurations. Furthermore, we can modify the Metropolis acceptance criterion from \( \exp(-\beta \Delta E_i) < u \), where \( u \in [0, 1) \) is drawn uniformly at random, to a cheaper decision \( \Delta E < r \), where \( r = -\frac{1}{\beta} \log u \). The values \( r \) can then be stored instead of \( u \). For the case of integer couplings we can further optimise by using integer comparison instead of floating point comparisons.

Correlations introduced by reusing random numbers can be significantly reduced with minimal additional effort by cyclically shifting the precomputed array of random numbers for each sweep (a sweep is defined as one attempted update per spin) by a random offset. We observed that the remaining correlations have only a minimal impact on the performance of the simulated annealer.

8.1.6 Optimisations for bipartite graphs

If the graph is bipartite, the complexity of finding the ground state configuration can be reduced to finding the optimal spin-vector for only one sub-lattice. Let’s split the set of spins into two sets \( A \) and \( B \) such that the spins from one set only couple to spins of the other. Without loss of generality we assume that all on-site fields \( h_i = 0 \), \( N_A \leq N_B \) and let’s sort the spins such that all spins in \( A \) come before those in \( B \). We use the notation \( s_A = \{s_1, s_2, \ldots, s_{N_A}\}^T \), \( s_B = \{s_{N_A+1}, s_{N_A+2}, \ldots, s_N\}^T \) and \( s = \{s_A, s_B\} \).

The couplings \( J_{ij} \) are then in matrix form

\[
J = \begin{pmatrix} 0 & C^T \\ C & 0 \end{pmatrix}
\] (8.2)
where the energy can be calculated as

$$E = \frac{1}{2} s^T J s = \frac{1}{2} \left( s_A^T C s_B + s_B^T C s_A \right) = s_B^T C s_A. \quad (8.3)$$

which can be minimised by finding the optimum $s_A$ such that

$$E = \min_{s_A} \left\{ -\sum_{i=0}^{N_B} \sum_{j=0}^{N_A} C_{ij} s_j \right\} \quad (8.4)$$

as we can always align the spins in $s_B$ accordingly. Therefore, for bipartite graphs we need to simulate and update only $N_A \leq N/2$ spins. In the above argument we for simplicity made the assumption that all on-site fields are zero. If this is not the case, we can transform the Hamiltonian by introducing two ancillary spins, one coupled to all spins in sub-lattice $A$ with couplings $h_i$ for $i \in [1, N_A]$ and the other to all spins in sub-lattice $B$ with couplings $h_i$ for $i \in [N_A + 1, N]$. The two spins are also coupled to each other with a strong ferromagnetic bond. After this transformation, the graph remains bipartite, but all on-site fields are expressed as couplings between spins and the above argument can be applied.

### 8.1.7 Multi-spin coding

In contrast to the standard implementations of simulated annealing where one uses an integer to store every spin, higher efficiency can be archived by representing spins by single bits which allows one to update many spins simultaneously. This approach is known as **multi-spin coding**. We here present two different implementations of multi-spin coded simulated annealers. The codes were written for different ranges of couplings and with up to six nearest neighbours, using words of $S = 64$ bits to stores 64 spins. Rather than storing spins from a single lattice across a word, as described in \[151\], we store 64 replicas of the same spin in one word.

**Approach one**

Multi-spin coded versions of simulated annealing were first suggested in Ref. \[152\] and later extended in Refs. \[153, 154, 155\]. The implementation presented here is based on the outline found in Ref. \[155\]. For completeness, we summarise this approach. In a standard Monte Carlo simulation of the Ising model, one selects a random spin and computes the energy $\Delta E$ gained by flipping this spin. The move is either accepted or rejected with a probability given by the Boltzmann factor $e^{-\beta \Delta E}$, where $\beta$ is the inverse temperature.
In the approach suggested in Ref. [155], this part of the algorithm is slightly altered. For a finite number of lattice neighbours and integer couplings, the number of all possible values of $\Delta E$ is finite and these values can be ordered in descending order $\Delta E \in \{-2E_0, \ldots, -2E_m\}$, where $E_i$ are the local energy levels of the spin and $m + 1$ is the number of energy levels. Instead of calculating $\Delta E$, one can efficiently calculate the index $i$ of $-2E_i$ for all the 64 spins simultaneously by bitwise summation, i.e. indices are stored in $\lceil \log_2(k) \rceil$ words, where $\lceil \rceil$ denotes the next largest integer value. The spins which need to be flipped with probabilities $p_{t,i} = e^{2\beta E_i}$ can be determined by performing simple boolean logic on words that represent indices $i$ and by comparing the probabilities $p_{t,i}$ with a uniformly distributed random number $0 \leq u < 1$ starting at the highest through the lowest level. This can be illustrated by the following pseudo code example for a spin that couples to three neighbours with $J_{ij} = \pm 1$.

```plaintext
10 = jzw0 ^ (spin ^ neighbour0.spin)
11 = jzw1 ^ (spin ^ neighbour1.spin)
12 = jzw2 ^ (spin ^ neighbour2.spin)
i1 = 10 ^ 11
i0 = i1 ^ 12
i1 = (10 & 11) ^ (i1 & 12)
double u = rand(1)
if (u < p0) {
    spin = spin ^ (-1)
} else if (u < p1) {
    spin = spin ^ (i1 | i0)
} else {
    spin = spin ^ i1
}
```

Here spin, jzw0, jzw1, jzw2, 10, 11, 12, i0, and i1 are 64-bit words, spin stores 64 spins for 64 replicas, jzw represents the coupling constant $J$ (all bits of jzw are set to zero if $J = 1$ and all bits of jzw are set to one if $J = -1$). In the first three lines of the code, we determine whether the interaction energy is positive or negative for every pair of interacting spins. A bit of lj is set to one if the corresponding interaction energy is positive and set to zero if the interaction energy is negative. In the next three lines of the code, we calculate the index $i$ by bitwise summation of 10, 11, and 12. In this simple example, we need only two words to store the index. All possible indices $i$ for one replica, corresponding energies $E_i$, energy gains $\Delta E_i$, and probabilities $p_{t,i}$ are listed in table 8.1. In the seventh line, we draw a uniformly
8.1. Optimisations

<table>
<thead>
<tr>
<th>Index $i$</th>
<th>Energy $E_i$</th>
<th>Energy gain $\Delta E_i$</th>
<th>Probability $p_{t,i}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$-3</td>
<td>J</td>
<td>$</td>
</tr>
<tr>
<td>1</td>
<td>$-1</td>
<td>J</td>
<td>$</td>
</tr>
<tr>
<td>2</td>
<td>$1</td>
<td>J</td>
<td>$</td>
</tr>
<tr>
<td>3</td>
<td>$3</td>
<td>J</td>
<td>$</td>
</tr>
</tbody>
</table>

Table 8.1: Indices, energies, energy gains when a spin is flipped, and probabilities for one spin that couples to three neighbours with $|J| = 1$.

distributed random number. In the next lines, we compare it to the probabilities $p_{t,i}$ and flip spins. It is easy to deduce from the correspondence between indices and probabilities which spins should be flipped. Namely, we find by simple boolean logic the spins with indices $i \geq k$ such that $p_{k-1,t} < u < p_{k,t}$ and flip these spins. As the probabilities $p_{t,i}$ are used often throughout the simulations, these are precomputed when the algorithm is initialised.

Correlations

This approach results in correlations between the replicas because only one random number is used per update for all the 64 replicas in a word. For example, if at any point during the annealing two replicas are in the same state, they will follow the same path, making one of the replicas redundant. In the extreme case of fully correlated replicas, all of them find the same state. Correlations can be measured by computing the correlation ratio

$$C = \frac{V}{W(1 - W)}$$

over multiple repetitions of the annealing process, where $W = (1/R) \sum_{i=1}^{R} w_i$ is the mean success rate (the probability of finding the ground state), $R$ is the number of repetitions, $V = (1/R) \sum_{i=1}^{R} (w_i - W)^2$ is the variance and $w_i = (1/64) \sum_{j=1}^{64} w_{ij}$ is the mean success rate of the $i$th repetition (each repetition has $S = 64$ replicas). It can be shown that $C$ is close to zero for uncorrelated replicas and $C = 1$ for fully correlated replicas.

In figure 8.2, we show the correlation ratio $C$ as a function of the number of sweeps for instances on the chimera graph. Correlations increase with the ratio of the number of sweeps to the system size. However, they can be substantially reduced by not flipping one random spin (bit) in each update. The random number that is used to make an update can be reused. As can be seen, our decorrelation strategy reduces the correlations significantly but not fully. It should be emphasised that these correlations are usually irrelevant because strong correlations appear only when the number of sweeps is much
larger than the optimal number of sweeps to find the ground state with very high probability, as discussed below.

**Approach Two**

The second approach follows the ordinary algorithm where a spin is picked, its local energy is computed and it is flipped with probability \( p \). However, instead of just flipping one spin, one determines the individual energies of 64 spins simultaneously and computes whether the spins should be flipped from a set of probabilities. The average case complexity of generating \( Q \) 1-bits with probability \( p \) in parallel is \( \mathcal{O}(\log_2(Q) + 2) \) [156]. This way 64 bits are generated in on average 8 iterations. Since we are considering sparse graphs with a limited range, only a limited number of flipping probabilities can be attained by a spin at each time step and this makes the generalisation of the above algorithm to individual flipping probabilities \( p^{(1)}, p^{(2)}, \ldots, p^{(Q)} \) straight forward. While this algorithm is more than a factor of three slower than the one presented in Sec. 8.1.7, correlations are here of the order of the pseudo random number generator.
8.2 Optimising annealing strategies

It is important to optimise both the slope and the length of the annealing schedule.

8.2.1 Optimising the schedules

We follow the ideas in Ref. [157]. Based on considerations of keeping the average energy difference between two successive steps $k$ and $k+1$ below a threshold $\langle E_{k+1} \rangle - \langle E_k \rangle \leq -\lambda \sigma_k$ it can be shown that $\beta_{k+1} = \beta_k + \lambda \sigma_k^{-1}$ where $\sigma_k$ is the standard deviation of the energy at step $k$ and $\lambda \leq 1$. We optimised the schedule using this approach, but found that the performance was only slightly better than a linear schedule. Much more important is that the starting value of the temperature $T$ is around the same order as the maximal energy required to flip any spin and that the end value is low enough such that the state does not jump out of the final minimum. For bi-modal couplings $J_{ij} = \pm 1$, we found that the inverse temperatures $\beta_s = 0.1$ and $\beta_e = 3$ were good initial and final values up to 512 spin problems, and these values were used for the benchmark runs.

