

# Applications of variational methods for quantum computers

Manpreet Singh Jattana

IAS Series

Band / Volume 53

ISBN 978-3-95806-700-4





Forschungszentrum Jülich GmbH  
Institute for Advanced Simulation (IAS)  
Jülich Supercomputing Centre (JSC)

# **Applications of variational methods for quantum computers**

Manpreet Singh Jattana

Schriften des Forschungszentrums Jülich  
IAS Series

Band / Volume 53

---

ISSN 1868-8489

ISBN 978-3-95806-700-4

Bibliografische Information der Deutschen Nationalbibliothek.  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der  
Deutschen Nationalbibliografie; detaillierte Bibliografische Daten  
sind im Internet über <http://dnb.d-nb.de> abrufbar.

Herausgeber  
und Vertrieb:           Forschungszentrum Jülich GmbH  
                                  Zentralbibliothek, Verlag  
                                  52425 Jülich  
                                  Tel.: +49 2461 61-5368  
                                  Fax: +49 2461 61-6103  
                                  zb-publikation@fz-juelich.de  
                                  www.fz-juelich.de/zb

Umschlaggestaltung:   Grafische Medien, Forschungszentrum Jülich GmbH

Titelbild:                ©QC Creations – stock.adobe.com (modifiziert)

Druck:                    Grafische Medien, Forschungszentrum Jülich GmbH

Copyright:              Forschungszentrum Jülich 2023

Schriften des Forschungszentrums Jülich  
IAS Series, Band / Volume 53

D 82 (Diss. RWTH Aachen University, 2022)

ISSN 1868-8489  
ISBN 978-3-95806-700-4

Vollständig frei verfügbar über das Publikationsportal des Forschungszentrums Jülich (JuSER)  
unter [www.fz-juelich.de/zb/openaccess](http://www.fz-juelich.de/zb/openaccess).



This is an Open Access publication distributed under the terms of the [Creative Commons Attribution License 4.0](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

# Abstract

The primary subject of this dissertation is the analysis and improvement of variational methods that combine the use of classical and gate based quantum computers. The secondary subject is the development of matrix based error mitigation and benchmarking protocols for noisy quantum computers.

Variational methods run on quantum computer emulators are used to find the ground state energies of the Heisenberg and Hubbard models and selected molecules in chemistry. An algorithm is developed and deployed to automate the creation of variational circuits. The theory and overview of variational methods and gradient based optimisation algorithms are presented. We learn that while variational methods make it possible to use current generation quantum computers, guarantees of always finding the ground state energy are elusive. We introduce noise in our emulations and adapt the optimisation algorithms to withstand it. We observe the emergence of local minima and barren plateaus which hinder variational methods from finding the ground state energies. It is discerned that clever choices of initial states and parameters are necessary ingredients for success. We develop the technique of quasi-dynamical evolution inspired by quantum annealing. It overcomes the limitations of standard variational algorithms by systematically improving the ground state energy estimate. Our tests show that the heuristic improves the energy estimate even in facile settings.

We introduce seven criteria for ideal error mitigation protocols. A new protocol is developed on its basis. Our tests on IBM Q quantum computers show noticeable error mitigation. The matrix generated during the execution of the protocol helps detect and visualise errors and biases. We invent and use small depth quantum circuits for benchmarking quantum computers.



# Zusammenfassung

Das Hauptthema dieser Dissertation ist die Analyse und Verbesserung von Variationsmethoden, die den Einsatz von klassischen und gatterbasierten Quantencomputern kombinieren. Das sekundäre Thema ist die Entwicklung von matrixbasierte Fehlermitigation und Benchmarking-Protokollen für verrauschte Quantencomputer.

Die Variationsmethoden werden auf Quantencomputeremulatoren implementiert. Hierbei beschränkt sich der Anwendungsbereich auf die Berechnung der Grundzustandsenergien des Heisenberg-Modells, des Hubbard-Modells und ausgewählter Moleküle. Es wird ein Algorithmus entwickelt und eingesetzt, um die Erstellung von Variationsschaltungen zu automatisieren. Eine Zusammenfassung von Variationsmethoden und gradientenbasierten Optimierungsalgorithmen wird bereitgestellt. Wir finden, dass Variationsmethoden den Einsatz von gatterbasierten Quantencomputern der aktuellen Generation ermöglichen. Natürlich besteht keine Garantie, dass der Grundzustand tatsächlich gefunden wird. Wir führen Rauschen in unsere Emulationen ein und passen die Optimierungsalgorithmen an, um diesem Rauschen zu widerstehen. Wir können das Auftreten von lokalen Minima und kargen Plateaus feststellen, die die Variationsmethoden daran hindern, die Grundzustandsenergien zu finden. Eine geschickte Wahl des Variationsansatzes und der Variationsparameter ist unabdingbar für den Erfolg. Wir entwickeln die Technik der quasi-dynamischen Evolution, die von der Quantenannealing inspiriert ist. Sie überwindet die Beschränkungen von Standard-Variationsalgorithmen, indem sie die Schätzung der Grundzustandsenergie systematisch verbessert. Unsere Tests zeigen, dass die Heuristik die Energieschätzung sogar in einfachen Situationen verbessert.

Des Weiteren stellen wir sieben Kriterien für idealen Fehlermitigation-Protokollen vor. Ein neues Protokoll wird auf dessen Grundlage entwickelt. Diese Strategie wird auf IBM Q Quantencomputern getestet. Hierbei lässt sich eine erkennbare Fehlermitigation feststellen. Die während der Ausführung des Protokolls erzeugte Matrix hilft, Fehler und Verzerrungen zu erkennen und zu visualisieren. Wir erfinden und verwenden Quantenschaltkreise mit geringer Tiefe für das Benchmarking von Quantencomputern.



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Zusammenfassung</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Gate-based quantum computing</b>	<b>7</b>
2.1 Quantum digits . . . . .	7
2.1.1 One qubit . . . . .	7
2.1.2 Multiple qubits . . . . .	8
2.2 Quantum gates . . . . .	10
2.2.1 Single qubit gates . . . . .	11
2.2.2 Multiple qubit gates . . . . .	12
2.3 Quantum circuits . . . . .	13
2.4 Quantum computer emulator . . . . .	14
2.5 Quantum algorithms . . . . .	16
2.5.1 Phase estimation algorithm . . . . .	17
2.5.2 Drawbacks of phase estimation . . . . .	19
2.5.3 Motivation for variational algorithms . . . . .	19
<b>3 Theory of variational methods</b>	<b>21</b>
3.1 Introduction . . . . .	21
3.2 Theorem of variation . . . . .	21
3.3 Variational methods . . . . .	25
3.3.1 Automating circuit preparation . . . . .	30
3.3.2 Problem Hamiltonian . . . . .	33
3.3.3 Initial state . . . . .	35
3.3.4 Classical calculations . . . . .	35
3.3.5 Error resilience . . . . .	37
3.4 Classical optimisation . . . . .	38
3.4.1 Mathematical foundations . . . . .	38
3.4.2 Gradient based algorithms . . . . .	41
3.4.3 Calculating gradients . . . . .	43
3.4.4 Summary . . . . .	45
<b>4 Applications</b>	<b>47</b>
4.1 Heisenberg model . . . . .	47
4.1.1 Introduction . . . . .	47
4.1.2 Finding an ansatz . . . . .	51
4.1.3 The XY ansatz . . . . .	56
4.1.4 Initial state . . . . .	56
4.1.5 Results - Part I . . . . .	57

4.1.6	Results - Part II . . . . .	67
4.1.7	Discussion . . . . .	71
4.2	Quantum chemistry . . . . .	71
4.2.1	Introduction . . . . .	71
4.2.2	Water molecule . . . . .	74
4.2.3	Dissociation of Li-Li . . . . .	77
4.3	Hubbard model . . . . .	78
4.3.1	Introduction . . . . .	78
4.3.2	Ansatz . . . . .	80
4.3.3	Initial state . . . . .	81
4.3.4	Results . . . . .	82
4.3.5	Discussion . . . . .	83
4.4	Conclusion . . . . .	83
<b>5</b>	<b>Quasi dynamical evolution</b>	<b>85</b>
5.1	Introduction . . . . .	85
5.2	Problems in variational methods . . . . .	85
5.2.1	Local minima . . . . .	85
5.2.2	Barren plateaus . . . . .	87
5.3	Evolution . . . . .	88
5.3.1	The quasi dynamics . . . . .	89
5.3.2	Demonstration . . . . .	89
5.3.3	Connection to annealing . . . . .	91
5.4	Results . . . . .	91
5.4.1	Heisenberg model . . . . .	91
5.4.2	Random Hamiltonians . . . . .	92
5.4.3	Large lattices . . . . .	93
5.5	Discussion . . . . .	94
5.6	Conclusion . . . . .	97
<b>6</b>	<b>Error mitigation and benchmarking</b>	<b>99</b>
6.1	Introduction . . . . .	99
6.2	Mitigation criteria . . . . .	100
6.3	Standard protocol . . . . .	100
6.4	Improved protocol . . . . .	102
6.5	Experimental demonstration . . . . .	103
6.5.1	Improved protocol . . . . .	105
6.5.2	Standard protocol . . . . .	105
6.6	Benchmarking . . . . .	107
6.6.1	Detecting bias . . . . .	109
6.6.2	Mean field model . . . . .	111
6.7	Scalable mitigation . . . . .	113
6.7.1	Subspace reduction . . . . .	113
6.7.2	Demonstration . . . . .	115
6.7.3	Discussion . . . . .	117
6.8	Summary . . . . .	118
<b>7</b>	<b>Conclusion and outlook</b>	<b>119</b>
<b>A</b>	<b>Sample files</b>	<b>123</b>

<b>B Equivalence to parameter shift rule</b>	<b>125</b>
<b>Bibliography</b>	<b>127</b>
<b>Eidesstattliche Erklärung</b>	<b>157</b>
<b>Acknowledgements</b>	<b>159</b>



## Chapter 1

# Introduction

Humans compute. They employ tools for assistance. The earliest of them, *notched bones*, were used at least twenty thousand years ago to store numbers [D'E+12]. The last dozen centuries have seen an increase in computing tools, including the abacus, the machines of Schickhard, Pascal, Leibniz, Odhner, Baldwin, Hollerith, Kelvin, Bush, up to the Babbage's conception of an analytical engine in the mid-19th century [IH01; Bab37]. Afterwards, in the last century, there has been a rapid development of *computers* [Lud09; Zus36; Tur37; Ran82]. One essential characteristic of computers is *scalability*, that is, the ability that the computational power can be increased in proportion to the needs of the humans by reasonable adjustments. However, the highest computational needs of humans continue to surpass any computer built to date.

Humans describe. They employ physical laws, theories, and models for assistance. More than two thousand years ago, Archimedes indulged in discovering physical principles [Arc09]. Some progress has been made since then. In the last century, humans have developed quantum theory, which has vast descriptive authority [Lam92]. Peculiarly, quantum theory's descriptive power is statistical [Bal14]. The theory's framework is derived from a few axioms [VBG55]. Given that the theory predicts probabilities of plausible events, at its core it is based on probability theory [Jay03]. Despite the progress in the description of nature, the equations of quantum theory are not easily solvable in general.

Humans compute to describe. Universal digital computers or classical computers have become omnipresent in daily life. From transistors to supercomputers, computers assist humans in many tasks. A vast amount of research today is assisted by computers and can be expected to continue in the future. The computational demands of the research community exceed the supply. In particular, it is believed that digital computers are inefficient in simulating general quantum systems [Pre21]. It has been envisioned that quantum computers may prove helpful in simulating physical phenomena or solving specific problems more efficiently [Fey82; Ben82]. The time is ripe for identifying more such problems and testing them.

### Quantum computing

The core idea of quantum computing is to use efficient and scalable computing machines to assist in the description of nature. If other problems exist that fit such a description, then they may also be solved efficiently on a quantum computer. The realisation of such a concept is envisioned to require a computing machine that works on the principles of quantum theory.

Two types of quantum computers are currently under research and development: annealers and gate-based systems [NM19]. The first approach to quantum computing is termed quantum annealing [Fin+94; KN98; Far+00; Har+10; Joh+11]. Quantum annealing is primarily used

to solve specific types of optimisation problems [Wil+20a; PO+19; Ohz20; Hsu+19]. Recently, novel applications of annealers such as the realization of spin ice [Kin+21; Kin+22a; Kin+22b], studying phase transitions [Har+18], observing a topological phenomenon [Kin+18], studying the two dimensional Ising model [Wei+20], and machine learning [PL13; Wil+20], among others [AS21; Cha19], have been found.

The other type of quantum computing is gate-based. As the name suggests, gate-based quantum computers rely on logic operations or gates to manipulate the stored information. Such computers are considered universal because any arbitrary operation can be decomposed as a sequence of unitary gates [Bar+95; DiV95; DBE95]. In this work, we focus on gate-based quantum computers. Henceforth, the mention of a quantum computer shall imply a gate-based one unless otherwise specified.

There are parallels between classical and quantum computing. The classical computer's elementary storage and manipulation unit is one bit consisting of two states. Due to the *scalability* of semiconductor devices, they are highly preferred for the physical implementation of the classical bit. A bistable multivibrator [RSR10], commonly known as the flip-flop, can store one bit of information, either 0 or 1, corresponding to each of its stable states. Similarly, the quantum computer's elementary storage and manipulation unit is also a two-level system commonly regarded as a quantum bit or qubit. Several physical implementations of the qubit are possible, e.g. nuclear spin, electron spin, trapped ions, quantum dots, superconducting transmons [NC10; LD98; Koc+07]. Which implementation is the most suitable remains part of active research. The essential difference between the two types of computing platforms is the following. The best-known theoretical description of classical computers states that the bit stays in either one of the two possible states at all times during the computation. In contrast, quantum theory tells us that the quantum computer is not necessarily in either of the two states during the computation. Only the measurement produces either one of the two states.

Classical computers manipulate their bits by relying on logic gate operations. Common operations include the **NOT** gate, which takes the input 0 or 1 and changes it to 1 or 0, respectively. To cover all possible Boolean operations, one can use multiple **NAND** gates. The **NAND** gate gives the output 0 when both inputs are 1 and 1 otherwise. Since all Boolean operations can be expressed in combinations of **NAND** gates, it is referred to as one of the universal gates [Mai07]. Quantum computers manipulate their qubits relying on reversible logic operations, in contrast to their classical counterparts. Universal gates also exist for quantum computers [SW95; DiV95].

We have come a long way from the first abstract model of a quantum computer [Ben80]. The 21st century has seen a rapid development of experimental devices physically implementing the quantum computer. It has been shown that a quantum computer can perform a task, albeit without a meaningful task it was originally envisioned, faster than a classical computer [Aru+19]. The ultimate purpose is to leverage the framework of quantum theory to gain a meaningful computational advantage over classical computers [EJ96]. It is not envisioned that quantum computers will replace all classical computers [JM19]. The advantage is theoretically predicted for several areas, among them (1) algorithms to find the prime factors of integers [Sho94; Sho99] or for database search [Gro96]; (2) efficient simulation of quantum mechanical systems [Fey82]. In this dissertation, we devote our attention to the latter.

## Motivations

The properties of any isolated physical system that is described by quantum mechanics can be calculated by solving Schrödinger's equation. It describes the dynamical evolution of the

systems on the atomic scale. The time-dependent form is given by

$$i\hbar \frac{\partial}{\partial t} |\Psi(t)\rangle = H(t) |\Psi(t)\rangle, \quad (1.1)$$

where  $H(t)$  is a time-dependent Hamiltonian describing the system of interest, and  $|\Psi(t)\rangle$  describes the state of the entire system at a given time  $t$ , which according to quantum mechanics contains all the information that can be known about the system. We briefly motivate systems that rely on solving Eq. (1.1) to obtain these properties.

**Heisenberg model:** The phenomenon of magnetism has perplexed humanity since the discovery of the lodestone. The multitude of superstitions arising after this discovery were first questioned by William Gilbert in the year 1600. An interesting account of the history of magnetism is found in [Ver96; Mat06]. His work initiated the first arguments to bring about a description of the phenomenon [Gil60; MB00]. More than four centuries later, the current understanding of magnetism is as follows.

The description of magnetism in a material is given in terms of a *magnetic order*, which occurs from the presence of atoms with *magnetic moments* and their *interactions*. The two broad types of interactions which can be important in allowing the magnetic moments in a material to communicate with each other and produce magnetic order are magnetic dipole interaction and exchange interaction. We are led to the Heisenberg model by focusing on the exchange interaction. The model is analytically solvable only for the case of a one-dimensional lattice using the Bethe ansatz [Bet31]. Since the spin magnetic moments are inherently quantum mechanical in nature, their mapping to the qubits of a quantum computer is straightforward. After the mapping, the properties of the state of a quantum computer then encode the corresponding properties of the state of the Heisenberg model.

**Hubbard model:** For more than half of the twentieth century, collective electron theory was the dominant theory explaining ferromagnetism in metals [Woh53; Sto47]. Despite its success, it failed to accurately describe the itinerant ferromagnetism of transition metals, such as iron and nickel. Hubbard [Hub63; Hub64a; Hub64b], and independently Kanamori [Kan63] and Gutzwiller [Gut63], proposed a model which is now named after Hubbard that arose to fill in the gap. The model was initially applied to describe the behaviour of the transition-metal monoxides, e.g. FeO and other antiferromagnetic insulators, which had been predicted to be metallic by methods which do not take into account strong electron interactions [Sca16].

Since the Hubbard model describes fermionic systems, the mapping to a quantum computer is not straightforward and requires suitable transformations. The transformed Hamiltonians of the Hubbard model can then be mapped to the quantum computer. Interesting properties of the Hubbard model, e.g. the ground state energy, have a direct correspondence to the properties of the quantum computer. In essence, if the transformed Hamiltonian can be brought into the ground state on a quantum computer, we have also found the ground state of the Hubbard model. This would be a significant step in understanding and predicting the numerous quantum properties of materials.

**Quantum chemistry:** The goal is to calculate the ground state energy for molecular systems described by Eq. (1.1). Given the combinations of atoms to create all kinds of relevant molecules, chemistry applications can range from prototypes to those well beyond the capabilities of any supercomputer. For quantum advantage, such applications need to be tailor-fitted so that they are not irrelevant due to (a) unavailability of experimental results,

(b) availability of efficient classical computations, (c) complexity, or (d) industrial inapplicability [Elf+20].

In the above examples, the governing idea is to solve Eq. (1.1). The  $\Psi(t)$  in Eq. (1.1) is a complex-valued function. To process  $\Psi(t)$  in general on a classical computer, the first step is to represent  $\Psi(t)$  using floating-point arithmetic on a combination of flip-flops. A  $2^N$  dimensional array of floating-point numbers is required to represent all the information contained in  $\Psi(t)$  at a particular time  $t$  and for  $N$  qubits. Since this representation doubles with every new qubit added, classical computers quickly run out of storage space. On the other hand, the representation of  $\Psi(t)$  on a quantum computer is more natural or efficient. Therefore, the motivation is to use such computers to solve this type of problems.

While it is clear that such an ideal computer can solve the problems mentioned above, there is no guarantee that such a computer can ever be built. A fully working, large-scale, error-free or error-corrected, ideal quantum computer is neither currently available nor foreseen in the near future. All current quantum computers belong to the *noisy intermediate scale quantum* (NISQ) regime [Pre18; Pre21]. The motivations of current research in applications of quantum computing are as follows.

1. To develop methods that can use quantum computers despite (a) the small number of gate operations possible on them and (b) the presence of noise.
2. To investigate if, and under what conditions, quantum advantage is possible through such methods on such computers.
3. To develop techniques that help mitigate noise.

## Objectives and methods

In line with the above motivations, we investigate the following things. We also mention the methods to be used to investigate.

- I. Hybrid variational methods for quantum computers, which are based on the variational principle, have recently been used to leverage the resources of small-scale prototype quantum computers. The variational principle allows any *ansatz* to be used. However, not every *ansatz* will solve the problem. Therefore, it is vital to develop ways to create *ansätze* that are problem-specific and test them on their quality to find the ground state energy. By mimicking and using an ideal quantum computer through an emulator, we will solve for the ground state energy of the problems and compare them against the true values found using classical methods. To analyse the effects of noise, we will add artificial noise to the emulator while solving for the ground state energy.
- II. One aim is to identify and illustrate deficiencies of hybrid variational algorithms. We remedy the deficiencies by proposing, implementing, and testing a heuristic that is able to improve the results of standard variational algorithms. This heuristic quasi-evolves the given system to its ground state in a systematic way. We observe that the heuristic avoids problems common to variational algorithms. Furthermore, we time the emulator against a realistic but hypothetical quantum computer to ascertain which will solve a given task faster.
- III. We aim to develop error mitigation and benchmarking methods for current and near future quantum computers. We will use actual noisy quantum hardware to perform error mitigation and simple benchmarking experiments.

## Outline

The contents are structured as follows. In chapter 2, we introduce the gate-based model of quantum computing. Starting from the elementary qubit, which is manipulated using gates, which combine to form circuits, we briefly describe a quantum algorithm.

In chapter 3, we introduce the hybrid variational methods for quantum computers. By introducing the variational principle using two simple examples, we illustrate how the circuit preparation can be automated and the ingredients necessary to effectively use such methods. We then briefly introduce the basic classical optimisation theory relevant to such methods.

In chapter 4, the variational algorithms are used to find the ground state energy of three problems: the Heisenberg model, the Hubbard model, and some molecules in quantum chemistry. A quantum computer emulator is used for this purpose.

In chapter 5, we first numerically demonstrate problems facing variational methods, then propose a heuristic called *quasi-dynamics* that alleviates some of these problems. We show a connection between the heuristic and quantum annealing and numerically demonstrate its working on selected problems.

In chapter 6, we introduce criteria for an ideal error mitigation protocol. Based on these, we develop a matrix-based mitigation protocol that improves previous protocols. We test this protocol and a benchmarking circuit on real quantum hardware.

Chapter 7 contains the main conclusions of this work. We also offer an outlook for future work.



## Chapter 2

# Gate-based quantum computing

## 2.1 Quantum digits

To understand the working of a quantum computer, it is essential to start from its most elementary unit. In this section, we review the qubits. We look at how the qubit is different from a bit, and what differences highlight their basic information storage capabilities.

### 2.1.1 One qubit

A classical computer's elementary information storage unit is a binary digit or bit. The binary digit takes precisely two values, 0 or 1. Theoretically, it does not exist in any state other than these two. Physical implementations ensure that the state of such a computer stays in either one of the states unless it is required to change. Naturally, a useful bit is the one which can be changed from one state to another as per need. When the state is changed, the time taken is negligibly small. During this negligibly small time the system exists in a classical superposition of the two states. From this argumentation, it is clear that classical digital computers are only an abstract construct. In reality, computers are always analogue in the strict sense, but with a good enough physical implementation, they can be reasonably well approximated as digital computers.

Similarly, the elementary storage unit of a quantum computer is a quantum binary digit, or qubit. Since the state is a variable of quantum theory, it is represented by  $|\Psi\rangle$ . The state of the two-level system is described by

$$|\Psi\rangle = c_0|0\rangle + c_1|1\rangle, \quad (2.1)$$

where  $|0\rangle$  and  $|1\rangle$  represent the computational basis states corresponding to the binary states 0 and 1, respectively. The variables  $c_0$  and  $c_1$  are complex numbers in general and  $c_0^2 + c_1^2 = 1$ . By requiring a restriction condition  $c_0 \neq c_1$  and  $(c_0, c_1) \in \{0, 1\}$  for Eq. (2.1), we get the corresponding classical bit. Thus, classical computers can be considered to be a special case of quantum computers and we can expect quantum computers to have a greater computational power [DiV00]. In general, a qubit described by Eq. (2.1) can take any linear combination of values of the two complex variables. This property in the description of the theory of quantum mechanics is termed as *quantum superposition* of states.

A striking difference between classical and quantum computers arises when considering the read-out of the final state after performing a certain computation. The binary digit does not need any special arrangement for its read-out, e.g. the waveform of the final state when using a flip-flop continues to exist in the last computational state. On the contrary, the final state  $|\Psi\rangle$  after a computation on a quantum computer is not directly accessible. The complex variables in Eq. (2.1) only define the *probabilities* to observe the state  $|\Psi\rangle$ , given by  $|c_0|^2$  and  $|c_1|^2$  to observe the qubit states  $|0\rangle$  and  $|1\rangle$ , respectively. Since the total probabilities must

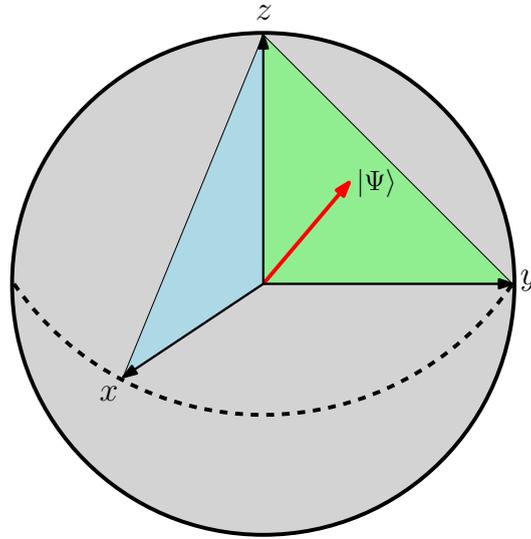


FIGURE 2.1: Bloch sphere representation of a general single qubit state  $|\Psi\rangle$ . The blue colour, green colour, and the dotted line show the  $xz$ ,  $yz$ , and  $xy$  planes, respectively.

sum to unity,  $|c_0|^2 + |c_1|^2 = 1$ . The read-out of  $|\Psi\rangle$  is a statistical process, requiring a set of several repeated preparations and measurements.

Since the bit exists in two states before, during, and after the measurement of a computation, it justifies the name *binary* digit. On the contrary, the qubit can be in any arbitrary superposition of its states before and during the computation. It only exists in a *binary* state after the measurement if the word "binary" is to be consistently understood in the classical sense. A more appropriate name may have been *qit* or quantum digit. However, for consistency throughout the literature, the word "qubit" will be followed.