8.2.2 Optimising annealing times

As simulated annealing is a heuristic algorithm, one can strive towards maximising the probability of finding the ground state by either increasing the number of sweeps, increasing the number of repetitions, or both at once. The optimal choice which achieves this goal with minimal total computational effort depends on the the class of problem at hand. Furthermore, for a set of such problems, how should one run the code to lower the computational resources needed for 5% easiest ones? For the 50% easiest? Or for the 99%-easiest? We refer to the various percentages as quantiles, and to address these questions, we consider an annealer which is ran using $S$ sweeps with $R$ repetitions. If the probability of finding the ground state for a single repetition is $w(S)$, then the total number of repetitions needed to find the ground state with 99% is

$$R = \left\lceil \frac{\log(0.01)}{\log(1 - w)} \right\rceil,$$

(8.5)

where we take the ceiling $\lceil \rceil$ as the number of repetitions of annealing must be an integer. This gives a total annealing time of

$$t_T = t_a \cdot R.$$  

(8.6)
Figure 8.3: Quantile optimisation for $\pm J$ instances on the chimera graph.
8.2. OPTIMISING ANNEALING STRATEGIES

Here \( t_a \) is the annealing time for one repetition which is given by \( t_a = S \cdot N / f \) (when \( S \cdot N \) is large), where \( S \) is the number of sweeps performed in the simulation, \( N \) is the system size and \( f \) is the number of attempted spin updates per second. Since \( w(t_a) \) is a non-trivial function of \( t_a \), the total time \( t_T \) to find the ground state with 99% probability is a non-trivial function of \( t_a \), and therefore one needs to minimise \( t_T \) as a function of \( t_a \) in order to find the optimal running parameters for the algorithm. We then plot \( t_T(t_a) \) in Figure 8.3, obtained from 1000 random problems on the chimera graph (see 8.1). It is evident that the code runs optimally for \( 400 \cdot N / f < t_a < 1000 \cdot N / f \), depending on which quantile the problem belongs to.

We also consider the mean time to find the ground state [146], given by

\[
 t_M = t_a \left\lceil \frac{1}{w} \right\rceil, \tag{8.7}
\]

where we replaced \( \log(0.01) / \log(1 - w) \) by \( 1 / w \) in eq. 8.6. If \( w \) is small, \( t_M \) is a time to find the ground state with 63% probability.
CHAPTER 8. OPTIMISED SIMULATED ANNEALING FOR ISING SPIN GLASSES
The combination of disorder and frustration in spin glasses [158] creates a complex energy landscape with many local minima that makes finding their ground state a formidable challenge. In particular, finding the assignments of spins $s_i = \pm 1$ which minimises the total energy of an Ising spin glass with Hamiltonian
\[
H = \sum_{ij} J_{ij}s_is_j + \sum_i h_is_i,
\]
where $s_i = \pm 1$ and $J_{ij}, h_i \in \mathbb{R}$, is non-deterministic polynomial (NP) hard [137] and no polynomial time algorithm is known for the hardest instances. NP-hardness also means that any problem in the complexity class NP can be mapped to an Ising spin glass with only polynomial overhead. This includes the travelling salesman problem, satisfiability of logical formulas, and many other hard optimization problems. Explicit mappings for a number of these problems have recently been given in Ref. [159]. Efficient solvers for Ising spin glass problems hence can have an impact far beyond spin glass physics.

This broad spectrum of applications has also motivated the development of the devices by the Canadian company D-Wave Systems [160, 161, 162, 163]. These devices have been designed to employ quantum annealing [164] for Ising spin glass problems using superconducting flux qubits. However, it has not yet been shown that they can outperform classical devices [165, 166]. Determining the complexity of solving the spin glass problems on the so-called “chimera graph”, which is implemented by the hardware of the D-Wave devices, and finding the best classical algorithms for them is important in the search for quantum speedup on these devices [166].

Motivated by these comparisons and the importance of efficiently solving Ising spin glass problems, here we consider the complexity of solving such
problems for random spin glass instances on finite-dimensional lattices, including the chimera graph. In Sec. 9.1 we discuss the effects of non-zero temperature and magnetic field on Ising spin glasses and argue that the absence of correlations outside the spin glass phase allows for polynomial time algorithms. Section 9.2 presents an exact solver based on this idea which solves the system quasi-locally by considering finite patches of the lattice.

Finally, in Sec. 9.3 we present a hierarchical heuristic approach, which recursively solves groups of spins by splitting each group into smaller subgroups. The idea of the approach is to divide the task of finding a cluster of variables which are to be flipped into two parts. The first part locates the sub-set of variables which contains such a cluster with high probability. The second part searches for the cluster inside this sub-set. The idea is that much of the information for locating a good sub-set can be extracted purely from the structure of the cost function and pre-computed before running the optimiser.

For our benchmark problems on two and three dimensional periodic lattices and chimera graphs with random disorder this approach outperforms, to our knowledge, any other solver currently available and scales significantly better than simulated annealing. While we give a qualitative explanation of the advantage of the hierarchical solver, it remains an open theoretical question to give a quantitative argument for its improved scaling. The interested reader can skip directly to this section as it can be understood independently of the scaling analysis earlier in the paper.

9.1 Boundary condition dependence in frustrated spin systems

It is evident that if the fields $h_i$ in Eq. (9.1) are very large, the problem can be solved by simply aligning each spin relative to the field. The problem becomes more difficult at smaller $h_i$, and the meaningful question is whether a phase transition intervenes at some non-zero value of the field strength, where the difficulty increases greatly. In this section, we argue that the relevant transition indeed is already known in the literature, where it is referred to as the de Almeida-Thouless line. We argue that above this transition (which happens for any non-zero random choice of $h_i, J_{ij}$ in two dimensions), the problem can be solved by considering larger patches of spins, with the patch size diverging as the field strength goes to zero; the spins in the middle of these large patches become independent of those outside the patch and can be fixed using a local algorithm. We first review the relevant literature at
9.1. BOUNDARY CONDITION DEPENDENCE IN FRUSTRATED SPIN SYSTEMS

$h_i = 0$ and the scaling theory at small $h_i$.

9.1.1 Review

In the particular ensemble where fields vanish ($h_i = 0$), the behaviour of the model depends strongly upon both the dimensionality of the system and upon the choice of the ensemble for the couplings between spins. In this discussion, we will focus on the case of a continuous distribution, e.g., a Gaussian one, with vanishing mean.

We will also refer to results in the literature that study nearest-neighbour couplings on a square or cubic lattice, rather than the chimera graph. One important distinction between the two-dimensional square lattice and the chimera graph is that for the square lattice, as for any planar graph, if the magnetic fields vanish there are efficient polynomial time matching algorithms for finding exact ground states \[167\], while on non-planar graphs, such as the chimera, it is NP-hard. We discuss this further below.

In two dimensions it is accepted that there is no spin glass phase at temperature $T > 0$ \[168\]. To quantify this, consider a pair of sites $i, j$. Let $\langle \ldots \rangle$ denote the thermal average of an operator at temperature $T$ and let $[\ldots]_H$ denote the disorder average over Hamiltonians $H$. Since the couplings are chosen with zero mean, we have that $[\langle s_is_j \rangle]_H = 0$ exactly. However, generically the ground state is unique and hence $[(\langle s_is_j \rangle)^2]_H = 1$ at $T = 0$, and this average is expected to be positive at $T > 0$; however, the average vanishes in the limit of large distances between $i, j$.

The reason for the absence of a spin glass phase is that it costs very little energy to flip a domain of spins. Consider flipping a cluster of spins of linear size $\ell$. In a ferromagnetic state, this costs energy proportional to $\ell$. In a spin glass ground state, it is possible, however, that a cluster can be found which costs very low energy to flip. Using various methods of generating flipped patches (by for example boundary condition changes), it is found that the energy of the domain wall scales proportional to $\ell^\theta$ with $\theta \approx -0.282(2)$ \[168\]. Thus, it costs less energy to flip larger clusters, and no matter how small $T$ is, for $T > 0$ there eventually will be some $\ell$ such that flipping clusters at that scale costs energy smaller than $T$. Hence there will be many thermally excited domain walls. On the other hand, for three dimensions and higher, there is believed to be a phase transition temperature $T_c > 0$ with a domain wall exponent $\theta > 0$ for excitations above the ground state \[169\].

Similarly, we can consider random models with nonzero fields \[170, 171, 172\] and denote standard deviation of the field magnitude by $h$. In this case,
we consider the quantity

\[
\left( \langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle \right)^2_H.
\]

If this quantity tends to a non-zero limit at large distance between \(i, j\), then we term this a spin glass phase. It has been shown that such a spin-glass phase can exist in a mean-field model at \(h \neq 0\) \[173\]; the line in the \(h - T\) plane separating the spin glass from the paramagnetic phase is termed the de Almeida-Thouless line. However, it is unclear whether such a spin glass phase at \(h \neq 0\) can persist in a local finite-dimensional model. Numerical work \[174, 175\] suggests that it exists for dimension \(d > d_c = 6\). However, it is accepted that the spin glass phase at \(h \neq 0\) does not persist in dimension \(d = 2\) and in the next subsection we will explain why this is expected given the exponent \(\theta\) discussed above.

It should be emphasised that it is not necessarily difficult to find ground states in a spin glass phase, as exemplified by the matching algorithm for the planar case in \(d = 2\) at \(h = 0\). Conversely, even if a random ensemble is not in the spin glass phase, particular instances may be difficult, as exemplified by the fact that in \(d = 2\) at \(h \neq 0\) the model is not in a spin glass phase, but finding the ground state of arbitrary instances is still NP-hard.

### 9.1.2 Weak Field Scaling in \(d = 2\)

We now consider the effect of a weak magnetic field \(h \neq 0\) in \(d = 2\). Our general goal is to show that in this case, we expect that the value of a given spin in the ground state can often be fixed using a purely local calculation. The argument is a version of the Imry-Ma argument applied to disordered systems \[176\] and in the specific application to spin glasses is an example of the droplet picture \[177\]. We conjecture that a similar argument (with different exponents) will work if there is no de Almeida-Thouless line (i.e., whenever there is no spin glass phase at non-zero magnetic field).