For the case of a single qubit, it is instructive to look at a geometric representation on the *Bloch sphere*. The state of the qubit can be parametrised as

$$|\Psi(\theta, \phi)\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle, \quad (2.2)$$

where  $\theta$  and  $\phi$  are real numbers, and we ignore a global phase factor which has no observable effects [NC10]. The Bloch sphere is shown in Fig. 2.1. The unit vector  $|\Psi\rangle$  stretching to the sphere's surface represents the general state. The computation on a single qubit moves the vector around in arbitrary directions. The computational states  $|0\rangle$  and  $|1\rangle$  are shown in Fig. 2.2. Every point can be reached on the surface of the sphere with a suitable combination of  $\theta$  and  $\phi$  in Eq. (2.2). The choice of direction of the computational states, e.g.  $|0\rangle$  ( $|1\rangle$ ) towards  $+z$  ( $-z$ ) is a matter of choice or definition.

### 2.1.2 Multiple qubits

The elementary storage units of a classical computer need to be put together to store meaningful information consisting of combinations of 0s and 1s. For two such units put together,

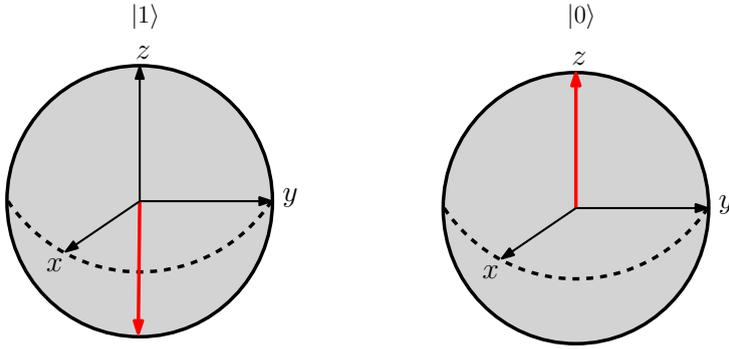


FIGURE 2.2: Bloch sphere representation of the two computational states (red line) of a single qubit.

the system is described by a set of four *ordered pairs*

$$\{00, 01, 10, 11\}, \quad (2.3)$$

where the order in which the bits appear is significant. The significance of the order implies that unless the bits are the same, any two orderings are not equivalent. Generalising to more than two units, the corresponding state of  $N$  bits is described by an  $N$ -fold Cartesian product containing  $2^N$  elements, given by

$$X = x_0 x_1 \dots x_{N-1} \in \{0, 1\}^N = \{00 \dots 0, 00 \dots 1, \dots, 11 \dots 1\}. \quad (2.4)$$

Similarly, a single qubit is not a very useful unit of processing information in itself unless multiple qubits are combined together to store and manipulate meaningful information. The quantum state of such a system is described by the multi-qubit state  $|\Psi\rangle$  which is a linear superposition of all possible computational states given by

$$\begin{aligned} |\Psi\rangle &= c_0 |00 \dots 0\rangle + c_1 |00 \dots 1\rangle + \dots + c_{2^N-1} |11 \dots 1\rangle \\ &= \sum_{x_0 \dots x_{N-1} \in \{0,1\}^N} c_{(x_0 x_1 \dots x_{N-1})_{\text{base } 10}} |x_0 x_1 \dots x_{N-1}\rangle \\ &= \sum_{k=0}^{2^N-1} c_k |X_k\rangle \quad \text{where} \quad \sum_k |c_k|^2 = 1. \end{aligned} \quad (2.5)$$

The multi-qubit states of the combined system of qubits can be in any state out of all the possibilities in Eq. (2.5). Note that the fundamental difference in classical and quantum computing does not merely arise by the description of their multi-unit storage given by Eq. (2.4) and Eq. (2.5), respectively. The difference arises from the fact that quantum superposition is an active part and is used during the computation, in contrast to classical superposition. Clearly, there exist states in Eq. (2.5) that cannot be written in a Cartesian product form as given in Eq. (2.4). A commonly known example of such a state is

$$|\Phi_2^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle). \quad (2.6)$$

The fact that  $|\Phi^-\rangle$  is a valid state is readily verified since  $\langle\Phi^-|\Phi^-\rangle = 1$ , implying that the total probability adds to unity, as required. Since  $|\Phi^-\rangle$  in Eq.(2.6) cannot be written as a product of

two independent single particle states, it is termed as an *entangled* state. A counterexample to an entangled state is a product state,

$$|\Phi_2'\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |01\rangle) = \frac{1}{\sqrt{2}}|0\rangle(|0\rangle - |1\rangle), \quad (2.7)$$

which can be written as a product of individual qubit states. Consider another simple example of a state

$$\begin{aligned} |\Phi_4^+\rangle &= \frac{1}{\sqrt{4}}(|1010\rangle + |0100\rangle + |0010\rangle + |0001\rangle) \\ &= \frac{1}{\sqrt{4}}((|101\rangle + |010\rangle)|0\rangle + |00\rangle(|10\rangle + |01\rangle)), \end{aligned} \quad (2.8)$$

for a four qubits system, which can only be partially written in a product of individual states. From a quantum computing perspective, most interesting are then the "fully-entangled" states, since it is their description that requires a complete statevector of length  $2^N$ . Since  $|\Phi\rangle$  is never directly observed, it can be said that entanglement is a property of our description of Nature and not of Nature itself, an important requisite. Therefore, it is misleading to say that entanglement itself (or real numbers, for that matter) can be observed.

Work has been done in identifying entangled states in physical phenomena through the separability criterion [HHH96] and entanglement witnesses [Ter00] in the context of Bell inequalities [Bel64; BA04]. The idea is to test a mathematical inequality where if a violation is observed, it is inferred that the state must have been entangled. Violations of the inequality are also used to infer that it is only a property of non-local systems. However, even local-realistic models reproducing the results of quantum mechanics violate Bell inequalities [De+07; Jat19]. Other alternative "subquantum" models that can reproduce the results of quantum mechanics using event-based simulations can be found in [Wil+20b; De+20]. The local realistic models obtain this violation only by post-selection on the full dataset, similar to the experiments.

The visual representation of a single qubit was easy to realise using the Bloch sphere, but this is not the case for multi-qubit systems. While the mapping from Eq. (2.1) to Eq. (2.2) is straightforward, a similar geometric mapping of Eq. (2.5) does not provide a complete picture of all the linear combinations possible. Attempts have been made that require more than one sphere per qubit to cover all possibilities [Wie14]. It is still useful, nevertheless, to visualise each qubit in a multi-qubit system on a single Bloch sphere per qubit [Wil20].

## 2.2 Quantum gates

A successful computation requires not only the storage of information but also its manipulation on a computer. The quantum computer would not be very useful without the means to manipulate the stored information. In this section, we look at the various operations that can be done on a quantum computer. These operations are known as quantum gates.

Most operations on the bit are designed to be irreversible. It is possible to design classical gates that are reversible [Ben73]. Reversibility is not a necessary condition for classical computing. In contrast, quantum gates and operations are necessarily reversible due to the unitarity of the underlying operators in quantum mechanics. In principle, there can be gates acting on an arbitrary number of inputs. However, it turns out that it is sufficient to have gates acting on up to two inputs to realise a universal computation [DiV95].

### 2.2.1 Single qubit gates

Single bit gates on a classical computer are one-input one-output gates. The simplest example of such a gate is the **NOT** gate, which *negates* the input value at its output. Similarly, for a quantum computer we can define a **NOT** gate that changes the computational state of the qubit from  $|0\rangle$  to  $|1\rangle$ , and vice versa. The interesting difference between the classical and quantum **NOT** gate, however, appears when the qubit is in some linear combination, e.g.  $c_0 \neq 0 \neq c_1$  in Eq. (2.1). In this case, given an input state as described in Eq. (2.1), the output is given by

$$\mathbf{NOT}(|\Psi\rangle) = \mathbf{NOT}(c_0|0\rangle) + \mathbf{NOT}(c_1|1\rangle) = c_0|1\rangle + c_1|0\rangle. \quad (2.9)$$

A more convenient way of representing the operation in Eq. (2.9) is to define a matrix of the quantum **NOT** gate, given by

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad (2.10)$$

and a vector corresponding to  $|\Psi\rangle$ , given by

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix}. \quad (2.11)$$

In this new notation, Eq. (2.9) is rewritten as

$$X|\Psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_0 \end{bmatrix}. \quad (2.12)$$

Using the matrix and vector representation, we have moved from a symbolic and abstract notation to a more concrete and numerical one. This opens the possibility to emulate quantum computers on classical computers.

There also exist other gates that do not have a corresponding classical counterpart. An example of such a gate is the  $Z$  gate, given by

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (2.13)$$

The action of the  $Z$  gate on the state  $|\Psi\rangle = c_0|0\rangle + c_1|1\rangle$  gives  $Z|\Psi\rangle = c_0|0\rangle - c_1|1\rangle$ . This produces no effect on the probabilities to measure the computational states,  $|c_0|^2$  and  $|c_1|^2$ .

A general single-qubit gate is realised as a three dimensional rotation on the Bloch sphere due to the correspondence between Eq. (2.1) and Eq. (2.2). Such a rotation is expressed as

$$R_\alpha(2\theta) = \exp(-i\theta\sigma^\alpha), \quad (2.14)$$

where  $\sigma^\alpha$  for  $\alpha \in \{x, y, z\}$  are the Pauli matrices

$$\sigma^x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma^y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma^z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (2.15)$$

Every point on the Bloch sphere can be transformed to every other point using suitable rotations. The rotations written explicitly in matrix form are

$$\begin{aligned} R_x(2\theta) &= \begin{bmatrix} \cos\theta & -i\sin\theta \\ -i\sin\theta & \cos\theta \end{bmatrix}, \\ R_y(2\theta) &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \\ R_z(2\theta) &= \begin{bmatrix} \cos\theta - i\sin\theta & 0 \\ 0 & \cos\theta + i\sin\theta \end{bmatrix}. \end{aligned} \quad (2.16)$$

The terms in Eq. (2.16) define a complete set of elementary single qubit rotations that can be used to manipulate any kind of information stored in the qubit(s).

Single qubit gates are also applied to entangled qubits. In this case, the gate acts only on the intended qubit and changes its state similar to the case of a single qubit. Let the state of a multi-qubit system be given by  $|\Psi\rangle$ , then the Pauli rotation on a single qubit with an index  $i$  by an angle  $2\theta$  is given by

$$R_{\alpha,i}(2\theta) = \exp(-i\theta\sigma_i^\alpha) = I_0 \otimes I_1 \otimes \cdots \otimes R_\alpha(2\theta) \otimes \cdots \otimes I_{N-1} \quad (2.17)$$

where  $I$  is the identity matrix denoted by

$$\sigma^I = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (2.18)$$

and the indexes correspond to the numbering of the qubits.

All the gates discussed in this section acted on one qubit at a time. Series of such gates can be combined to perform manipulation of the information only for each individual qubits. To make meaningful manipulations, information also needs to be interchanged or inter-acted between multiple qubits, which necessitates the existence of multi-qubit gates. We discuss such gates in the next section.

## 2.2.2 Multiple qubit gates

The logic gates that take input from at least two bits form the backbone of the classical computer. The combination of the **NOT** and the **AND** gate gives the **NAND** gate, from which every other digital logic gate can be constructed by suitable combinations. The **NAND** gate is a two-input, two-output gate and is considered universal for classical computing. Similarly, there exist quantum gates that also take two inputs and give two outputs. A very commonly used gate is the controlled **NOT** gate often termed as the **CNOT** gate. It is defined in the matrix notation as

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2.19)$$

The action of the **CNOT** gate is such that it reads one input, often termed as the control input, and if the input is  $|1\rangle$  it performs a **NOT** operation on the second input, often termed as the target input. If the control input is  $|1\rangle$ , the target input undergoes  $|0\rangle \rightarrow |1\rangle$  or  $|1\rangle \rightarrow |0\rangle$ . If the control input is  $|0\rangle$ , then it leaves the target qubit unchanged,  $|0\rangle \rightarrow |0\rangle$  or  $|1\rangle \rightarrow |1\rangle$ . For a concise notation, we use the convention **CNOT**( $c, t$ ), where  $c$  is the control input index and  $t$  is the target input index of the qubit. Note that the structure of the matrix in Eq. (2.19)

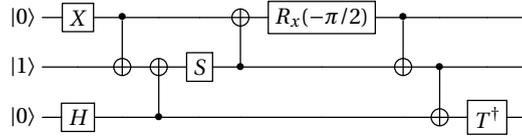


FIGURE 2.3: An exemplary circuit consisting of three qubits and some elementary gates. The **CNOT** gate consists of a solid dot for the control qubit and a circle with a plus sign in it for the target qubit.

assumes a certain notation of the corresponding state vector  $|\Psi\rangle$  it acts on. This is given by placing all the components such that they are arranged in the increasing order in the binary representation,

$$|\Psi\rangle = \begin{bmatrix} \Psi_{00} \\ \Psi_{01} \\ \Psi_{10} \\ \Psi_{11} \end{bmatrix}. \quad (2.20)$$

Taking the **CNOT** gate as a starting point, other gates can be envisioned that act on three or more qubits simultaneously, for example, the control-control-**NOT** or **CCNOT** gate. Since universal computation can be achieved by using the **CNOT** gate, the construction of three or more qubits gates is not necessary. Due to the fact that any unitary operator can be converted into a quantum gate, a large number of gates can be imagined. The practical implementation of actual hardware often has a smaller universal gate set and can vary from one device to another or over different architectures.