Consider a spin \(s_{\text{cent}}\) at the center of a patch of size \(\ell\) inside a larger system of linear size \(L\). Suppose that we have found some configuration of spins which is a ground state. At \(h = 0\), it is impossible to know whether \(s_{\text{cent}} = +1\) or \(s_{\text{cent}} = -1\) without knowing the value of the boundary spins because there is a \(Z_2\) symmetry. However, at \(h \neq 0\), it may be possible to determine the value of the spin \(s_i\) independent of the value of the boundary spins. That is, there may be some choice (either \(s_{\text{cent}} = +1\) or \(s_{\text{cent}} = -1\)) that minimises the energy inside the patch for all choices of boundary spins. In this case, we know that in the global ground state the spin \(s_{\text{cent}}\) will take the given value. To analyse the ability to fix the spin independently...
9.1. Boundary Condition Dependence in Frustrated Spin Systems

Figure 9.1: If the central spin is forced to be opposite to its optimal orientation while keeping the spins on the boundary of the patch fixed, a cluster of spins around it will also flip. Central spin marked in yellow, flipped cluster marked in red and the boundary in black.

of boundary conditions, we again begin with the case $h = 0$ to develop a scaling argument that will apply at small $h$. Consider a given configuration of boundary spins, which we write as $s_{\text{bdry}}$, where we write this as a vector to emphasise that there are many boundary sites. At $h = 0$, we can minimise the energy inside the patch for this choice of boundary spins, uniquely fixing all spins inside the patch. Suppose that this minimisation gives $s_{\text{cent}} = +1$. Now consider the case in which we force $s_{\text{cent}} = -1$, defining a new configuration of spins inside the patch which minimises the energy subject to the given boundary conditions $s_{\text{bdry}}$ and given that $s_{\text{cent}} = -1$. Forcing $s_{\text{cent}}$ to take the opposite value will flip also a cluster of spins around the central spin, creating a domain wall around that cluster of spins, as shown in Fig. 9.1. The energy of this domain wall will be proportional to $\ell^\theta$ which therefore decreases with increasing $\ell$. The number of spins in the cluster scales also as a power of $\ell$, with the power slightly less than $[178] 2$; in that reference, the exponent $1.80(2)$ was found for one specific method of constructing droplets. Our numerical studies, shown in Fig. 9.2, indicate that the number scales as $\ell^{d_{\text{clust}}}$, with a fractal dimension $d_{\text{clust}} \approx 1.84$; while this dimension might revert to 2 for larger system sizes, we use the fractal dimension extracted at these system sizes to facilitate comparison with our complexity analysis below.

This cluster then defines a larger effective spin. The cost to flip this effective spin relative to the rest of the patch is proportional to $\ell^\theta$. We now consider the case that $h \neq 0$, and analyse the effect of the non-zero $h$ on this effective spin. Given that the magnetic fields acting on the spins in the
For $\ell$ larger than this number, the coupling of the cluster to the effective field exceeds its coupling to the rest of the patch, so that the value of the
9.1. BOUNDARY CONDITION DEPENDENCE IN FRUSTRATED SPIN SYSTEMS

The above scaling analysis gives an estimate of the length scale at which we can fix the central spin in a patch. The total number of spins which can be fixed in the system depends on the local fields and patch size and can be estimated from the probability of fixing a single spin. To quantify this probability we define

\[ \chi_B(h, \ell) = 1 - \left( \langle [s_c]_B \rangle^2 \right)_H \]  

(9.5)

where \( s_c \) is the central spin and where \( [\ldots]_B \) denotes the average over boundary conditions. We term this quantity \( \chi_B \) as it measures the response of the central spin to change in boundary conditions. If this quantity is equal to 0, then the spin can be fixed independently of boundary conditions as it assumes the same value for all choices of boundary. For this averaged quantity,
we find a scaling collapse as shown in Fig. 9.4. The scaling collapse onto a single curve is implement by defining the scaling variable \( \xi = h \cdot \ell^{1.19} \). This implies a scaling 
\[
\ell \sim h^{1/1.19} = h^{0.840...} .
\]
This should be compared with the estimate in Eq. (9.4); the agreement of exponents is reasonable, and if we use \( \theta = -0.28 \) instead of our measured \( \theta = -0.33 \) the agreement becomes more accurate. We find that the scaling collapse can be approximately fit by the form
\[
\chi_B(h, \ell) = \exp (-\text{poly}(h, \ell)).
\] (9.6)

To obtain statistical information about whether we can fix a spin independent of the boundary, it suffices to determine the behaviour of \( \chi_B \) in the tail, see Fig. 9.4, where we fit \( \chi_B \) with the constants \( a \) and \( b \). We cannot be completely confident about the tail behaviour of \( \chi_B \) at large \( h, \ell \) from these simulations, but let us use this estimate to try to determine the complexity of a simple solver which tries to solve each spin by taking a sufficiently large patch that \( \chi_B = 0 \). The complexity of the solver will depend upon the scaling of \( \chi_B \), but we will estimate that it takes a polynomial time (in \( N \)) for any non-zero \( h \). We will find in the next section that we can
improve on this, by using the fact that once a single spin is fixed it simplifies
the fixing of other spins.

Since there are only $2^{4\ell}$ possible boundary conditions, the minimum non-
zero value of $\chi_B$ is of order $2^{-4\ell}$. Considering the $N$ possible choices for the
central spin, only $O(1)$ spins correlate with the boundary if $\exp \{ -\text{poly}(h, \ell) \} =
O(1/N)2^{-4\ell}$. Equivalently, this holds if $2^{4\ell}\exp \{ -\text{poly}(h, \ell) \} = O(1/N)$; since
$\zeta_b > 1$, the scaling of the left-hand side of this equation is dominated by the
second term. Hence, the equation will hold when

$$h \cdot \ell^{1.19} \sim \log(N)^{1/b}. \quad (9.7)$$

Thus, we expect that for $\ell$ larger than this, it will be possible to fix all spins.

Since each patch can be solved exactly with complexity $\exp \{ \ell \}$ using a
dynamic programming method [181], at a fixed $h$ the whole system can be
solved with complexity

$$\text{poly}(L) \exp \{ h^{-1/1.19} \cdot (\log N)^{1/(1.19 \cdot b)} \}. \quad (9.8)$$

Since $b > 1$ and $1/(1.19 \cdot b) < 1$, the exponential term is sub-linear and the
total complexity is therefore polynomial; however, it diverges as $h \to 0$. The
exact estimate may depend sensitively upon the tail of the curve which we
cannot determine with full confidence.

It should, however, be emphasised that the data in Fig. 9.4 arises only
from an average over a finite number (in this case, 1000) of boundary condi-
tions. This finite number was chosen to enable rapid sampling of the curve.
To exactly solve a specific sample, we need to consider all possible boundary
conditions, as discussed in the next section.

9.2 Finding the exact global ground state

Following the argument above, correlations in a typical finite-dimensional
lattice decay exponentially if $h > 0$ and the ground state for such a system
can therefore be found in polynomial time as the optimal orientation of single
spins can be determined with high probability by considering only finite
regions of the system. Furthermore, even for zero fields we present strong
numerical evidence that the typical two-dimensional case can be solved in
polynomial time with a more general approach which we describe below.

9.2.1 Single spin reduction

Let us consider a spin in the center of a patch in our system. If for all
boundary configurations of the patch the optimal orientation of the central
spin is the same, then it is independent of the boundary and can thus be fixed to that value. Based on this idea, the simplest way to find the ground state is by determining the optimal orientation of each spin independently by building a patch around it and checking if the optimal orientation of the central spin is independent of the boundary. If this is not the case, we increase the patch size and check again until the spin becomes independent of the boundary. When all spins are fixed the system is solved.

This approach can be further improved by solving the system similar to a crossword puzzle rather than considering each spin independently. If a spin gets fixed, this will reduce the number of possible configurations for patches containing that spin, which in return may allow more spins to get fixed without increasing the patch sizes.

Fixing single spins is a simple algorithm which can be very efficient for systems with large fields. In the limit of very large fields the complexity approaches $O(N)$ as each spin becomes independent of its neighbours. However, for small fields the computational effort increases as the correlation length diverges when the field approaches zero requiring patches comparable to the total system size. A more general approach, discussed next, remains effective in that limit.
9.2.2 Patch reduction

Instead of only attempting to fix the central spin, correlations between spins inside a patch can be captured by considering all possible configurations of a patch that minimise the energy for a given choice of boundary conditions (see Fig. 9.5). These configurations are then constrained by requiring consistency between overlapping patches. We find numerically that this approach is significantly more efficient than the single spin algorithm.

The algorithm starts with a small patch size (e.g. a single spin in the center) and sequentially builds patches around each spin. For each boundary configuration of a given patch we store the configuration of the boundary together with the corresponding optimal configuration of the center spins. If the local ground state of a patch turns out to be degenerate for a given boundary condition, we arbitrarily pick any of these configurations if our aim is to obtain just one of the potentially degenerate global ground states. Note that if instead we are interested in finding all ground states, then for each boundary configuration all degenerate interior configurations need to be stored.

The number of potential ground state configurations within a patch (boundary and interior) is then further reduced by removing those configurations which are inconsistent with the constraints imposed by overlapping patches.

After a pass through all spins we increase the patch size and repeat the above steps with larger patches until only a single configuration remains or all remaining configurations have the same energy. As the patch size increases, the set of configurations which satisfy all constraints is strongly reduced and typically scales much better than the exponential worst case.

9.2.3 Improved patch reduction

One way to significantly reduce the cost of storing configurations is by removing some spins from the system. If for a pair of neighbouring spins $s_i$ and $s_j$, their product $s_is_j$ is constant in all configurations, they can be replaced by a single spin. If only one ground state is targeted, this procedure will finally eliminate all, but one spin. More generally, any arbitrary spin can be removed by replacing it with multi-spin interactions such that for each configuration of the neighbouring spins the local energy is conserved given that the spin to be removed aligns optimally with respect to its neighbours.
Figure 9.6: Median wall clock time (in seconds) for different system sizes and various fields

Figure 9.7: Scaling exponent for the runtime shown in Fig. 9.6, obtained from fitting the data to a power law, for different system sizes and various fields
9.2.4 Empirical scaling

As shown in Figs. 9.6 and 9.7 the median time to solution appears to scale polynomially in the number of spins for all values of the field $h$, including zero field\(^1\). While faster specialised exact solvers are available [183], this algorithm is not necessarily intended as a general purpose optimiser, but rather to demonstrate polynomial scaling in the number of spins at all values of $h$ for typical low-dimensional spin glass instances.

9.3 Hierarchical search

In this Section we present a general purpose heuristic hierarchical algorithm for finding the ground state of Ising spin glasses based on recursively optimising groups of variables. Before describing the algorithm we motivate why solving groups of variables is significantly more efficient than solving the whole system at once.

The arguably simplest heuristic algorithm for finding the ground state is by generating random spin configurations and recording the energy, in other words random guessing. The probability to find the global ground state of $N$ spins this way is trivially $2^{-N}$ per guess, assuming for simplicity a non-degenerate ground state in the discussion here and below. A more sophisticated way to guess the solution is to generate random configurations of only $N_r = N - N_g$ spins and for each configuration find the lowest energy of the remaining $N_g$ spins by some other algorithm, e.g. by enumerating all possible combinations. This improves the probability of guessing the correct solution, but as the cost of finding the optimal orientation of the remaining $N_g$ variables may be as much as $2^{N_g}$, we might not have gained much. This idea can, however, be extended to solving multiple groups. Let’s consider two groups with $N_1$ and $N_2$ spins respectively, chosen such that spins in one group do not couple to any of the spins in the other group. For each random guess of the remaining $N_r = N - N_1 - N_2$ spins, the complexity of finding the optimal configuration of both of them with respect to the rest of the system is $2^{N_1} + 2^{N_2}$, thus reducing the total complexity by an exponential amount from $2^N = 2^{N_r + N_1 + N_2}$ to $2^{N_r} (2^{N_1} + 2^{N_2})$. In our algorithm, described below, we find a significant reduction in complexity even if spins in subgroups are coupled and overlap with each other.

\(^1\)The median was calculated from 5280 random instances for each system size and field strength. In our implementation we use a library [182] for binary decision diagrams to store these configurations which we found to be significantly more efficient than using simple boolean tables.
9.3.1 Optimization of groups

The above argument provides a basis for a simple algorithm to find the global ground state by iteratively optimising groups of spins. We start with a random state, sequentially pick $M$ groups with $N_g$ spins each and optimise their configurations by calling some – as yet unspecified – solver. The algorithm is outlined in Algorithm 3. Here, \text{SOLVE GROUP}(G)$ is a solver that solves the

\begin{algorithm}[H]
\begin{procedure}
\textbf{Solve}()
\begin{algorithmic}
\State initialise random spin configuration
\For{$j \in \{1 \ldots M\}$}
\State pick a random spin $i$
\State build group $G$ of size $N_g$ around spin $i$
\State $\vec{\sigma} \leftarrow \text{Solve group}(G)$
\State $\text{Update configuration}(G, \vec{\sigma})$
\EndFor
\end{algorithmic}
\end{procedure}
\end{algorithm}

group $G$ (taking into account the interaction with spins outside $G$ to produce an effective field) and returns an optimized configuration $\vec{\sigma}$ for the spins in that group. The procedure $\text{Update configuration}(G, \vec{\sigma})$ updates the spins inside $G$ to configuration $\vec{\sigma}$; if the solver \text{Solve group} is a heuristic solver, then $\text{Update configuration}(G, \vec{\sigma})$ only makes this change if the energy is lowered. Alternatively one may also consider an algorithm which replaces the group configuration probabilistically with a Metropolis-type criterion or similar.

If we pick trivial groups of size $N_g = 1$, consisting of a single spin, the group solver just returns the spin direction which minimises its energy with respect to its neighbours. For larger groups – as will usually be the case – we can use any arbitrary exact or heuristic solver, including potentially special purpose classical or quantum hardware. We note in passing that in the case of $N_g = 1$, if the new configuration is accepted probabilistically depending on its energy this algorithm reduces to simulated annealing.

9.3.2 Hierarchical recursive algorithm

If solving a given system in groups is more efficient than solving the whole system at once, performance can be increased even further by solving each group by subdividing it recursively into sub-groups, thus giving a hierarchical version of the algorithm. That is, in the pseudo code written above, we could
use the function \texttt{SOLVE()}, restricted to the spins in a group, as the solver \texttt{SOLVE GROUP()}. The recursion terminates at some (small) group size, which is solved by another algorithm.

Note that the hierarchical scheme randomises the configuration of each group before solving it by optimising subgroups, thus implementing random local restarts without affecting the global spin configuration. This randomisation also implies that it makes no sense to solve a particular group more than once in a row, but rather a new group should be chosen after one group has been optimized. It should be emphasised that random restarting is just one possible way to initialise the state of a group and the one we used here. Other ways are possible and could be more efficient. We explore one such method, termed partial reinitialisation, in Section 10.

The total complexity of the hierarchical algorithm is dominated by the number of calls to the solver for the bottom level group rather than by the group size at each level. This is because for a given group of size $N_g$, the effort to calculate the local energy and randomise spins is at most $O(N_g^2)$ for dense graphs, which is typically negligible relative to the effort of finding a lower energy configuration of that group.
9.3.3 Selecting groups

Up to now, we ignored the hard problem of how to best pick groups. Here we provide a simple strategy that turned out to work well. Intuitively, in a well chosen group spins are strongly coupled to each other and more weakly coupled to the rest of the system, see Fig. 9.8. We thus build a group $G$ by starting from one spin and greedily adding spins until the group $G$ has reached the desired size. We add the spin $i$ that maximises $W_i = \sum_{j \in G} |J_{ij}| - \sum_{j \not\in G} |J_{ij}|$, if this maximum is positive and a random neighbour of one of the spins in $G$ otherwise.

Other ways of building a group may be more effective. For example, single spins could be added probabilistically, or instead of single spins we could consider sets of spins which can be added to the group.

9.3.4 Results

To test the performance of our algorithm we compare it to simulated annealing, which is currently one of the most versatile and efficient solvers for finding ground states of spin glasses. As mentioned above, simulated annealing is a special case of our algorithm. For our benchmarks we perform hierarchical search with two levels, using simulated annealing to solve groups of size $N_1$ with the optimised configuration being accepted if its energy is lower or the same as the current configuration.

As a measure of complexity we use the median total number of spin updates required to find the ground state with a target probability $p_0 = 0.99$. Since a heuristic algorithm will find the ground state with some probability $p_s < 1$ we may have to repeat the optimization multiple times if $p_s < p_0$. Assuming independent repetitions, the required number of repetitions is $R = \lceil \log (1 - p_0)/\log (1 - p_s) \rceil$ (see Section 8.2.2 for more details). For each set of parameters the probability $p_s$ was estimated by performing 1024 repetitions from random initial states.

For both algorithms and each class and size of problems we optimise the simulation parameters to minimise the median effort in terms of single spin updates. For simulated annealing the total effort for a single repetition is $SN$, there $S$ is the number of sweeps and $N$ is the system size. We used a linear schedule in $\beta = 1/T$ where the initial and final values of inverse temperature, $\beta_0$ and $\beta_1$ respectively, as well as the number of sweeps $S$ are chosen to minimise the total effort. We list the parameters used in Tables 9.1 – 9.5.

For the hierarchical approach a single repetitions requires a total effort $MS_gN_g$, where $M$ is the number of groups, $S_g$ is the number of simulated
9.3. HIERARCHICAL SEARCH

<table>
<thead>
<tr>
<th>$N$</th>
<th>$M$</th>
<th>$N_g$</th>
<th>$S_g$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>78</td>
<td>9</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>72</td>
<td>80</td>
<td>37</td>
<td>5</td>
<td>24</td>
</tr>
<tr>
<td>128</td>
<td>100</td>
<td>60</td>
<td>7</td>
<td>64</td>
</tr>
<tr>
<td>200</td>
<td>349</td>
<td>41</td>
<td>4</td>
<td>192</td>
</tr>
<tr>
<td>288</td>
<td>408</td>
<td>68</td>
<td>6</td>
<td>400</td>
</tr>
<tr>
<td>392</td>
<td>500</td>
<td>105</td>
<td>13</td>
<td>1024</td>
</tr>
<tr>
<td>512</td>
<td>642</td>
<td>129</td>
<td>14</td>
<td>2048</td>
</tr>
</tbody>
</table>

Table 9.1: Optimal parameters for chimera graphs with random bimodal disorder. $N$ is the system size, $M$ is the number of groups, $N_g$ is the group size, $S_g$ is the number of simulated annealing sweeps per group, $S$ is the number of sweeps for plain simulated annealing.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$M$</th>
<th>$N_g$</th>
<th>$S_g$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>28</td>
<td>8</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>72</td>
<td>237</td>
<td>9</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>128</td>
<td>388</td>
<td>9</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>200</td>
<td>197</td>
<td>26</td>
<td>3</td>
<td>1281</td>
</tr>
<tr>
<td>288</td>
<td>454</td>
<td>29</td>
<td>3</td>
<td>4800</td>
</tr>
<tr>
<td>392</td>
<td>470</td>
<td>27</td>
<td>3</td>
<td>24576</td>
</tr>
<tr>
<td>512</td>
<td>718</td>
<td>28</td>
<td>3</td>
<td>131072</td>
</tr>
</tbody>
</table>

Table 9.2: Optimal parameters for chimera graphs with cluster bimodal disorder. The parameters have the same meaning as in Tab. 9.1.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$M$</th>
<th>$N_g$</th>
<th>$S_g$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>72</td>
<td>8</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>64</td>
<td>314</td>
<td>9</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td>144</td>
<td>273</td>
<td>33</td>
<td>21</td>
<td>891</td>
</tr>
<tr>
<td>256</td>
<td>573</td>
<td>47</td>
<td>43</td>
<td>30189</td>
</tr>
</tbody>
</table>

Table 9.3: Optimal parameters for two dimensional lattices with Gaussian disorder. The parameters have the same meaning as in Tab. 9.1.
CHAPTER 9. FROM LOCAL TO GLOBAL GROUND STATES IN ISING SPIN GLASSES

Table 9.4: Optimal parameters for three dimensional lattices with Gaussian disorder. The parameters have the same meaning as in Tab. 9.1.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$M$</th>
<th>$N_g$</th>
<th>$S_g$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>39</td>
<td>10</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>64</td>
<td>118</td>
<td>22</td>
<td>4</td>
<td>45</td>
</tr>
<tr>
<td>125</td>
<td>232</td>
<td>30</td>
<td>5</td>
<td>512</td>
</tr>
<tr>
<td>216</td>
<td>271</td>
<td>58</td>
<td>23</td>
<td>2700</td>
</tr>
<tr>
<td>343</td>
<td>562</td>
<td>86</td>
<td>42</td>
<td>13056</td>
</tr>
<tr>
<td>512</td>
<td>614</td>
<td>113</td>
<td>101</td>
<td>61440</td>
</tr>
</tbody>
</table>

Table 9.5: Optimal parameters for two dimensional lattices with bimodal disorder. The parameters have the same meaning as in Tab. 9.1.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$M$</th>
<th>$N_g$</th>
<th>$S_g$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>235</td>
<td>56</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>400</td>
<td>224</td>
<td>119</td>
<td>23</td>
<td>227</td>
</tr>
<tr>
<td>576</td>
<td>384</td>
<td>103</td>
<td>17</td>
<td>768</td>
</tr>
<tr>
<td>784</td>
<td>686</td>
<td>208</td>
<td>31</td>
<td>3506</td>
</tr>
<tr>
<td>1024</td>
<td>656</td>
<td>151</td>
<td>29</td>
<td>7680</td>
</tr>
</tbody>
</table>

annealing sweeps per group and $N_g$ is the group size. The same annealing schedule is used for each group. The values of $M$, $S_g$ and $N_g$ are chosen to minimise the total effort and are listed in Tables 9.1 – 9.5. As benchmark problems we used typical spin glass problems on two and three-dimensional lattices: two-dimensional square lattices, three dimensional simple cubic lattices, and so-called two-dimensional chimera graphs. The unit cell of the chimera graph [185], shown in Fig. 9.9, is a complete bipartite graph with eight vertices and is coupled to the neighbouring unit cells with four edges each. Hence, each vertex has either five or six edges corresponding to four edges to spins within the unit cell and one or two edges to neighbouring unit cells depending on if it is on the edges of the graph or in the interior respectively.