## 2.3 Quantum circuits

Qubits and gates acting on them are combined in quantum *circuits* to perform meaningful operations. A quantum circuit can be thought of as a set of instructions defined as a sequence of gates acting on pre specified qubits. Similar to classical computation, the idea is to decompose any arbitrary algorithm into a finite sequence of gates that perform the intended manipulations.

A prevalent understanding of circuits is through visual representations. A sample three qubits circuit is shown in Fig. 2.3. Each qubit is initialised in the computational basis states  $|0\rangle$  or  $|1\rangle$ . The horizontal lines show the working space of each individual qubit where gates can be applied. A gate operation is given by a box with an alphabet representing the type of gate. The control and target inputs of the **CNOT** gate are identified with the symbols  $\bullet$  and  $\oplus$ , respectively. The vertical line between these symbols acts as a visual guide for connecting the two qubits involved. In the circuit, the time can be thought of as flowing from left to right, where each gate takes some finite amount of time to be completed.

A circuit can either be (A) run directly on a quantum computer or be (B) run using an emulator on a classical computer. For case (A), we do not have direct access to  $|\Psi\rangle$ , and the circuit needs to be run several times where each time it produces one of the possible *bitstrings* from the underlying probability distribution. In certain cases, the individual measurements of the final states of the qubits need to be performed in a different basis. To achieve this, appropriate gates performing a basis change operation can be appended to the end of the

circuit. For case (B), it is possible to obtain  $|\Psi\rangle$  and perform *exact* calculations of the properties of the system. This case is for testing purposes. Note that the precision of numeric 'exactness' is limited to floating point value of  $10^{-15}$  and is sufficient for our purposes. Since the underlying probability distribution  $|\Psi\rangle$  is available, it is also possible to sample from it at random to directly emulate an ideal quantum computer. In this dissertation, we use all the above possibilities.

Despite the similarity to circuits used in classical computing, there are three essential things not allowed in quantum circuits [NC10]. First, the *feedback loops* common in classical computing where one of the wires is connected to another part of the circuit, is not allowed. Second, unlike classical wires, two quantum wires cannot be merged at any point of time in the computation. Third, classically it is possible to produce multiple copies of the output at the end of the computation, but this is not possible using a quantum computer. There is a simple reason why such possibilities do not exist for quantum wires, as follows. While the classical wire carries electrical current in the form of moving electrons under the influence of an electric potential, this picture does not accurately transform to the quantum 'wire' where something moving along the circuit cannot be pictured. Our understanding of the quantum computer is based on quantum theory, where the state of the system  $|\Psi\rangle$  is revealed only after the end of the computation when a measurement is performed, as a collection of frequencies of measured values. Quantum theory does not give us a detailed picture on an *ontological* level of each step of the computation. There is no conflict in this idea if we accept that quantum theory only makes statistical predictions.

## Properties

Given the visual representations of quantum circuits, it is helpful to define notions that will help characterise differences in circuits. Here we briefly define some properties of quantum circuits.

- **Wire:** horizontal lines which show the flow of time from left to right and on which gates are placed. See Fig. 2.3.
- **Total gates:** number of gate operations along all the paths on the wire.
- **Depth:** number of gate operations along the longest path on the wire counted such that all total gates can be executed. The depth is always an integer.

There may be other properties of a circuit that may become relevant for particular applications. These will be defined as necessary.

## 2.4 Quantum computer emulator

With the elementary units of a computer, namely the qubits, gates, and circuits ready, it becomes possible to visualise a quantum computer. A five-point criterion has been proposed for the physical realisation of any quantum computer [DiV00]. A significant amount of research has gone into developing them. Despite that, no error-free fully-functional quantum computer is currently available which has a larger number of qubits than can be *emulated* on current classical computers. Therefore, it is interesting to investigate if applications of quantum computers are successful on such emulators. Since current computers are small-scale and serve as prototypes, emulators have shown significant importance.

The terms emulator and simulator are often mixed or used interchangeably. We distinguish these two based on the following discussion. At least four factors influence the definition and

meaning of these words. These tend to vary more widely for the word simulator in contrast to the emulator. These factors are

1. **Aim:** The purpose of the effort, be it simulation or emulation, is a contributing factor to the definitions. For example, the aims of a simulation could be to model the laws of physics or train a medical student for emergency procedures.
2. **Object:** The aim is focused on an object to be studied. For example, the simulation of crowds exiting a stadium could be performed either on a digital computer (simulator A) or in a small-scale real-life experiment with humans (simulator B).
3. **Context:** The broad field of the aim and object under study influence how the words are perceived. The meaning may remain relevant only within that context or field.
4. **Platform:** The execution of the simulation or emulation is platform-dependent. In contrast to emulators, simulators are often implemented on a platform different from the object.

By fixing the above four criteria, a consistent and unambiguous definition of a simulator or an emulator can be defined. These can also help distinguish the different meanings of each word among two communities. For example, the word simulator is used extensively both in experimental and theoretical physics. On the one hand, it refers to actual hardware used as a 'quantum simulator' of interesting physics problems [BR12; BDN12], e.g. using ultracold atoms in an optical superlattice as the platform with the aim of studying many-body phenomena [Tro+12]. On the other hand, it refers to the modelling of physical phenomena in order to 'simulate' their properties. For example, using a programming language run on a classical computer as a platform with the aim of finding the ground state energy of the Hubbard model.

## Quantum computing

The broad field of quantum computing sets the context factor for defining simulators or emulators. These two words are used in the quantum computing context throughout the rest of this dissertation. By changing the context or the field, say to medicine, one or both the words may not exist, may exist but without regular usage, or may be used to imply different things. The meaning of both the words can change depending on the combination of aim, object, and platform for which they are used. In what follows, we set the definitions of each by fixing these three factors.

### Simulator

1. **Aim:** to produce and use a model that helps study, analyse, or imitate.
2. **Object:** a gate based quantum computer and the physics behind it.
3. **Platform:** programming language run on a classical digital computer.

### Emulator

1. **Aim:** to mimic or use as substitute.
2. **Object:** a gate based quantum computer.
3. **Platform:** programming language run on a classical digital computer.

### Jülich universal quantum computer simulator (JUQCS)

JUQCS [De +07; De +19] is a simulator insofar as it models the behaviour of a computer whose workings are described by quantum mechanics. This is done by representing the quantum computer's state through the wavefunction in the form of a complex state vector stored in the memory of a classical computer. Knowing the state vector, one models everything interesting about the quantum computer. This state vector can be used to calculate any property of interest or manipulated to *imitate* the time evolution of the quantum computer. The state vector is manipulated by multiplying it with appropriate matrices. While there exist emulators that can approximate the wavefunction to increase the total number of qubits, we only work with those that manipulate the complete wavefunction. For larger problems beyond approx. 30 qubits, it becomes necessary to employ supercomputers. We use the JUWELS supercomputer [Juw].

Consider that the following features are added to the simulator handling the state vector. First, a finite set of "gate" instructions are allowed as input along with information about the initial state of the state vector. Second, converting the state vector into samples or bitstrings that appear to be generated at random. These new features, along with the simulator, can mimic a quantum computer completely. This describes an emulator.

The following example considers both emulation and simulation. Assume that we are interested in finding the ground state energy of some material. Let the description of the material lattice be given in terms of a one-dimensional Heisenberg model. In the conventional sense, analytically solving for the energy on pen and paper is neither emulation nor simulation. In the sense of the four factors outlined above, it is a simulation where the platform is pen and paper. In the conventional sense, numerical techniques to solve for the energy on classical computers is termed as simulation. Variational techniques used here when run on an actual quantum computer would also be simulation. However, it would be emulation when run through JUQCS.

## 2.5 Quantum algorithms

The final step in using a fully functional quantum computer is an algorithm that can exploit its features. Once a quantum computer or an emulator is available, it becomes possible to execute a circuit. However, not every circuit is interesting. The quantum computer is not foreseen to perform every computation faster than a classical computer, only certain specific computations. The interest lies in those computations that can offer an advantage when executed on a quantum computer. A compilation of computations that perform meaningful calculations with the potential of offering some advantage will be referred to as quantum algorithms.

One of the first algorithms to show that quantum computers can deliver an exponential speed-up against a classical computer is Shor's algorithm [Sho94; Sho97]. There has been a large amount of research on algorithms ever since, and we would not attempt to make an exhaustive list here. This dissertation restricts itself to hybrid variational algorithms. However, before the stage for such algorithms can be set up, it is helpful to look at yet another established algorithm which carries out similar tasks, namely, the phase estimation algorithm. This algorithm is also a subroutine of Shor's algorithm and helps understand whence the computational power of quantum computer comes.

### 2.5.1 Phase estimation algorithm

We look at our first algorithm that makes use of the concepts discussed so far in this chapter. The phase estimation algorithm (PEA) was one of the first algorithms used to calculate the ground state energies of two small molecules [AG+05]. Algorithms often make use of sub-routines. The PEA makes use of a subroutine known as the *quantum Fourier transform*. To understand the working of PEA for the molecular problems, it is essential to understand the working of the quantum Fourier transform.

#### Quantum Fourier transform

A detailed introduction and suitable applications of the quantum Fourier transform are found in [NC10]. Here we only give a brief overview. The quantum Fourier transform takes as input a vector of complex numbers of length  $L$  (say  $|x\rangle$ ) and outputs another vector of the same length (say  $|y\rangle$ ) guided by the transformation

$$|x\rangle \rightarrow \frac{1}{\sqrt{L}} \sum_{y=0}^{L-1} e^{2\pi i xy/L} |y\rangle. \quad (2.21)$$

Since the length of the vector for the purpose of quantum computing will be always a power of 2, it is possible to set  $L = 2^N$  where  $N$  is the number of qubits. Furthermore,  $y$  can be written in fractional binary notation such that

$$y = y_1 \dots y_N \quad \text{and} \quad \frac{y}{2^N} = \sum_{k=1}^N \frac{y_k}{2^k}. \quad (2.22)$$

Substituting Eq. (2.22) in Eq. (2.21), we get

$$\begin{aligned} |x\rangle &\rightarrow \frac{1}{2^{N/2}} \sum_{y=0}^{2^N-1} e^{2\pi i x (\sum_{k=1}^N y_k / 2^k)} |y_1 \dots y_N\rangle \\ &= \frac{1}{2^{N/2}} \sum_{y=0}^{2^N-1} \prod_{k=1}^N e^{2\pi i x y_k / 2^k} |y_1 \dots y_N\rangle \\ &= \frac{1}{2^{N/2}} \bigotimes_{k=1}^N (|0\rangle + e^{2\pi i x / 2^k} |1\rangle). \end{aligned} \quad (2.23)$$

The task to perform the above transformation using a quantum circuit is carried out as follows. By using the Hadamard gate  $H$ , as shown in Fig. 2.4, each qubit is put into a superposition of states where each qubit attains the state  $(|0\rangle + |1\rangle)/\sqrt{2}$ . To implement the coefficient of  $|1\rangle$  given in Eq. (2.23) a series of unitary operations are applied using the gate  $U_k$  defined as

$$U_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i / 2^k} \end{bmatrix}. \quad (2.24)$$

By using the circuit shown in Fig. 2.4 and performing swapping operations at the end to reverse the order of the qubits, the quantum Fourier transform is efficiently implemented.

The quantum Fourier transform is used in several quantum algorithms, e.g. Shor's algorithm for factorisation [Sho99], solving linear systems of equations [HHL09], Grover's algorithm [Gro96], and quantum counting algorithm [BHT98].

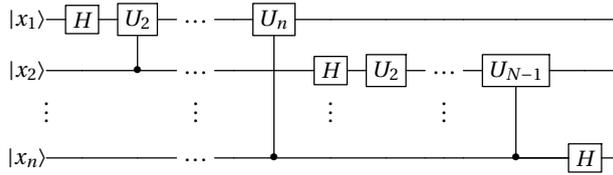


FIGURE 2.4: Circuit performing the quantum Fourier transform. After the end of the circuit swap operations (not shown) reverse the order of the qubits. The  $U_k$  gate is defined in Eq. (2.24).

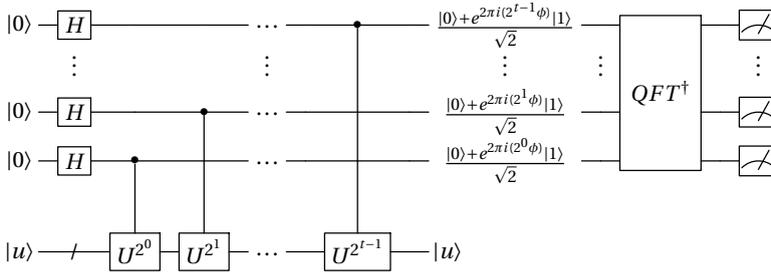


FIGURE 2.5: Circuit performing phase estimation. The qubits are divided into two sets, top and bottom. Top set is initialised in the  $|0\rangle$  state and bottom set in the  $|u\rangle$  state. The intermediate state of the top set of qubits is written explicitly. The inverse quantum Fourier transform is applied before measurement of the top set.