One choice of benchmark problems are spin glasses with bimodal disorder i.e., couplings $J_{ij} = \pm 1$ and another choice will be Gaussian disorder with couplings drawn from a normal distribution with zero mean and unit variance. In all benchmarks we use zero local fields.

A special benchmark problem is chimera graphs with cluster structure,

\footnote{We used the spin glass server [184] to computed the exact ground states for three dimensional lattices}
Figure 9.9: Chimera graph with 512 spins composed of an $8 \times 8$ grid of unit cell. Each unit cell is a complete bipartite graph with 8 spins.
which has recently been proposed as a class of problems to explore an advantage of quantum annealing over simulated annealing [186]. In these problems the spins within each unit cell are coupled ferromagnetically with $J_{ij} = -1$. Of the four edges connecting neighbouring pairs of unit cells one randomly chosen edge is assigned a random coupling $J_{ij} = \pm 1$ and the rest is set to zero.

In all benchmarks we find that hierarchical search performs significantly better than simulated annealing. The gain is evidently more significant for problems that are harder for simulated annealing, such as cluster chimera graphs and systems with Gaussian disorder, see Fig. 9.11, 9.12 and 9.14 respectively. Random bimodal disorder is easier for simulated annealing and hence the speedup on those problems is smaller, although still substantial, see Fig. 9.10 and 9.13 respectively.

A further comparison was also made with parallel tempering [147], another state the art method for finding ground states of Ising spin glasses. For each class and problem size, the total number of replicas and sweeps per replica was optimised by minimising the median total number of spin updates. For a single repetition the total effort is $N_R S N$, where $N_R$ is the
number of replicas, $S$ is the number of sweeps per replica and $N$ is the system size. On chimera graphs its performance is very similar to simulated annealing, see Fig. 9.10. On two dimensional lattices with Gaussian disorder it performs slightly better, see Fig. 9.12. However, analogous to simulated annealing its performance can be significantly improved by optimising groups of spins rather than the whole system at once, see Fig. 9.15.

Note that although in all cases the advantage of hierarchical search over plain simulated annealing and parallel tempering grows with problem size, a spin update is effectively more costly due to the additional overhead of randomising the spins and computing the energy of a group. However, the difference is typically insignificant. For example, the wall clock time per spin update is only about 7% higher than plain simulated annealing for $8 \times 8 \times 8$ 3D lattices with Gaussian disorder ran with optimal parameters.

9.4 Conclusion

It has long been established that the complexity of finding ground states of spin glasses is strongly dependent on the ensemble of couplings and is in the worst case NP-hard. However, whilst the most trivial cases, like the ferromagnetic Ising model, are relatively evident, the hardest problems are
Figure 9.12: Two dimensional square lattices with Gaussian disorder. Speedup and the inset are defined the same as in Fig. 9.10. As the energy gap between the ground state and first excited state decreases linearly with system size, the final temperature is also reduced with the number of spins. $\beta_0 = 0.014$, $\beta_1 = 0.037N + 2.5$ for plain simulated annealing (SA) and parallel tempering (PT) and $\beta_1 = 0.037N_g + 2.5$ for each group of the hierarchical algorithm (HS). Optimal parameters for SA and HS are listed in Tab. 9.3.
9.4. CONCLUSION

Figure 9.13: Two dimensional square lattices with bimodal disorder. Speedup and the inset are defined the same as in Fig. 9.10. For both plain simulated annealing and for each group, \( \beta_0 = 0.2, \beta_1 = 3 \). Optimal parameters for both algorithms are listed in Tab. 9.5.

Figure 9.14: Three dimensional cubic lattices with Gaussian disorder. Speedup and the inset are defined the same as in Fig. 9.10. \( \beta_0 = 0.05, \beta_1 = 0.028N + 5.68 \) for plain simulated annealing and \( \beta_1 = 0.028N_g + 5.68 \) for each group of the hierarchical algorithm. Optimal parameters for both algorithms are listed in Tab. 9.4.
far more elusive [187].

One way to look for hard problems is by sampling randomly distributed couplings. Although this approach certainly includes such problems, in this work we presented strong numerical evidence that the average complexity of low-dimensional spin glasses with randomly distributed couplings is actually polynomial in the number of spins and looking for hard problems in such a large ensemble might be next to futile. Another way to generate hard cases is to map 3-SAT problems at the critical clause to variable ratio [188], where previous studies have shown evidence of a universal peak in complexity, to the Ising model. Further studies are to be done in this direction.

Our most significant result reported here is a hierarchical approach as a way to potentially improve the performance of a given algorithm for finding ground states of Ising spin glasses. With simulated annealing as a reference solver, on all our benchmark instances we find that optimising groups of spins is significantly more efficient than solving the whole system at once.

It should be noted that approaches other than simulated annealing can be used at the bottom level of the hierarchical solver. Suppose for some class of problems, another algorithm (or special purpose classical or quantum device) outperforms simulated annealing. In that case, we can use that algorithm or
device at the lowest level. Let $T_0$ denote the time annealing takes to optimise the bottom level. If it is now replaced by a device which takes time, including communication overhead, $T_1 \ll T_0$, we expect the potential speedup to the whole algorithm to be $\sim T_0/T_1$. As the complexity of finding the ground state scales exponentially with the number of spins, this can be significant even for small groups.

Although we limited our investigation to spin glasses, similar ideas can be applied directly to other problems such as machine learning, protein folding, travelling salesman etc. by constraining groups of variables independently relative to the rest of the system.
Chapter 10
Partial reinitialisation for optimisers

A strategy that is frequently used to escape a local optimum is to restart the optimisation process from a new random initial configuration. Repeating this process multiple times, the local optimum with the lowest cost is then returned, which is with high probability better and with certainty no worse than the initial configuration. Although such restarts allow the optimiser to get out of local optima, different restarts are also completely decoupled from each other. That is, information which was learned about the structure of the problem in one restart is not passed on to the next and has to be relearned from scratch. Hence, this way of running an optimiser is effectively coarse grained random guessing, but where each guess is further improved towards a local optimum.

To improve upon randomly guessing the initial state of the optimiser, algorithms combining local and global search are often used, which is a branch of evolutionary computing called memetic algorithms [189, 190, 191]. Memetic algorithms have been applied for a wide range of problems, including combinatorial optimisation [192] and machine learning [193, 194, 195]. Another related algorithm is chained Lin-Kernighan [196, 197] for finding optimal tours in the travelling salesman problem, which achieves significant improvements over standard Lin-Kernighan by perturbing only a few edges in the tour between iterations. Local search has previously also been combined with simulated annealing, comparing favourably to state-of-the-art approaches on standard benchmark problems for continuous optimisation [198].

In this work we present what we believe is a simpler and more general approach which can be applied hierarchically to any multivariate optimisation problem, allowing the global optimum to be found with high probability in a single run. We benchmark our algorithm against standard full reinitialisa-
Figure 10.1: An illustration of the partial reinitialisation algorithm. On each level $l$, subsets of $k_l$ variables are reinitialised (black circles) such that $(k_m = N) > k_{m-1} > \ldots > k_0$. On the bottom level the $k_0$-optimal optimiser is called (yellow circles). Each level starts with the most optimal configuration of its parent and reinitialises a subset of $k_l$ variables before calling the optimiser on the next level (green arrows). Checkpoints of the most optimal configuration found are kept on each level with a flow from left to right (blue arrows).

10.1 Algorithm

Given a core local heuristic optimiser, let us transform the problem of finding an optimal configuration for the original problem to the problem of finding a starting configuration for our optimiser which leads it to that optimum. If the optimisation problem has any kind of structure and given the core optimiser only exploits parts of this structure, which is often the case when using general purpose optimisers, the problem of finding an initial configuration for this optimiser should also have structure.

Taking advantage of this structure by using a more sophisticated optimiser could be significantly better than random guessing. The cost of a given configuration seen by this higher-level optimiser is now the cost returned by the core optimiser with this configuration as input. The same
10.1. ALGORITHM

Figure 10.2: Chain of optimisers. Input and output configurations of variables are denoted by $v_{in/out}$ respectively and $Cost()$ represents a call to the cost function.

argument can be repeated by adding a third optimiser and so on until any initial configuration leads to the global optimum with high probability when passed through the chain of optimisers. The local optimiser we had initially is now refined into a global one. An illustration of such a chain is shown in Fig. 10.2.

It remains to choose a suitable optimiser for the higher levels. For generality we will here use perhaps the simplest one of them all, which we call partial reinitialisation. Starting from an initial configuration, it reinitialises a small sub-set of variables, rather than the whole configuration, between applications of the local optimiser. If the cost has improved, it accepts the new configuration, otherwise rejects and tries again. This is repeated some pre-defined number of times. By applying these partial reinitialisations interspersed with full reinitialisations, the optimisation algorithm gains the ability to more efficiently explore nearby optima while retaining the ability to find a globally optimal solution.

To better understand why this approach often outperforms conventional full reinitialisations, let us first introduce a notion of optimality for an optimiser. We define an optimiser to be $k_0$-optimal if for any locally optimal configuration, reinitialising fewer than $k_0$ variables causes the optimiser to find the same configuration. However, reinitialising subsets of $k_1 > k_0$ variables may allow the optimiser to find new local optima.

Let us start from a random initial configuration of variables and find a local optimum by applying a $k_0$-optimal optimiser. If the problem has structure, in the sense that the probability of finding an improved local optimum is greater for solutions in the vicinity of a local optimum than it is for an
arbitrary initial configuration, reinitialising a subset of \( k_1 < N \) variables for the next repetition could significantly increase the probability of finding a better local optimum compared to a full reinitialisation.