### Phase estimation

The Fourier transform expressed in Eq. (2.23) will help in the *phase estimation* as follows. Assume a unitary operator  $U$  with an eigenvector  $|u\rangle$  and eigenvalue  $e^{2\pi i\phi}$  where  $\phi$  is the unknown variable. Further assume that by some means the eigenvector  $|u\rangle$  has already been prepared on a quantum computer. The phase estimation algorithm uses two sets of qubits. The first set contains  $t$  qubits all initialised in the state  $|0\rangle$ . The choice of  $t$  depends on the accuracy to which  $\phi$  needs to be estimated as well as the probability of the procedure to be successful [NC10]. The second set of qubits contain the state  $|u\rangle$ . The number of qubits in the second set depends entirely on the specific problem.

The circuit used for the phase estimation is shown in Fig. 2.5. After preparing one set of qubits in the  $|0\rangle$  state and applying Hadamard gates on all of them, they are allowed to interact with the set of qubits containing the state  $|u\rangle$  through a sequence of controlled- $U$  operations. The state of the individual qubits after all the controlled- $U$  gates is also written in Fig. 2.5. This state resembles closely the last term in Eq. (2.23). This means we have a Fourier transformed state available with the unknown variable  $\phi$  embedded in the phase of the coefficient of  $|1\rangle^{\otimes N}$ . To extract the  $\phi$ , we need to perform an *inverse* Fourier transform given in the circuit shown in Fig. 2.5 as  $QFT^\dagger$ . Measurements of the top pair of qubits reveal the value after the inverse transform.

### Application to molecular problems

An essential ingredient of the PEA is the availability of the state  $|u\rangle$ . In the ideal case, it should be the ground state if one is interested in estimating the ground state energy of a system. If it is not the ground state, then the success of the PEA is proportional to the overlap of the state  $|u\rangle$  with the true ground state of the system [AG+05]. In quantum chemistry, approximations to the ground state wave functions are often given by the Hartree-Fock state  $|\Psi^{HF}\rangle$ . For small systems, the  $|\Psi^{HF}\rangle$  is a good approximation to the true ground state, however, it is known that for larger systems or systems close to their dissociation limit,  $|\Psi^{HF}\rangle$  has a vanishing overlap with the ground state [Koh99]. To overcome the problem of vanishing overlap of the initial state, methods have been suggested [BF28; Far+01]. Preparing  $|u\rangle$  is a problem to be solved in itself.

### 2.5.2 Drawbacks of phase estimation

PEA is one of the first algorithms to facilitate the estimation of eigenvalues of interesting Hamiltonians on a quantum computer. While a variant of it has been experimentally implemented for small systems [Pae+17], its drawbacks make it impractical to use on current or near-future quantum devices for realistic problems. We highlight three of these drawbacks.

The first drawback concerns the precision of the phase estimate. To reach precision  $\epsilon$ , PEA requires  $O(1)$  repetitions of circuits with depth  $O(1/\epsilon)$  [WHB19]. Since the required precision is high (small  $\epsilon$ ) for most practical applications, the inverse proportionality to circuit depth implies that PEA is out of reach of current as well as near-term devices. The second drawback concerns the preparation of the approximate ground state  $|u\rangle$ . While there are efficiently preparable ground state approximations to some Hamiltonians, it is not generally possible for any arbitrary one. Hence, all problems for which  $|u\rangle$  is not known remain out of reach of quantum computers via PEA. The third drawback is the large number of qubits required. PEA requires two sets of qubits, and depending on the problem size and the precision required; both can be very high and far beyond the reach of near-future quantum computers. It is estimated that the algorithm will require qubits in the range of millions for practical applications in quantum chemistry [Lee+21].

### 2.5.3 Motivation for variational algorithms

To first motivation is to shorten the long circuit depths of PEA in order to bring them closer to what is supported by current quantum computers. We require an algorithm whose precision of the eigenvalue estimate does not depend on the circuit depth. As we will see in the next chapter, quantum circuits of variational algorithms are short enough to be implementable on current computers and the eigenvalue estimate can be improved by sampling repeatedly from the circuit.

We also wish to solve problems whose approximate ground state may not be known beforehand. Intuitively, this will not be achieved without sacrificing something else. Along with the benefit of realistic circuit depths, variational methods allow the use of any ansatz to solve the problem. The sacrifice here is that not every ansatz will guarantee a complete overlap with the ground state. Whether a given ansatz will solve for the ground state energy is then a problem specific question. Will variational methods work for the Heisenberg model, the Hubbard model, or quantum chemistry?

The first few applications of variational methods, among them the variational quantum eigensolver [Per+14; O'M+16], have been shown to work only for prototype, small scale problems. Variational methods have been used to calculate the ground state energies of small

molecules on superconductivity based [Kan+17; Col+18; McC+19; Aru+20] or trapped ion [Nam+20] computers. The number of qubits used were far less than what can be emulated using a classical computer, where memory is the limiting factor. All variational applications on real hardware known to the author remain below this threshold. Therefore, it is pertinent to ask: do variational methods work beyond the small scale?

Variational methods make use of the classical optimisation algorithms. However, achieving useful results using these algorithms has been recently argued to be NP-hard [BK21]. In what ways do we modify them to avoid the problem and make them usable for variational methods? How does noise influence our ability to solve for the ground state energy?

## Chapter 3

# Theory of variational methods

### 3.1 Introduction

The concept of hybrid variational methods is introduced in detail. The chapter is divided into two main sections, namely, variational algorithms and classical optimisation. The former focuses on the concept and implementation of variational methods for applications relevant for this work. The latter focuses briefly and specifically on that part of the field of classical optimisation that is relevant for variational methods used in the rest of the dissertation.

### 3.2 Theorem of variation

Consider a Hamiltonian  $H$  describing a physical system of interest. Let the different possible energies of the system be given by  $E_k$  for  $k$  eigenstates of the system. We are interested in finding the energy  $E_0$  of the ground state (represented by  $k = 0$ ) of this system. Consider two different wavefunctions describing the ground state of the system. The first one describes the true ground state and is given by  $|\widetilde{\psi}_0\rangle$ , which is unknown to us. The second, which we call a “trial wavefunction”, is given by  $|\psi_0\rangle$ . The expectation value of  $H$  using the trial wavefunction, is given as

$$\langle H \rangle = \frac{\langle \psi_0 | H | \psi_0 \rangle}{\langle \psi_0 | \psi_0 \rangle}, \quad (3.1)$$

where we allow the possibility that  $|\psi_0\rangle$  may not be normalised. The variational theorem states that the expectation value of  $H$  is an upper bound to the true ground state energy of the system,

$$\langle H \rangle \geq E_0, \quad (3.2)$$

where the equality sign holds only for the case where  $|\psi_0\rangle = |\widetilde{\psi}_0\rangle$ . We ignore differences in global phases. We assume the ground state is non-degenerate for simplicity but without loss of generality. Thus, the variational principle allows the use of any trial wavefunction to estimate the ground state energy. However, with this broad scope the associated drawback is that not every trial wavefunction will adequately estimate the energy.

#### Proof

Let us expand the trial wavefunction in terms of a normalised set of energy eigenfunctions of  $H$  as

$$|\psi_0\rangle = \sum_{k=0}^{\infty} |\widetilde{\psi}_k\rangle \langle \widetilde{\psi}_k | \psi_0 \rangle, \quad (3.3)$$

where  $|\widetilde{\psi}_k\rangle$  is an *exact* energy eigenfunction of  $H$ , giving

$$H|\widetilde{\psi}_k\rangle = E_k|\widetilde{\psi}_k\rangle. \quad (3.4)$$

We remind ourselves that the ground state is given for  $k = 0$ , giving  $H|\widetilde{\psi}_0\rangle = E_0|\widetilde{\psi}_0\rangle$ . Using Eq. (3.3) and Eq. (3.4) in Eq. (3.1), we get

$$\begin{aligned} \langle H \rangle &= \frac{\langle \psi_0 | H | \psi_0 \rangle}{\langle \psi_0 | \psi_0 \rangle} = \frac{\sum_{k=0}^{\infty} \langle \psi_0 | \widetilde{\psi}_k \rangle \langle \widetilde{\psi}_k | H | \widetilde{\psi}_k \rangle \langle \widetilde{\psi}_k | \psi_0 \rangle}{\sum_{k=0}^{\infty} \langle \psi_0 | \widetilde{\psi}_k \rangle \langle \widetilde{\psi}_k | \psi_0 \rangle} \\ &= \frac{\sum_{k=0}^{\infty} |\langle \psi_0 | \widetilde{\psi}_k \rangle|^2 \langle \widetilde{\psi}_k | \widetilde{\psi}_k \rangle E_k}{\sum_{k=0}^{\infty} |\langle \widetilde{\psi}_k | \psi_0 \rangle|^2} \\ &= \frac{\sum_{k=0}^{\infty} |\langle \psi_0 | \widetilde{\psi}_k \rangle|^2 (E_k - E_0 + E_0)}{\sum_{k=0}^{\infty} |\langle \widetilde{\psi}_k | \psi_0 \rangle|^2} \\ &= \frac{\sum_{k=0}^{\infty} |\langle \psi_0 | \widetilde{\psi}_k \rangle|^2 (E_k - E_0)}{\sum_{k=0}^{\infty} |\langle \widetilde{\psi}_k | \psi_0 \rangle|^2} + \frac{\sum_{k=0}^{\infty} |\langle \psi_0 | \widetilde{\psi}_k \rangle|^2 E_0}{\sum_{k=0}^{\infty} |\langle \widetilde{\psi}_k | \psi_0 \rangle|^2} \\ &= \frac{\sum_{k=0}^{\infty} |\langle \psi_0 | \widetilde{\psi}_k \rangle|^2 (E_k - E_0)}{\sum_{k=0}^{\infty} |\langle \widetilde{\psi}_k | \psi_0 \rangle|^2} + E_0 \\ &\geq E_0, \end{aligned} \quad (3.5)$$

where the first term is the product of squared values and  $(E_k - E_0)$ , in which the latter is always non-negative due to our definition that  $E_0$  is the lowest energy. Thus, for any trial wavefunction  $|\psi_0\rangle$  describing the ground state, the corresponding  $\langle H \rangle$  will always be more than or equal to  $E_0$ . Substituting  $|\psi_0\rangle = |\widetilde{\psi}_0\rangle$  in Eq. (3.5) gives the true ground state energy  $\langle H \rangle = E_0$ .

## Application and examples

To see the potential of the variational theorem we apply it to two simple problems using suitable trial wavefunctions. The application of the method to solving problems proceeds as follows.

1. Construct a trial wavefunction that has one or more parameters  $a, b, c, \dots$ , etc. It can be inspired by the dynamical variables of a problem. Check the trial wavefunction for consistency with the problem. For example, some problems require that the wavefunction must vanish at certain regions.
2. Evaluate the expectation value of the Hamiltonian with the trial wavefunction giving the energy as a function of the parameters,  $E(a, b, c, \dots)$ .
3. Minimise  $E(a, b, c, \dots)$  with respect to the parameters to evaluate the lowest energy obtainable using that trial wavefunction.

The obtained energy is then the best approximation to the ground state energy of the system of interest using that trial wavefunction. Note that the analytical expression for step 2 is often only available for elementary examples. In general, numerical estimates are relied upon. These numerical estimates offer local information about the function around a point. To better understand the variational principle and the benefits it offers, consider two examples for which exact wavefunctions are known. We will not use the exact wavefunctions but some other trial wavefunctions. The first example will be about the harmonic oscillator, owing to problems of theoretical interest and approximations, and the second example will be

of the hydrogen atom, owing directly to physical problems. Both examples will illustrate the working of the variational principle to calculate the ground state energy.

### Harmonic oscillator

The well known harmonic oscillator Hamiltonian is given as

$$H = \frac{-\hbar^2}{2m} \frac{d^2}{dx^2} + \frac{m\omega^2 x^2}{2}, \quad (3.6)$$

where  $m$  is the mass,  $\omega$  is the frequency, and  $\hbar$  is the Planck's constant. For the purpose of demonstrating the variational method, even though we know the correct functional form of the wavefunction which can give us the exact ground state energy, we will use a different one. This is done in view of the fact that in an arbitrary problem the correct functional form of the wavefunction giving the exact ground state energy is, in general, unknown. We will guess the trial wavefunction using insight, shape of the potential, and conditions that a wavefunction must satisfy. A reasonable guess (inspired from [Mei96]) is given by

$$\psi_0(x) = \frac{b}{px^2 + a^2}, \quad (3.7)$$

where we will alternatively use  $a$  and  $p$  as parameters, such that when  $a$  is a parameter,  $p$  is a constant, and vice versa. The numerator  $b$ , as our calculations will reveal, will drop out in both cases and thus cannot be used as a parameter. Note that  $\psi_0(x) \rightarrow 0$  when  $x \rightarrow \infty$ , as suggested by the potential.

Now we compute  $\langle H \rangle$  as per Eq. (3.1). The denominator of the term, giving the normalisation constant, is given as

$$\langle \psi_0(x) | \psi_0(x) \rangle = \int_{-\infty}^{\infty} dx \frac{b^2}{(px^2 + a^2)^2} = \frac{\pi b^2}{2a^3 \sqrt{p}}. \quad (3.8)$$

The numerator of the term is given as

$$\begin{aligned} \langle \psi_0(x) | H | \psi_0(x) \rangle &= \int_{-\infty}^{\infty} dx \left( \frac{b}{px^2 + a^2} \left( \frac{-\hbar^2}{2m} \frac{d^2}{dx^2} + \frac{m\omega^2 x^2}{2} \right) \frac{b}{px^2 + a^2} \right) \\ &= \int_{-\infty}^{\infty} dx \left( \frac{-b}{px^2 + a^2} \frac{2bp(3px^2 - a^2)}{(a^2 + px^2)^3} \frac{\hbar^2}{2m} + \frac{b^2 x^2}{(px^2 + a^2)^2} \frac{m\omega^2}{2} \right) \\ &= \frac{\hbar^2}{2m} \frac{\pi b^2 \sqrt{p}}{4a^5} + \frac{m\omega^2}{2} \frac{\pi b^2}{2ap^{3/2}}. \end{aligned} \quad (3.9)$$

Combining the numerator and denominator, we get

$$\langle H \rangle = \frac{\hbar^2 p}{4ma^2} + \frac{m\omega^2 a^2}{2p} = E(a, p), \quad (3.10)$$

which is the energy as a function of  $a$  and  $p$ .

**Case I:** Let  $a$  be the parameter and  $p$  be the constant, so that  $E(a, p) = E(a)$ . To find the ground state energy we have to minimise  $E(a)$ . Thus, we set

$$\frac{d}{da} E(a) = 0, \quad (3.11)$$

which gives

$$a_{\min}^2 = \frac{\hbar p}{\sqrt{2}m\omega}, \quad (3.12)$$

where  $a_{\min}$  is the minimum value. Substituting Eq. (3.12) into the energy in Eq. (3.10), we get the minimum energy

$$E(a_{\min}) = E_0^a = \frac{\hbar\omega}{\sqrt{2}}. \quad (3.13)$$

**Case II:** Let  $p$  be the parameter and  $a$  be the constant, so that  $E(a, p) = E(p)$ . Similar to the previous case, we set

$$\frac{d}{dp}E(p) = 0, \quad (3.14)$$

which gives

$$p_{\min} = \frac{\sqrt{2}ma^2\omega}{\hbar}, \quad (3.15)$$

where  $p_{\min}$  is the minimum value. Substituting  $p_{\min}$  into the energy, we get the minimum energy

$$E(p_{\min}) = E_0^p = \frac{\hbar\omega}{\sqrt{2}}. \quad (3.16)$$

The absence of  $b$  in Eq. (3.10) makes it clear that it could not be used as a parameter, but only as a constant, and in this case Eq. (3.10) directly gives the ground state energy. In case I, using  $a$  as the parameter the minimum energy was  $\hbar\omega/\sqrt{2}$ , which came out to be the same as that of case II, where  $p$  was used as the parameter. Thus, from this example of the wavefunction as given in Eq. (3.7), it seems that as long as a parameter appears in the energy equation, regardless of where it is placed in this trial wavefunction, it does not change the minimum energy. Given that Eq. (3.15) is the same as Eq. (3.12), there was some liberty in parameterising the function. In this example the overparametrisation does not influence our ability to reach the minimum energy. However, it is an open question if overparametrisation is helpful for variational algorithms relevant for quantum computers.

The wavefunction which gives the exact ground state energy of the harmonic oscillator is given by  $\widetilde{\psi}_0(x) = e^{-ax^2}$ , where  $a$  is the variational parameter. Doing the complete calculation using this wavefunction will give the exact ground state energy  $E_0 = \hbar\omega/2$ . Knowing the exact ground state energy, we may calculate the error in our guessed trial-wavefunction energy, which in this case is given as

$$\frac{E_0^a - E_0}{E_0} = \sqrt{2} - 1 \approx 41\%, \quad (3.17)$$

which is rather large and other guesses might perform better. A detailed and intuitive introduction to the variational method for the harmonic oscillator is found in [Goo77]. For solutions with different trial wavefunctions, see [Mei96]. Interestingly, trial wavefunctions that give the best approximation to the ground state energy may not be the best approximation to the exact wave function, as required to calculate other physical properties [RM86]. The application of the variational principle for a different potential, that of a particle in a box, can be found in [CD00].

### Hydrogen atom

The Hamiltonian for the radial part of the hydrogen atom (in Hartree atomic units) is given by

$$H = -\frac{1}{2r^2} \frac{d}{dr} r^2 \frac{d}{dr} - \frac{1}{r}, \quad (3.18)$$

where  $r$  is the dynamical variable describing the electron-proton distance. For the purpose of the demonstration, consider once again the trial wavefunction given in Eq. (3.7) and replace the variable  $x$  with  $r$ , having the parameters  $a, b, p$  giving the normalisation constant

$$\langle \psi_0(r) | \psi_0(r) \rangle = \int_0^\infty \frac{r^2 b^2 dr}{(pr^2 + a^2)^2} = \frac{\pi b^2}{4ap^{3/2}}. \quad (3.19)$$

Note that due to the constraints of the problem, we only use the positive part of our trial wavefunction, hence the new integration limits are from zero to infinity. We now have to calculate

$$\begin{aligned} \langle \psi_0(r) | H | \psi_0(r) \rangle &= \int_0^\infty r^2 dr \psi_0^*(r) H \psi_0(r) \\ &= \int_0^\infty dr \frac{br^2}{pr^2 + a^2} \left( -\frac{1}{2r^2} \frac{d}{dr} r^2 \frac{d}{dr} - \frac{1}{r} \right) \frac{b}{pr^2 + a^2}. \end{aligned} \quad (3.20)$$

Equation (3.20) is solved for  $b = p = 1$  in [Som11]. Dividing the solution by Eq. (3.19) gives the energy

$$E(a) = \frac{1}{4a^2} - \frac{2}{\pi a}. \quad (3.21)$$

The minimum energy is then given as

$$E(a_{\min}) = -\frac{4}{\pi^2} \approx -0.405 \text{ Hartree}. \quad (3.22)$$

This trial function gives the minimum energy of  $-11.0$  eV, which is higher than the exact value of  $-13.6$  eV [Som11].

For more advanced trial wavefunctions and how they improve the ground state energy, we refer to [HA18]. Once again, a wavefunction that gives a better ground state energy may not yield a better value for another molecular property, see [PT19]. If the trial ground state energy approaches the true ground state energy, the corresponding trial wavefunction also approaches the exact one, as can be clearly seen from the example presented in [Ben93]. These facts will continue to apply to quantum variational methods. Both the examples illustrate that the choice of the trial wavefunction, referred to as an ansatz in the context of hybrid variational methods, is critical to finding the ground state energy.

The aim of finding the ground state energy for practical quantum chemistry requires that the accuracy exceeds that of the available experimental techniques. This is known as chemical accuracy. An accuracy of 1 kcal/mole or 0.043 eV has been commonly accepted to be appropriate [Pop99].

### 3.3 Variational methods

The core idea of variational methods is to parametrise a trial wavefunction,  $|\psi\rangle \rightarrow |\psi(\theta)\rangle$ , such that an ansatz containing the parameters  $\theta$  is executed on the quantum computer while the numerical values of the parameters are decided by a supporting classical computer. In this way, variational methods are a hybrid of quantum and classical counterparts. Given that we are interested in the lowest eigenvalue or the ground state of physically interesting Hamiltonians, we focus our study of the variational methods to finding the ground states.

Figure 3.1 visualizes the different components common to any hybrid quantum-classical variational method. As a hybrid, it combines alternating workflows between a quantum and a classical processing unit. After both the units are set up, a calculation is started by initialising

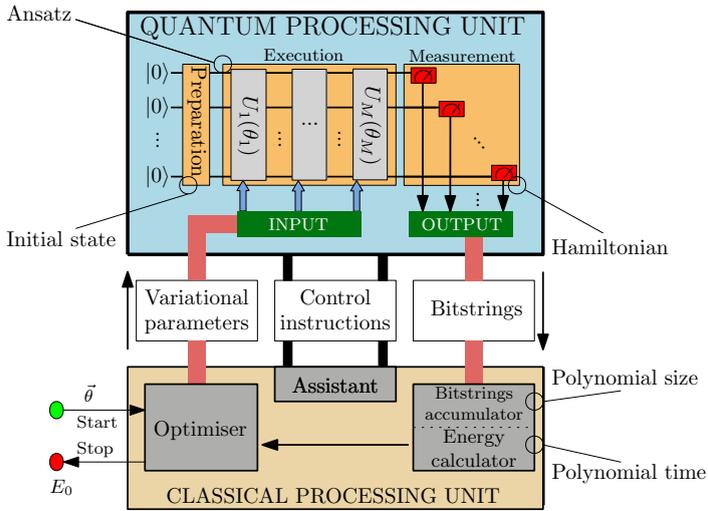


FIGURE 3.1: Schematic of the variational method showing quantum and classical processing units. The circuit prepares an initial state from  $|0\rangle^{\otimes N}$  and successively applies parametrized unitary operators  $U_1(\theta_1), \dots, U_M(\theta_M)$ . The bitstrings from measurement of the final state are accumulated and used to calculate the energy. The optimiser suggests parameters and iteratively minimises the energy.

the parameters as input to the classical unit. The first task of the quantum unit is to prepare a problem-specific initial state. This initial state can have a significant effect on the ability of variational methods to solve a problem. Next, it takes as input the numeric values of the variational parameters suitably placed in a quantum circuit. The circuit is then executed. After the execution, a measurement in a suitable basis gives the output as a sequence of bits. Each bit corresponds to the measurement outcome of each qubit. The bitstring of the outcome is then transferred to the classical unit. The entire process at the quantum unit is then repeated several times and the bitstrings are accumulated in the classical unit. The energy is calculated using the accumulated bitstrings. The numeric value of the energy is then fed to an optimiser. The optimiser computes the next set of parameters with the aim that successive iterations optimise the energy. The parameters are sent to the quantum unit and the quantum-classical loop repeats itself. The optimiser checks for some internal criteria each time it gets an energy input and it terminates the loop if the criteria are met, giving  $E_0$ .

For a variational method to be successful, it is necessary that none of the working parts act as an impediment. An impediment is any task within the method whose completion time or resources would scale exponentially. On the quantum side, the total number of samples required to estimate the expectation value should not scale exponentially. The number of qubits required to solve a problem should not scale exponentially when the problem size is increased. On the classical side, the accumulated bitstrings should avoid exponential storage requirements and the time required to calculate the energy from them should also not be exponentially scaling. The time required by the optimiser to suggest the next parameters should also not be exponentially increasing.

The three essential ingredients for the quantum processing unit of any variational algorithm are:

1. **Ansatz.** An ansatz is a quantum circuit that implements  $|\psi(\theta)\rangle$  on a quantum computer. The chosen ansatz will parametrically explore the Hilbert space. We require that (a) there is a systematic procedure to create a circuit corresponding to a parametrised state  $|\psi(\theta)\rangle$ , and (b) the circuit has a finite sequence of quantum gates that could be executed on the quantum computer. It is essential that the time taken to perform (a) and (b) does not scale exponentially with the number of qubits.
2. **Problem.** For physical systems, the given problem is written in terms of a Hamiltonian. For optimisation problems, the given problem is often encoded into a *cost function*, sometimes also known as the loss function. The circuit is appropriately adjusted to calculate the expectation value of the stated problem.
3. **Initial state.** For a quantum computer with  $N$  qubits, there are  $2^N$  possible computational states that the computer can be prepared in. The choices of initial states with arbitrary superpositions between  $N$  qubits is infinite. Often a variational algorithm's ability to solve the problem depends on the choice of the initial state.

Similarly, the essential ingredients for the classical processing unit of any variational algorithm are:

1. **Optimiser.** The task of an optimiser is to lower the energy after successive iterations and to stop the calculation. Optimisation algorithms can be broadly categorised in two categories, namely, gradient based and gradient free. Each has its own merits and drawbacks depending on whether a quantum computer or a quantum computer emulator is used for the task. Gradient based optimisers are commonly accepted to converge faster for noise-free functions. Popular candidates are the SLSQP and BFGS algorithms [JOP+01; Kra88; GF19]. However, the output of even an error-free quantum computer will be stochastic in nature, and it remains part of active research to answer which category is preferable. Several stochastic methods have been recently developed [Arr+20; Swe+20; O'B+19; MNM20; Sto+20; Ber+18].
2. **Bitstrings accumulator.** A quantum computer will output one bitstring at a time and the memory in the classical computer is used to store these. Each bitstring requires  $N$  bits of classical storage. Assuming that a polynomial number of measurements (each giving a bitstring) are sufficient to calculate the energy, the total storage requirements do not scale exponentially with increasing  $N$ .
3. **Energy calculator.** The accumulated bitstrings are suitably manipulated to calculate the expectation value or the energy. The manipulation would require polynomial time if the total bitstrings to be manipulated are polynomial in number. The energy may be only one of the many interesting properties of a given Hamiltonian. The hybrid quantum-classical loop can be over other scalar quantities as well, e.g. magnetisation, overlap with the ground state (for testing purposes), etc.