As subsets of \( k_1 \) variables are reinitialised and the \( k_0 \)-optimum optimiser called after each reinitialisation, the chance of finding a better local optimum decreases and the combined optimiser, including the reinitialisations, becomes \( k_1 \)-optimal.

To find better local optima with this optimiser, subsets of \( k_2 > k_1 \) variables can be reinitialised. Repeating this process iteratively, each time increasing the size of the subset until \( (k_m = N) \), the optimiser becomes \( N \)-optimal and finds the global optimum with high probability. This hierarchical procedure can hence refine a local optimiser into a global optimiser.

The algorithm for the hierarchical partial reinitialisation strategy is outlined in Figure 10.1 and Algorithm 4. With \( m \) levels in the hierarchy, the

**Algorithm 4 Partial reinitialisation**

**Input:** current level \( l \), number of reinitialisations \( M_l \), number of variables for each reinitialisation \( k_l \) and initial variable settings \( x \).

if \( l = 0 \) then
  call optimiser on \( x \)
else
  \( x_0 \leftarrow x \)
  for \( i \in \{1 \ldots M_l\} \) do
    reinitialise subset of \( k_l \) variables in \( x \)
    call partial reinitialisation on level \( l - 1 \)
    if \( \text{cost}(x) > \text{cost}(x_0) \) then
      \( x \leftarrow x_0 \)
    else
      \( x_0 \leftarrow x \)
    end if
  end for
end if

algorithm is started from the \( m^{\text{th}} \) level. The configuration is denoted by \( x \) and the checkpoints by \( x_0 \); the checkpoints are the most optimal configurations found thus far, to which the algorithm reverts if no improved configuration is found by partial reinitialization. At each level \( l \), \( M_l \) reinitialisations of \( k_l \) variables are performed and \( (k_m = N) > k_{m-1} > \ldots > k_0 \). Taking the cost of running the basic optimiser as \( O(1) \), the complexity of a single run of the algorithm is \( O(\prod_l M_l) \).
The method used to select the variables to be reinitialised can significantly impact the quality of the solution. The simplest approach is to pick variables at random. However, if variables are chosen according to a problem-specific heuristic, the probability that reinitialising a subset of a given size leads to a more optimal configuration can be maximised. One approach is to pick subsets such that the optimality of variables within the set depends on the values of the other variables in the set as much as possible (similar to the approach for building groups of variables in Section 9.3.3). This can increase the chance of getting out of a local optimum by reducing the number of constraints on the subset from the rest of the system.

Note that our approach introduces additional hyperparameters on top of the ones required for the local optimiser. Optimising these parameters to maximise performance requires additional effort which we do not consider in the benchmarks. However, for a given problem class, these parameters are often only tuned once and reused for all problems in this class. The overhead of the tuning is hence a constant which becomes less significant with the number of times it is reused.

If the outcome of the heuristic optimiser does not directly depend on the initial configuration, but also on a random seed, the partial reinitialization strategy could be used to optimise only the variables within a subset, while the other variables in the problem are kept fixed. Such an approach was employed in Section 9.3 for finding ground states of Ising spin glasses with simulated annealing and showed significant speedup relative to conventional global restarts.

If the optimisation problem is over the space of continuous variables, the concept of partial reinitialisation can be extended to partially reinitialising each variable in addition to subsets of variables. That is, rather than setting a variable to a random value within a pre-defined domain, it can be perturbed by, for example, adding noise with some standard deviation $\sigma$, that is

\[
x = \alpha x + (1 - \alpha)N(\mu, \sigma)
\] (10.1)

where $\mu$ is the mean and $\sigma$ is the standard deviation of the distribution $N$. Hence, we can either fully reinitialise subsets of variables, add small perturbations to all variables, or combine the two and partially perturb subsets of variables to further improve performance. For simplicity we use $\alpha = 0$ in the benchmarks below.

### 10.1.1 Choosing $M_\ell$

Although $M_\ell$ can be chosen empirically to optimise the performance of an optimiser over a set of randomly chosen optimisation problems, there is also
a theoretical basis behind the choice. $M_\ell$ should be chosen to ensure that the probability of being in a local optimum, with respect to reinitialisation of $k_\ell$ variables from the input, is maximised. We call such a configuration \textit{probabilistically $k_\ell$-optimal}.

If we require that the probability that a given reinitialization of $k_\ell$ variables does not improve the optimum is less than $1 - \delta$ and assert that the probability that this reinitialisation improves the objective function is greater than $\epsilon$, then it suffices to take \cite{199}

$$M_\ell \geq \left\lceil \frac{\ln(\delta)}{\ln(1 - \epsilon)} \right\rceil.$$ (10.2)

Thus the values of $\delta$ and $\epsilon$ specify the value of $M_\ell$ needed to conclude that the local optimum is probabilistically $k_\ell$-optimal within a margin of error. Furthermore, if we consider $1 - \epsilon$ to be a constant, only a logarithmic number of samples are required to conclude with high probability that the value is $k_\ell$-optimal. If, on the other hand, $\epsilon$ is small, such as in unstructured search, then the number of requisite reinitialisations can be exponential in the number of variables. Thus, the value of $\epsilon$ makes an implicit assumption about the structure and complexity of the problem.

### 10.2 Benchmarks

In the following, we will study the advantage of partial reinitialisations compared to standard full reinitialisations on a set of machine learning problems. These problems are picked from the ones with which we have most experience. The size of each problem was chosen to be large enough such that finding the global optimum is non-trivial using the respective standard algorithms.

It is worth noting that in machine learning a local optimum is sometimes sufficient or even more desired than a global optimum due to overtraining \cite{200}. However, as we show below, partial reinitialisation improves not only the quality of the final solution, but also the speed at which a local optimum of a given quality is obtained. A range of techniques can also be used to reduce overtraining and improve the generality of the model, including cross-validation \cite{201}, pruning for decision trees \cite{202} and different forms of regularisation \cite{203} and model selection \cite{204, 205}.

For simplicity we use only one level in the hierarchy between a full reinitialisation and calling the heuristic optimiser. That is, for each full reinitialisation, multiple reinitialisations of subsets of variables are performed. To maintain generality, we choose subsets at random for all benchmark problems.
10.2. BENCHMARKS

The parameters in the benchmarks, such as the size of each subset (denoted by $k_1$) and the number of partial reinitialisations (denoted by $M_1$) performed within each full reinitialisation, were manually optimised and are not the true optima for the respective performance metrics.

As a performance measure for each benchmark we use the most optimal cost obtained after a given elapsed time or number of iterations averaged over multiple runs with different random initial states. Elapsed time was measured on Intel Xeon E5-2660 v2 processors.

10.2.1 Training hidden Markov models

Learning temporal patterns in a signal is of central importance in a wide range of fields including speech recognition, finance and bioinformatics. A classic method to model such systems is hidden Markov models (HMM), which are based on the assumption that the signal follows a Markov process. That is, the future state of the system depends solely on the present state without any memory of the past. This assumption turns out to be
surprisingly accurate for many applications.

In discrete HMMs, which we will consider here, the system can be in one of $N$ possible states hidden from the observer. Starting from a discrete probability distribution over these states, as time evolves the system can transition between states according to an $N \times N$ probability matrix $A$. Each hidden state can emit one of $M$ possible visible states. The model is hence composed of three parts: the initial probability distribution of length $N$ over the hidden states; the $N \times N$ transition matrix between hidden states; the $N \times M$ emission matrix from each hidden state into $M$ possible visible states. During training on a given input sequence, these matrices are optimised such as to maximise the likelihood for this sequence to be observed.

The standard algorithm for training HMMs is the Baum-Welch algorithm [206]. It is based on the the forward-backward procedure, which computes the posterior marginal distributions using a dynamic programming approach. The model is commonly initialised with random values and optimised to maximise the expectation of the input sequence until convergence to a local optimum. To improve accuracy, multiple restarts are usually performed. Over the sequence of restarts, we will investigate if partial reinitialisation can improve the convergence rate towards a global optimum.

As a benchmark we choose learning a random 128-bit string. Although artificial, this problem is very simple and has few free parameters. At the same time, finding a model which accurately represents the sequence is non-trivial when starting from random transition and emission matrices. If the number of hidden states is at least as large as the length of the bit-string, the model can be trained until that bit-string is observed with certainty within the model. Here we will use two visible states and the same number of hidden states as bits in the input. For generality we do not impose any restrictions on the transition matrix.

Each full reinitialisation starts with random emission and transition matrices. In a partial reinitialisation only the elements in the matrices corresponding to a subset of hidden states are reinitialised. We found that about 8000 reinitialisations of 8 variables within each global restart is optimal and leads to a much more rapid increase in accuracy with training time than with full reinitialisations (see Fig. 10.3).

### 10.2.2 Clustering with k-means

Dividing objects into clusters according to a similarity metric is of central importance in data analysis and is employed ubiquitously in machine learning. Given a set of points in a finite-dimensional space, the idea is to assign points to clusters in such a way as to maximise the similarities within a clus-
10.2. BENCHMARKS

Figure 10.4: The benchmark is the A3 set from [207]. Median within-cluster sum of squares (WCSS) and elapsed time over 1000 runs of the k-means algorithm is shown. The blue curve is full reinitialisations and the red curve is partial reinitialisations.

ter and minimise the similarities between clusters. Similarity can be defined in many ways, depending on the particular needs of the application. Here we use the Euclidean distance.

One of the most widely used algorithms for finding such clusters is the k-means algorithm [208]. K-means searches for an assignment of points to clusters such as to minimise the within-cluster sum of square distances to the center. That is, it seeks to minimise the following cost function

\[ \sum_{i=1}^{N_c} \sum_{j \in C_i} \| x_j - \mu_i \|^2 \]

where \( N_c \) is the number of clusters. Starting from a random initialisation of centres, each iteration proceeds in two stages. First all points are assigned to the nearest cluster center. In the second part each center is picked to be the Euclidean center of its cluster. This is repeated until convergence to a local optimum. As the cost is never increased, convergence is guaranteed. It has been shown that there exists problems on which k-means converges in exponentially many iterations in the number of points [209]. The smoothed running time is, however, polynomial [210] and it typically converges very
quickly. Similar to the Baum-Welch algorithm, multiple global reinitialisations of centres are often performed to improve the quality of the clusters.

Our benchmark problem is set A3 in a standard clustering data set [207]. The clusters have points sampled from Gaussian distributions. There are 7500 points around 50 uniformly placed centres. Each full reinitialisation starts with an initial assignment of all centres to random points using Forgy’s method [211]. In a partial reinitialisation only a single cluster center is reinitialised. We found about 100 partial reinitialisations for each global restart to be optimal.