### The assistant

An additional ingredient of the classical processing unit is an "assistant" (see Fig. 3.1). For variational algorithms, the classical assistant serves the purpose of controlling how many repetitions of a circuit need to be performed, to change the state of the quantum computer at the end of a computation to a suitable basis as per a given Hamiltonian, etc. It is not directly involved in the variational loop discussed previously. The instructions are not necessarily sent from the same CPU that runs the optimiser. Current superconducting type quantum computers are assisted by microwave electronics and pulse-shaping to realise one- or two-qubit gates

[GCS17; Wil20]. See [Bra19] for an extensive list of control electronics. After a computation on the quantum processor, the measurements are performed using "classical" equipment, e.g. using a Josephson amplifier [Abd+11] or a low-inductance undulatory galvanometer [Hov+14]. We term the instruments as classical not because quantum effects do not appear in them, but because such effects are not relevant for their contribution to the process. Post-processing of the readout measurement is done digitally either by classical computers or customised field programmable gate arrays [Ris+12; Vij+12].

The number of individual terms in a Hamiltonian often increase with  $N$  and the elementary technique of measuring one term at a time can incur significant overhead even for a quantum computer. One solution to this problem is to measure commuting terms of the Hamiltonian together. Grouping such commuting terms, however, is a difficult and often a hard problem. Several recent studies have suggested new ways to minimise the total computational time by proposing methods that offer improvement over the elementary technique [Gok+20; Hug+21; Had+22]. When analytical arguments are exhausted to reduce such terms, pre-processing the terms is carried out on a classical assistant.

Often quantum circuits can be optimised to produce new circuits that perform the same computation but with (significantly) fewer number of gate operations; such a task is performed by an assistant. The optimisation of quantum circuits has emerged into a field of its own, often termed in the literature as quantum compiling [Mar+21]. Although quantum compiling can be used in the context of both gate-based quantum computers and quantum annealers, we restrict our focus on the former. Several algorithms have been proposed that perform quantum compiling. Examples include polynomial-time optimization of circuits consisting of Clifford and T gates using matroid partitioning [AMM14], techniques to reduce circuit depths of non-Clifford type circuits [KW20], using quantum Karnaugh maps [Bae+20], using machine learning for circuit optimisation [Mor+21; Fö+21], among several others [SPK13; Mas+08; PF13; Wu+20]. Thus, classical computers assist quantum computers by helping reduce the total number of either simultaneous and/or total gate operations. It is important to note that quantum compiling, except in the case of a single qubit, is a hard problem with algorithms requiring classical runtime exponential in the number of qubits [HC18] and heuristics do not guarantee optimality [Mas+08].

Furthermore, the assistant can completely take over the tasks of a quantum unit through an emulator [De +07; De +19] up to a certain number of qubits, limited by the memory which scales exponentially with  $N$ .

## Applications

Different applications of variational methods on quantum computers often inspire different names for such algorithms. We briefly differentiate between the use of the words *method* and *algorithm*. The mention of variational methods is used to imply the general systematic procedure to solve problems on a quantum computer using the variational principle. It is the method that is illustrated in Fig. 3.1. A method, in our definition, does not itself solve any specific problem but outlines the techniques to do so. In contrast, an algorithm consists of rules that make the method specific to a problem or set of problems. A method becomes an algorithm when at least the problem or the ansatz is specified. Some algorithms specify more rules than others. For example, the quantum approximate optimisation algorithm [FGG14] is specific to combinatorial optimisation problems combined with an alternating operator ansatz but the variational quantum eigensolver [Per+14; McC+16] generally only specifies that the problem be described by a physically relevant Hamiltonian. The underlying idea and mechanisms are very similar despite the different names. Several applications of variational methods have been found beyond solving optimisation problems or finding ground

states. In this section, we briefly outline the scope of applications that have been found for variational methods.

Algorithms for solving systems of linear equations exponentially faster than classical computers have been envisioned for quantum computers [HHL09]. Crude estimates outline that to get an advantage over classical computers, at least 340 fault free qubits and circuit depths of order  $10^{25}$  are needed [Sch+17]. These requirements put all quantum computers out of scope within this decade. Heuristics have emerged to fill the gap by proposing variational methods as the remedy [HBR21], e.g. the variational quantum linear solver [BP+19; PWK22]. Systems of nonlinear equations are also central to various scientific fields. Variational methods have been proposed that utilise multiple copies of variational quantum states to treat nonlinearities which solve such systems [Lub+20]. Another variational algorithm has been used to solve an instance of the Navier-Stokes equation [KPE21].

Factoring prime numbers is one of the earliest conceived applications that can give an exponential advantage [Sho99]. However, practical advantage is unforeseen for near term quantum computers. Variational quantum factoring [Ans+19] has been proposed for such NISQ computers. The proposal reformulates factoring as a solution to the ground state problem for a classical Ising model. See [Kar+21] for an experimental demonstration.

As mentioned in the previous section, quantum compiling is a hard problem for classical computers. Attempts have been made to utilise variational methods for it [Kha+19] that essentially transfer the task onto quantum computers (see [Car+20] for experimental demonstration). The task can also be achieved by reformulating it as a task of finding the ground state energy of a suitable Hamiltonian [JB22].

Quantum machine learning seeks to learn patterns from available data in order to make predictions for unseen data [Bia+17]. There exist several quantum machine learning models that can be understood from the viewpoint of variational methods [Cer+21]. There has been development of classifiers [Wil+20; Sch+20] (see [Hav+19; Kus+21] for experimental demonstrations), autoencoders [ROAG17; VPB18], and quantum neural networks [FN18].

While we have mentioned a few applications of variational methods for quantum computers, they are a work in progress and more should be expected to appear in the future. The wide range of application of such methods can be considered the stepping stone towards practical quantum computing until (or even after) fully fault tolerant machines become available.

### Excited states

While one of the main tasks of variational algorithms is to find ground state energies, they can be extended to also calculate excited states energies [HWB19; Jon+19; NMF19]. One way to do so is proposed in [HWB19] and is summarised as follows. The core idea is to change the function that is minimised such that a successful variational minimisation leads to the required excited state energy. The changed function to be minimised contains the original Hamiltonian  $H$  from Eq. (3.5) along with additional terms such that

$$\langle F \rangle = \langle H \rangle + \sum_{i=0}^{k-1} \beta_i |\langle \psi_k(\theta) | \psi_i(\theta') \rangle|^2, \quad (3.23)$$

where  $k = 1$  gives the first excited state,  $k = 2$  gives the second excited state, and so on, and  $\theta$  ( $\theta'$ ) is the set of parameters for the  $k^{\text{th}}$  run of the variational algorithm. The  $\beta_i$ 's are sufficiently large constants. Evaluating the  $k^{\text{th}}$  excited state energy requires the precalculation

of  $k-1$  previous lower energy states. While the calculation of the first term  $\langle H \rangle$  is straightforward using variational methods, the calculation of the overlap is estimated to a given precision by the fraction of all-zero bitstrings [Hav+19], which can be improved by using variational methods also for the overlap term [HWB19].

### 3.3.1 Automating circuit preparation

Prototype quantum computers have a small number of qubits and circuits have a shallow circuit depth, thus they can be created manually by placing each gate on a visual digital drawing board. In contrast, as discussed in the previous section, algorithms that will achieve quantum advantage will require a large number of qubits and large circuit depths. In such cases it would be no longer possible to manually create circuits and it becomes necessary to automate the process. Additionally, once the proverbial wheel has been invented, it can also roll the easier circuit creations more efficiently than doing it manually even before the realisation of quantum advantage. In this section, we design and implement a circuit preparation algorithm for a classical computer that takes as input alphabetical strings and creates a corresponding circuit. The algorithm is then used to create almost all the circuits used in this work. Although a similar algorithm already exists to create circuits for Hamiltonian dynamics [RWS12], it is unsuitable for variational methods. In contrast, the algorithm described here can be used to create circuits for Hamiltonian dynamics with a small modification.

An ansatz is a collection of unitary operators that acts on the initial state of a quantum computer. The state of the computer as a function of the parameters is given by

$$|\Psi(\boldsymbol{\theta})\rangle = \mathbb{U}(\boldsymbol{\theta})|\Psi_0\rangle, \quad (3.24)$$

where  $|\Psi_0\rangle$  is the initial state. A specific  $N$  qubits parametrised ansatz that contains sums of products of Pauli operators (see Eq. (2.15)) is given by

$$\mathbb{U}(\boldsymbol{\theta}) = e^{-iA(\boldsymbol{\theta})}, \quad (3.25)$$

where

$$A = \sum_{j=1}^m c_j \hat{a}_j^{\otimes N}, \quad \hat{a}_j^{\otimes N} = \otimes_{l=1}^N \hat{\Theta}_{jl}, \quad (3.26)$$

where  $\hat{\Theta}_{jl} \in \{\sigma^I, \sigma^x, \sigma^y, \sigma^z\}$  and  $m$  is the total number of products of Pauli operators. While an ansatz written in the form  $\mathbb{U}(\boldsymbol{\theta})$  is most general, several interesting ansätze can be written in the form  $\exp(-iA(\boldsymbol{\theta}))$ . The full implementation of such an ansatz as given in Eq. (3.25) is not a simple task if all the terms in an ansatz do not commute. In such a case, factorisation of the matrix exponential is required [DR87]. The factorisation further creates long series of products of more unitary operators whose implementation would require circuits with a large number of gates and circuit depths well beyond the scope of near future quantum computers.

To solve the issue of factorisation, we consider ansätze already in a factorised form which circumvent the creation of long circuits. A simple way forward is to consider each term in Eq. (3.26) separately such that the ansatz no longer contains a sum of operators but a set of operators,

$$A = \{c_j \hat{a}_j^{\otimes N}; j = 1, \dots, m\}. \quad (3.27)$$

Operators of the form given in Eq. (3.27) will form the basis of the construction of the circuits used in part of this work. The terms in the ansatz can be parametrised individually or in groups. Since the Pauli basis is a complete basis set, any other representation can be written in terms of the Pauli basis by a suitable transformation. The exponentiation of  $A$  and

individual parametrisation of each term in it gives the unitary operators

$$\mathbb{U}(\theta) = e^{-i\theta_m c_m \hat{a}_m^{\otimes N}} \dots e^{-i\theta_2 c_2 \hat{a}_2^{\otimes N}} e^{-i\theta_1 c_1 \hat{a}_1^{\otimes N}}. \quad (3.28)$$

The generated circuit can be stored and communicated with others through the standard open quantum assembly language [Cro+17]. An algorithm that takes an input, processes it, and systematically outputs a circuit works as follows.

### Input

The operators can be input as strings no longer than a polynomial size representation, in order to keep this process classically efficient. The operators in string representation are defined as

$$[A] \equiv \{(\mathcal{S}_j, c_j, p_j); j = 1, \dots, m\}, \quad (3.29)$$

where  $c_j$  is a real coefficient,  $p_j$  is a natural number enumerating an independent parameter, and  $\mathcal{S}_j = (S_{\sigma_j^x}, S_{\sigma_j^y}, S_{\sigma_j^z})$  where  $S_{\sigma_j^x}$ ,  $S_{\sigma_j^y}$ , and  $S_{\sigma_j^z}$  are the strings corresponding to the positions of the  $\sigma^x$ ,  $\sigma^y$ , and  $\sigma^z$  Pauli operators in  $\hat{a}_j^{\otimes N}$ , respectively. While  $c_j$  is the multiplicative factor to the parameter,  $p_j$  can be used to duplicate one or more parameters in the circuit. When all operators in the ansatz are required to have independent parameters,  $p_j = j$  for  $j = 1, \dots, m$ . In contrast, when only one operator has an independent parameter and all the others are dependent on it,  $p_j = 1$  for  $j = 1, \dots, m$ . Thus, the largest value enumerated by  $p_j$  is the total number of independent parameters in the circuit which need not necessarily be equal to the total number of parameters.

As an example, consider a set of operators for  $N = 3$  and  $m = 2$ ,

$$A = \{1.0 \times \sigma^y \otimes \sigma^y \otimes \sigma^I; 1.5 \times \sigma^y \otimes \sigma^I \otimes \sigma^z\}, \quad (3.30)$$

where one wishes to assign independent parameters to both the terms. The string representation has the terms

$$S_{\sigma_1^y} = (1, 2), \quad S_{\sigma_2^y} = (1), \quad S_{\sigma_2^z} = (3), \quad (3.31)$$

where the numbers in the parenthesis represent the location of the corresponding Pauli operator in the ansatz written from left to right (see Eq. 3.30). All other terms are non-existent. The coefficients are  $c_1 = 1.0$  and  $c_2 = 1.5$ . Thus, the representation is given by

$$[A] = \{(\mathcal{S}_1, c_1, p_1); (\mathcal{S}_2, c_2, p_2)\} = \{((0, (1, 2), 0), 1.0, 1); ((0, (1), (3), 1.5, 2))\}. \quad (3.32)$$

While a number in parenthesis inside  $\mathcal{S}_j$  gives the location(s), 0 denotes an empty string reflecting the absence of the corresponding term, e.g. no  $\sigma^x$  in  $\mathcal{S}_2$ . Note that a matrix representation of  $\exp(-iA(\theta))$  would be exponentially large in  $N$  but the string representation of its operators  $[A]$  has polynomial size. A sample input file is displayed in Appendix A.