Except in the very beginning, a given quality of clusters is obtained significantly quicker with partial rather than full reinitialisation (see Fig. 10.4). The reason full reinitialisation is here more efficient at finding low quality clusters is because in the regime where the cost is high, larger strides in reducing the cost can be made by just random guessing than optimising a bad global restart further with partial reinitialisations. However, as the cost threshold becomes harder to reach, partial reinitialisation quickly becomes more efficient.

10.2.3 Clustering with k-medoids

An often more robust approach than k-means for clustering data, called k-medoids, picks the best cluster center to be one of the points in the cluster rather than the Euclidean center. The standard algorithm for finding such clusters is partitioning around medoids (PAM) [214]. Similar to the k-means algorithm, PAM iterates between assigning points to their closest center and finding the best center of each cluster until convergence.

As the centres are constrained to the positions of the points and no cluster assignment can occur twice during a run, the trivial upper bound on the number of iterations until convergence is \( \binom{N}{N_c} \) for \( N \) points and \( N_c \) clusters. As with k-means, however, it typically converges very quickly.

The benchmark problem we use here is clustering 900 images of faces from the Olivetti database [212]. This dataset has been used previously to benchmark a novel method for finding k-medoids clusters, called affinity propagation [215], against PAM. Here we repeat this benchmark, but also compare to PAM with partial reinitialisations.

We chose to find 121 clusters as this was roughly the regime where affinity propagation had the largest advantage over k-medoids [215]. Within each global restart, we found that about \( 0.5 \cdot 10^6 \) reinitialisations of only a single randomly chosen cluster center is optimal. We use a pre-computed similarity matrix, available from [216], the entries of which are the pixel-wise squared distances between the images after some pre-processing.
Figure 10.5: The benchmark problem is clustering 900 images of faces from the Olivetti face database [212] into 121 clusters. Median within-cluster sum of squares (WCSS) and elapsed time over 1000 runs of k-medoids is shown. The blue curve is full reinitialisations and the red curve is partial reinitialisations. The green curve shows the performance of affinity propagation with timings done using the web interface [213].

Affinity propagation indeed quickly finds much better clusters than PAM with full reinitialisations [213]. However, after some time PAM with partial reinitialisations finds even better clusters which keep improving even further with time (see Fig. 10.5). Full reinitialisation finds better clusters in the beginning for similar reasons as k-means, discussed above.

10.2.4 Training Boltzmann machines

Boltzmann machines are a class of highly generalisable models, related to feed-forward neural networks, that have proven to be very useful for modelling data sets in many areas including speech recognition and computer vision [217, 218, 219]. The Boltzmann machine takes the following form: Let us define two layers of units, which we call the visible layer and the hidden layer. The visible units, \( v \), comprise the input and output, whilst the hidden units, \( h \), are latent variables that are marginalised over in order to capture correlations in the data. These units are typically taken to be binary, and
the joint probability of a configuration of visible and hidden units is

\[ P(v, h) = \frac{\exp(-E(v, h))}{Z} \]  

(10.3)

where \( Z \) is a normalisation factor known as the partition function, and

\[ E(v, h) = -v \cdot a - h \cdot b - v^T W h, \]  

(10.4)

is the energy. Here \( W \) is a matrix of weights that models the interaction between pairs of hidden and visible units. \( a \) and \( b \) are vectors of biases for each of the units. This model is commonly known as a Restricted Boltzmann Machine (RBM) and can be viewed as an Ising model on a complete bipartite graph that is in thermal equilibrium. Such RBMs can be stacked to form layered Boltzmann machines which are sometimes called deep Boltzmann machines. For simplicity we will focus on training RBMs since training deep Boltzmann machines using popular methods, such as contrastive divergence training, involves optimising the weights and biases for each layered RBM independently and is hence conceptually similar.

The training process involves optimising the maximum likelihood training objective, \( O_{ML} \), which is

\[ O_{ML} = \mathbb{E}_{v \in \text{train}} \left( \ln \left[ \mathbb{E}_h P(v, h) \right] \right) - \frac{\lambda}{2} \sum_{ij} W_{ij}^2, \]  

(10.5)
where $\lambda$ is a regularisation term introduced to prevent overfitting. The exact computation of the training objective function is $\#P$ hard, which means that its computation is expected to be intractable for large RBMs under reasonable complexity theoretic assumptions.

Although $O_{\text{ML}}$ cannot be efficiently computed, its derivatives can be efficiently estimated using a method known as contrastive divergence. The algorithm, described in detail in [218], uses a Markov chain approach that estimates the expectation values of the hidden and visible units which are needed to compute the derivatives of $O_{\text{ML}}$. Specifically,

$$
\frac{\partial O_{\text{ML}}}{\partial W_{ij}} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}} - \lambda W_{ij}.
$$

(10.6)

Here $\langle \cdot \rangle_{\text{data}}$ denotes an expectation value over the Gibbs distribution of (10.3) with the visible units clamped to the training data and the $\langle \cdot \rangle_{\text{model}}$ denotes the unconstrained expectation value. The derivative with respect to the biases is similar. Locally optimal configurations of the weights and biases can then be calculated by stochastic gradient ascent using these approximate gradients.

As a benchmark we will examine small synthetic examples of Boltzmann machines where the training objective function can be calculated exactly. Although this differs from the task based estimates of the performance of a Boltzmann machine, we focus on it because the contrastive divergence training algorithm formally seeks to approximate the gradients of this objective function. As such the value of the training objective function is a more natural metric of comparison for our purposes than classification accuracy for an appropriate data set.

The training set consists of four distinct functions:

$$
[x_1]_j = \begin{cases} 
1 & j = 1, \ldots, \lfloor n_v/2 \rfloor \\
0 & \text{otherwise} 
\end{cases}
$$

and their bitwise negations. In order to make the data set more challenging to learn, we add 10% Bernoulli noise to each of the training examples. One hundred training examples were used in each instance with a learning rate of 0.01 and $10^5$ epochs per reinitialization. We take the model to be an RBM consisting of 8 visible units and 10 hidden units. Finally, rather than reinitialising a fixed number of weights at each iteration we update each weight with a fixed probability. This formally differs from previous approaches, but performs similarly to reinitialising a constant fraction. Upon initialisation, each variable is drawn from a zero-mean Gaussian distribution.
We see in Figure 10.6 that partial reinitialization accelerates the training process of contrastive divergence, with an optimal probability of reinitialising a weight or bias in the RBM of roughly 10%. In particular, for the 10%-strategy it only takes 58 reinitialisations on average to obtain the same training objective function as 1000 full reinitialisations.

Strong regularisation is expected to lead to a much simpler optimisation landscape since as $\lambda \to \infty$, the problem becomes convex. We find that for $\lambda = 0.01$, the difference between the partial and full reinitialisation strategies becomes on the order of 0.01%. Although reinitialising only 10% of the variables continues to be the superior method, these small differences could also arise solely from the stochastic nature of contrastive divergence training and the fact that it takes fewer epochs for the partial reinitialization to approach a local optimum.

It is evident from these benchmarks that partial reinitialization can substantially reduce the complexity of training Boltzmann machines. We expect this approach to show an even more significant advantage for large task-based problems such as MNIST digit classification. Similarly, it should be possible to leverage these improvements in other forms of neural net training, such as feed-forward convolutional neural nets.

10.3 Conclusion

We introduced a general purpose approach, termed partial reinitialisation, to significantly improve the performance of an optimiser. We numerically explored comparisons to state of the art algorithms on a range of optimisation problems in machine learning, including clustering, training hidden Markov models and training Boltzmann machines. Although we only used the most basic version of our algorithm, with a single additional level in the hierarchy and picking subsets of variables at random, the advantage over the standard full reinitialisation on all our benchmark problems is substantial. We expect a hierarchy with multiple levels and with subsets picked according to more advanced problem-specific heuristics to lead to even further improvements.
Part IV
Scientific computing framework
Chapter 11

Introduction

A website for this project should be up in the near future. As of now, the code is not publicly available.

Compute power is growing at an exponential rate and it is becoming increasingly more complex to utilise it efficiently, particularly for scientific computing. This leads to large amounts of resources wasted on technical tasks, including human time, compute power, storage space, network bandwidth and data backups. To address this issue we introduce Project Whiplash, a high-performance, distributed framework for maximally efficient scientific computing. The goal is to make working on a project with a collaboration of people across the world as easy a writing a program for a single process on a laptop. Particular emphasis is put on usability, security, portability, fault tolerance, collaboration and performance.
Chapter 12

Project Whiplash

Compute power is growing at an exponential rate and different new forms of computing are emerging, including cloud computing, quantum computing, cold logic, and other specialised hardware. Increasingly more powerful and exotic hardware often leads to a steeper learning curve to use it efficiently.

In the computational science community, a significant amount of time is currently wasted on recovering and recomputing results, learning submission and queuing systems, doing backups, enforcing data formats, and other technical tasks.

Indeed, there are tools available for high-performance scientific computing applications which address some of these problems, including Vistrails for data provenance and visualisation [220], Tensor Flow for machine learning [221] and Microsoft HPC Manager [222], which is a convenient job-submission system for Windows clusters. None of these tools have, however, become a de facto standard in the computational science community due to bad design, proprietary license, special purpose, out of date, and other reasons.

Project Whiplash is a collaborative, distributed framework which makes using a cluster not any harder than running a script on a laptop. Aimed at computational scientists, the goal of Project Whiplash is to maximise the time spent on actual research and to become an industry standard for general purpose, high-performance scientific computing.

12.1 Overview

Project Whiplash is composed of a container registry for storing executables, a database for storing data, schedulers for allocating compute resources, an API server for handling queries and a web interface for account settings and
to view job statistics (see Figure 12.1). In Project Whiplash an executable is treated a function with a given input and output. It can hence be called from within another executable as a sub-routine and used as a building-block for more complex programs. Each executable is a Docker [223] container which reads an input file and writes the results into an output file. The container stores both the executable and all the necessary dependencies and sources required to build it. The user wraps the executable into a container on their machine, pushes the container to our registry and adds a descriptor of the executable to our database.

To run a given executable with given inputs and fetch certain results, the user submits a query which contains filters for the executable and input parameters. If a similar query has already been performed, only the difference is actually computed and the rest is fetched from the database directly. For each task, the scheduler fetches the required input from the database, creates the input file, runs the container with that file, reads the output file and commits the result back to the database.

The framework focuses on several aspects of high-performance computing:

### 12.1.1 Usability

Usability is perhaps the most important aspect of any service. The interface is the first layer of interaction with the user and if it is not easy to use or requires a manual, it immensely brings down the value of the whole product. An effective front-end also significantly increases the productivity and general
satisfaction of the user. Project Whiplash has an intuitive and powerful REST API which can either be used directly or through Python bindings.

12.1.2 Security

In a distributed framework, especially if it contains sensitive data, security is an important issue. The database is running with an authentication protocol and is accessible for users through the API server. The API server can be accessed with a token which is generated and periodically renewed for each registered user. By default, each user can access only their own documents and executables.

The containerised runtime environment of each executable uses only core features of a Linux kernel, which provides complete isolation from the host machine and circumvents the evident security issue of natively running binaries written by any registered user.

12.1.3 Portability

Any executable generally has a range of dependencies, including the build environment, compiler and libraries. It can compile and run on a given machine at a given point in time. However, even on the same machine, the build environment constantly changes, compilers are updated and libraries are replaced. This results in a constant need to maintain a project even within a given build environment. More importantly, the output of an executable could depend on the runtime environment and data could even become irreproducible if details of the environment are not noted.

In Project Whiplash all these problems are solved with Linux containers, which are machine agnostic and unlike virtual machines, do not have any performance penalty over running an executable natively. Containers are also lightweight, with only a small memory overhead of a few tens of megabytes on top of the requirements of the process and can be removed and spawned up again in seconds [224].

12.1.4 Fault tolerance

Backups are the backbone of any serious computing project and the only way to ensure effective backups is by entirely automating the process. In Project Whiplash backups are done on two levels. Firstly, the database is replicated multiple times, with replicas stored in different locations. If any replica stops working, it does not affect the integrity of the database in general and can be
inserted back when the problem is resolved. This does incur a small performance penalty as any insertion or deletion from the database is performed on each replica individually. It also requires significantly more hardware resources. However, for big-data applications with a high-performance, distributed system, hardware faults are inevitable and full replication is the only way to guarantee persistent up-time. Second level backups consist of regular dumps of both the database and container registry to archive storage. As this is more a long-term, low-performance solution, data is first compressed into binary dumps before being stored.

12.1.5 Collaboration

Particularly in science, collaboration is very important for rapid progress. In Project Whiplash collaborations can be formed which give users in a collaboration read access to all data and executables which belong to it. This way collaborative programming can be done very efficiently, with each group working on a sub-routine of a larger project.
12.1.6 Memoisation

Memoisation is the concept of caching the results of function/input pairs. A standard function takes an input and computes an output every time it is called (see Figure 12.2a). When a memoised function is called with a given input, the cache is first checked if it was called with the same input before. If it has, the result is retrieved directly from the cache and if not, it is computed, stored and then returned (see Figure 12.2b). Memoisation is the basis of dynamic programming, which is a key technique for efficiently solving combinatorial optimisation problems and provides massive reductions in complexity over a brute force approach. Perhaps the most evident example for the advantage of memoisation is for computing Fibonacci numbers recursively. The standard function is the following:

```cpp
def fibonacci(const int n):
    if (n == 1 || n == 2):
        return 1;
    else:
        return fibonacci(n-1) + fibonacci(n-2);
```

which requires $O(2^n)$ operations for the $n^{th}$ Fibonacci number. With memoisation this can be done in linear time as follows

```cpp
def fibonacci_memo(const int n):
    if (n == 1 || n == 2):
        return 1;
    else:
        static std::unordered_map<int, int> memo;
        if (memo.find(n) == memo.end()):
            memo[n] = fibonacci_memo(n-1) + fibonacci_memo(n-2);
        return memo[n];
```

where a hash-map is used as a cache.

In the scenario of a large collaboration working on a project, data and executables are often reused multiple times. Manual book-keeping of which executables have been ran with which input is tedious and with large projects simply not possible. On the other hand, recomputing the same results multiple times is a huge waste of compute power and time.

Project Whiplash introduces the concept of ”distributed memoisation”. Instead of a memory, we use a database; instead of the if-loop checking we use an API server and instead of a standard function, we use Linux containers (see Figure 12.2c). This way, when a given executable is called with
a given input, the framework operates on that input once and only once. All subsequent times that executable is called with the same input, the result is retrieved from the database rather than recomputed. This process is automated by storing the object signatures of all data and executables.

12.1.7 MapReduce

When statistics are computed on data distributed across multiple nodes, it is more efficient to do it with parallel reduction operations rather than redirecting all effort onto a single process. The standard way of doing this in a database is MapReduce [225], which consists of a Map operation for filtering and sorting and Reduce for performing the actual reduction. The time-complexity in the number of nodes is this way improved from linear to logarithmic. Also, if memory is an issue and multiple pieces of data do not fit onto a single node, such a reduction may be the only way out. The Python interface in Project Whiplash accommodates MapReduce with arbitrary mapping, reduction and post-processing functions, which allows for efficient manipulation of data on par with specialised BigData tools, like Hadoop [226] and Spark [227].

12.1.8 Scheduling

To optimise for the granularity of batch systems and minimise the amount of time spent in the queue, in Project Whiplash scheduling is done at multiple levels, including a local scheduler within a node, a distributed scheduler on a machine and a global scheduler across multiple machines.

12.1.9 Query optimisation

Both the client and scheduler communicate with the database through the API server. Particularly for small amounts of data, the overhead of creating a request can be significant and constitutes additional load on the server. To send maximum amount of data with as few requests as possible, the Python interface of Project Whiplash batches multiple requests together into a single query, which is further compressed on the client and decompressed on the server.

However, even with optimisation for the number of queries, the load on the API server and the database can be significant, particularly with a very large number of running jobs. If this load exceeds the thresholds the deployment can handle, it will result in crashes and increased downtime. To
keep those thresholds as high as possible, we try to make maximum use of available hardware in a number of ways.

Large collections in the database are sharded and each shard runs on its own node. This way, documents are split across multiple nodes and a query which only addresses documents located on one node can be served without requiring any resources from other nodes. If the splitting is done uniformly with respect to the common query type, the hardware support of the database effectively increases proportional to the number of shards. To further reduce congestion and accommodate more simultaneous queries, we use multiple routers on different ports. The API server also operates as a cluster of workers which can serve queries in parallel. To minimise latency, the schedulers, database and the API server are all deployed close to the compute environment.

12.1.10 Database

For maximum horizontal scaling our database is document-based NoSQL and we use MongoDB [228] as it is one of the most battle-tested databases available. However, Project Whiplash is built with an emphasis on modularity and different components can be replaced with minimal additional coding effort.

In Project Whiplash data is stored as JSON-documents [229], which is an efficient, human-readable dictionary-based format and is rapidly becoming an industry standard with a wide range of libraries and tools. To accommodate for memory limits, large documents are stored directly in binary, split into multiple fixed-size chunks and controlled by a metafile. This is, however, significantly slower than the native JSON format and currently makes Project Whiplash less suited for working with large data objects.

12.2 Conclusion

We introduced a high-performance framework for scientific computing, termed Project Whiplash, which allows for efficient and maximally convenient utilisation of compute resources. It has a range of important technical features, including heavy use of Linux containers with a private container registry; distributed, sharded and replicated document-based NoSQL database; automatic distributed memoisation; efficient indexation of data; MapReduce; a powerful and easy-to-use API; multi-level schedulers; possibility for collaboration; efficient utilisation of network bandwidth; regular backups to multiple locations and many others.
Project Whiplash has to date mostly been used within our group for spin-glass physics. This task is an ideal use-case, as it requires extensive statistics performed on the results and mostly consists of a large number of relatively short, independent jobs with small input and output files. Doing this manually with conventional file and job-submission systems would effectively not be possible.

The first priority for future work, apart from further improving stability, performance and scalability of the framework, is opening Project Whiplash up to a wider audience. In particular, that involves working with computational science groups and companies around the world who make heavy use of compute resources, as well as establishing partnerships with providers of cloud and supercomputing systems to field-test Project Whiplash as a next generation batch system.
Bibliography


[184] Spin Glass Server http://www.informatik.uni-koeln.de/spinglass/, http://www.informatik.uni-koeln.de/spinglass/


List of publications


LIST OF PUBLICATIONS
List of preprints


List of patents


List of talks


[243] Domain wall topologies in weakly repulsive harmonically confined fermi gases, 2012. QSIT Seminar, ETH Zürich, Switzerland.

[244] Average case complexity of low-dimensional ising spin glasses, 2014. APS March Meeting, Denver, CO, United States.


Afterword

The reason this thesis is titled ‘In search of optimality’ is because it is mainly about optimisation. Optimisation, as in, algorithms and computing and also because throughout the last four years I have myself been looking for what to do next. This is also the reason for the four disconnected parts.

The main three questions I have started asking myself with whatever I do are ‘Do I enjoy doing it?’; ‘Am I learning something?’; ‘How will it change the world?’. If one doesn’t enjoy it, one will quit. If one doesn’t learn something, one doesn’t grow. If one doesn’t see how it will change, or at least, positively impact the world, then doing it is worthless. Sometimes the answers are not immediately obvious, especially with little experience, but I believe it’s important to get them as quickly as possible and move on.
Acknowledgements

Above all I would like thank Matthias Troyer for letting me work in this group and most importantly for giving me the freedom to explore, discover, fail and succeed. I would neither have reached this point without Helmut Katzgraber, who convinced me to write up this thesis and gave me lots of advice.

I am very grateful to many others. Lei Wang, Damian Steiger and Thomas Häner for keeping it real in the office. Sebastiano Pilati for numerous discussions back in the days of dft. Juan Osorio for introducing me to climbing and creating the group meetings. Michele Dolfi for organising more things than I can remember. Jan Gukelberger for contributing the \LaTeX-foundations of this thesis. Andreas Hehn for making sure the coffee shipments are always on time. Iztok Pizorn for very fruitful discussions on optimisation and other things. Jakub Imriška for our ponderings about algorithms and for being an awesome person. Alexey Soluyanov for interesting arguments on the boundary of things I know nothing about and lots of advice. Troels Rønnow for officially opening the whisky club with the first bottle and the many arguments we had. Sergei Isakov for advice on how to write seriously fast code. Alex Kosenkov for the crazy brainstorm sessions. Ethan Brown for being the co-founder. Krysta Svore for inviting me to intern at QuArC, it was awesome. Matt Hastings for amazing ideas, help and persistence. Nathan Wiebe for resurrecting my knowledge of quantum mechanics and being my mentor at Microsoft. Dave Wecker for lots of help and guidance and, of course, the barbecue. Tameem Albash for fruitful collaboration and hosting me at USC. Giuseppe Carleo for exciting discussions and translating the abstract of this thesis to the language of his people.

Thanks to the past and present members of the G-floor for contributing to a stimulating environment. A big credit to the IT support teams at the Physics Department at ETH Zurich and CSCS who made most of the work here possible.
Thanks to my family and friends for their love, patience, confidence and support.