### Processing

The bitstring representation contains all the information about an ansatz assuming the ordering of the operators is controlled using  $p$ , where the smaller values appear first. Before a circuit is constructed from the string representation, it is sometimes useful to perform certain operations. For example, an ansatz may benefit in a reduction of independent parameters by placing all the commuting operators together and assigning them a single parameter. This

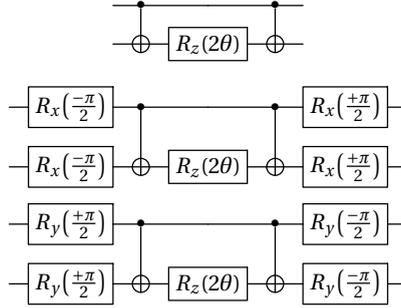


FIGURE 3.2: Circuits showing the implementation of  $e^{-i\theta\sigma^z\otimes\sigma^z}$  (top),  $e^{-i\theta\sigma^x\otimes\sigma^x}$  (middle), and  $e^{-i\theta\sigma^y\otimes\sigma^y}$  (bottom).

has the benefit that fewer number of independent parameters require less computational effort of the classical resources. Additionally, classical optimisation is NP-hard [BK21] and reducing the number of active parameters can help avoid the problem. An algorithm to group the commuting terms can be found in [RWS12]. A commonly used example of such grouping is the Hamiltonian variational ansatz [Wie+20; Ber+22; Lar+21; SY22].

Let us assume we have successfully grouped the  $m$  operators in an ansatz into  $\overline{m}$  groups, where  $1 \leq \overline{m} \leq m$ , and the left or right equality holds for the cases when all operators in the ansatz are commuting or non-commuting, respectively. The ansatz then becomes a set of these  $\overline{m}$  group(s) of operators, where all the terms in a group mutually commute. The next step is to exponentiate the operators to form unitary operators and convert them to a circuit. To understand the conversion, it is worthwhile to first consider a single term consisting of  $\sigma^z \otimes \sigma^z$ , whose exponentiation is given by  $\exp(-i\theta\sigma^z \otimes \sigma^z)$ . This operator is represented by the first circuit shown in Fig. 3.2 [WBAG11; NC10]. First, a CNOT gate is applied which entangles the two qubits, then a parametrised rotation in the  $\sigma^z$  basis, followed by a second CNOT gate. Varying the parameter  $\theta$  explores different superposition states of the Hilbert space and changes the energy of the system. The circuit is better understood by representing each gate in its matrix form. The matrix product of the gates is given by

$$\begin{array}{ccc} \text{CNOT} & R_z(2\theta) & \text{CNOT} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} e^{-i\theta} & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & e^{-i\theta} & 0 \\ 0 & 0 & 0 & e^{i\theta} \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} e^{-i\theta} & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 \\ 0 & 0 & 0 & e^{-i\theta} \end{bmatrix}. \end{array} \quad (3.33)$$

Now we can prepare circuits for other combinations of the Pauli operators by appropriate basis changes. To create the circuits for  $\exp(-i\theta\sigma^x \otimes \sigma^x)$  and  $\exp(-i\theta\sigma^y \otimes \sigma^y)$  we may use the circuit identities  $HZH = X$  and  $S^\dagger HZHS = Y$ , respectively [Lom03]. However, since the number of gate operations that can be performed on current and near-future quantum computers are limited, we prefer other identities that require fewer gates. The above identities could alternatively be achieved by appropriate  $R_x$  and  $R_y$  rotations. These are available on current generation superconducting type quantum computers [5-qc; 5-qb]. Using these identities, the circuit for  $\exp(-i\theta\sigma^x \otimes \sigma^x)$  is given by sandwiching  $\exp(-i\theta\sigma^z \otimes \sigma^z)$  between  $R_x$  gates. The second and last circuits in Fig. 3.2 show the implementation of  $\exp(-i\theta\sigma_1^x\sigma_2^x)$

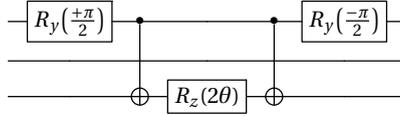


FIGURE 3.3: Circuit showing the evolution of  $e^{-i\theta\sigma_1^y\sigma_2^y\sigma_3^z}$ .

and  $\exp(-i\theta\sigma_1^y\sigma_2^y)$ , respectively. Knowing the circuit implementation of  $\exp(-i\theta\sigma^\alpha \otimes \sigma^\beta)$  for  $\alpha, \beta \in \{x, y, z\}$  and the identities, we can create a quantum circuit for any arbitrary combination of terms. For example, the second term  $S_2$  in the example given in Eq. (3.30) translates to the circuit shown in Fig. 3.3.

### Output

Although the algorithm described until now can be used to instruct a quantum computer to run a circuit directly, it is preferred that the output of the algorithm be stored in a commonly accessible format. Access to quantum computers is expected to remain cloud based, as currently offered by IBM [Ale+], Rigetti [Kar+20], or others, in the near future. Additionally, cloud based service providers often do not provide "direct" access to a quantum computer but require an intermediate circuit execution format which can be checked for consistencies before it is run. To this end, we choose to convert the output to the standard open quantum assembly language using Qiskit [Ale+]. A sample output is shown in Appendix A.

### 3.3.2 Problem Hamiltonian

Similar to the expression of  $N$  qubit ansatz given in Eq. (3.25), the Hamiltonian of a given problem having  $m$  terms can be expressed as

$$\hat{H}^{\otimes N} = \sum_{j=1}^m c_j h_j^{\otimes N}, \quad h_j^{\otimes N} = \otimes_{l=1}^N \hat{\Theta}_{jl}. \quad (3.34)$$

Following the same procedure as in section 3.3.1, the Hamiltonian can also be expressed in a string format

$$[H] = [\hat{H}^{\otimes N}] \equiv \{(S_j, c_j^r, c_j^*); j = 1, \dots, m\}, \quad (3.35)$$

where  $c_j^r$  and  $c_j^*$  are the real and imaginary parts of the complex coefficient, respectively. Note again that the matrix representation of  $\hat{H}^{\otimes N}$  requires exponentially increasing classical memory to store while  $[H]$  has polynomial size. The index  $j$  can be used to group terms together.

### Connection to circuit preparation

To calculate the expectation value of the Hamiltonian it is necessary that the final state of the quantum computer is measured in the appropriate basis. The relevant basis for each term is defined in Eq. (3.34). The circuit preparation algorithm that creates the ansatz can read the Hamiltonian input file and suitably change the basis after the end of the computation but before the measurement. A sample input file for the Hamiltonian is shown in Appendix A. There are essentially four different types of operators in  $H$  each of which can be measured as follows

1.  $\sigma^x$ : add an  $R_y(\pi/2)$  gate at the end of the circuit,
2.  $\sigma^y$ : add an  $R_x(\pi/2)$  gate at the end of the circuit,
3.  $\sigma^z, \sigma^I$ : no need for any additional gates.

The indices of the qubits on which the gates should be added is given by the index  $l$  in Eq. (3.34). Thus, the circuit preparation algorithm reads each string of each term in  $[H]$  and adds a relevant gate on which a non-identity operator exists.

### Efficiency of measurements

The total energy of  $H$  is the sum of energies of the individual terms multiplied with their coefficients. The total number of strings the circuit preparation algorithm processes (at most) is  $\propto N \times m$ , the latter of which ought not to be scaling exponentially to remain classically efficient. For example, the number of distinct Pauli strings in chemical Hamiltonians scales as  $N^4$  for large molecules [Cer+21]. If the problem that the number of samples required to measure each term within a certain accuracy (which can be impractically large [WHT15]) is not an issue, several techniques have been proposed to reduce the total number of measurement terms, as discussed below.

An intuitive solution is to measure multiple terms at once which pertains to the partitioning of the Pauli operators into sets whose members mutually commute. The task of partitioning is equivalent to combinatorial problems of graph colouring [JGM19] or finding a minimum clique cover [VY120; Izm+20; Zha+20b]. A simple and easy to implement solution, which may not give the best grouping, is to enumerate the strings and create sets of those that commute on the relevant index per term.

Several other solutions have been summarised in [Cer+21]. One solution is to optimise the samples per term by considering the weight of the coefficient of that term. If the coefficient is small it will contribute less to the overall variance. Other solutions explore *classical shadows* [HKP20] or *neural network tomography* [Tor+20].

### Exact diagonalisation

The ground state energy can be calculated for small systems using exact diagonalisation of the Hamiltonian matrix. The same string representations can be used to streamline the process with the circuit preparation algorithm. The string representation  $[H]$  contains all the information about  $H$ . Once input to a classical computer, one can create the matrix from  $[H]$ , at least for small systems. This matrix can be diagonalised *exactly* within machine precision to calculate the eigenvalues and eigenvectors. An exact diagonalisation is helpful for benchmarking and verifying variational methods on small systems.

Beyond small-size systems, the construction, as well as the diagonalisation of the matrix, quickly become intractable for classical computers. Despite the exponential growth of the matrix and, therefore, the inability of classical computers to solve for the eigenvalues, it is worthwhile to mention that certain methods have been developed that have extended the ability of exact diagonalisation by reducing resource consumption. Two such methods are the Lanczos method and the Jacobi-Davidson method. While exact diagonalisation relies on matrix transforming techniques, these methods are iterative [VG96]. Note that sometimes the Lanczos method is also called exact diagonalisation in the literature. The Lanczos method can be found in [CW02]. The Jacobi-Davidson method was first suggested in [SV00] by combining Davidson's method [Dav75] with Jacobi's procedure [Jac46]. For a comparison of the two methods, we refer to [WF08].

### 3.3.3 Initial state

An important ingredient for variational methods is the initial state of quantum computers. Often in variational methods, the initial state is variedly defined. On one side, the initial state can mean only one of the  $2^N$  possible combinations of computational states in which the quantum computer can be prepared. On the other side, it can also mean any efficiently preparable initial state (requiring several gates) for which the energy estimate is closer to the ground state than any random point on the landscape. In this work, the initial state mostly refers to the latter cases. It is possible that the two cases coincide (see section 4.1) or do not coincide (see section 4.3).

The combination of the ansatz and the problem Hamiltonian gives rise to a rugged multi-dimensional landscape whose global minimum (ideally) corresponds to the ground state of the problem. The dimension of the landscape is equal to the number of independent parameters in the circuit. It is possible that for a wrong choice of an ansatz, the global minima of the landscape do not correspond to the ground state. For example, in the case of the trial wavefunction of the harmonic oscillator given in Eq. (3.10) where if  $p$  is the parameter, the landscape is convex, and the global minimum is easy to calculate. We saw in that example that the global minima did not correspond to the ground state energy, representing a bad choice of ansatz. Since we are primarily interested in the energy, its estimate calculated from variational methods for multi-dimensional landscapes can be significantly above the ground state energy.

Traversing the rugged energy landscapes is a difficult problem. It has been shown that classical optimisation is *intrinsically* hard, for variational methods NP-hard, and does not merely inherit the hardness from the ground state problem [BK21]. Thus, finding the global minima of the landscape from an arbitrary starting point is fraught with problems. One such problem is the existence of *barren plateaus* in the energy landscapes [McC+18; CNB21; Hol+22] (see also section 5.2.2). While efforts are underway to avoid barren plateaus [Pat+21; ZG21], it remains an open problem. Additionally, there may be a large number of local minima surrounding and prohibiting accessing the global minimum starting from an arbitrary point.

The above mentioned problems can be avoided by preparing good initial states. However, it is not in general clear what the good initial states should be for any given problem. For some problems, classical approximations can offer some insight but need not necessarily always help. It is an open question if such initial states can be known for any given problem in advance. For the problems we test, there exist initial states that offer a better initial energy than a random point on the energy surface.

### 3.3.4 Classical calculations

The role of the classical subroutine is twofold: energy calculation and parameter optimisation. In this section, we discuss the former. The subroutine run on a quantum computer gives after measurement one bitstring at a time. This output is stored on a classical computer. Since the quantum theory is probabilistic, the quantum computer must be run several times to build the statistics required for a meaningful computation. Furthermore, interesting problem Hamiltonians often consist of more than one term that does not necessarily commute amongst others. Not all the terms can be measured at once in such cases. We are interested in calculating the expectation value of a Hamiltonian of the type given in Eq. (3.34). For simplicity of explanation, we assume that each term  $h$  is measured on the corresponding basis separately. The process can be readily generalised to grouped terms. The procedure to compute the expectation value is as follows.

### Energy calculation

Assume that the parametric optimisation of an ansatz has prepared the state  $|\Psi\rangle$  on a quantum computer. Let  $s$  samples be obtained on the relevant basis of the term on the ground state. The output corresponding to each measurement is accumulated in a classical computer. Assume that the sequential preparation and measurement of the state  $|\Psi\rangle$  several times gives a (random) sequence of  $N$ -length bitstrings

$$\begin{aligned} r_0 &= [010\dots 01] \\ r_1 &= [110\dots 00] \\ &\vdots \\ r_{s-1} &= [100\dots 11] \end{aligned} \tag{3.36}$$

where  $r_i$  is the  $i$ -th bitstring produced when sampling  $s$  times. The data collected from the quantum computer can be stored in a matrix  $R$  of size  $N \times s$ , where each element is a classical bit 0 or 1,

$$\begin{array}{c} r_0 \\ r_1 \\ \vdots \\ r_{s-1} \end{array} \rightarrow \begin{array}{c} [010\dots 01] \\ [110\dots 00] \\ \vdots \\ [100\dots 11] \end{array} \Bigg]_{N \times s} = R. \tag{3.37}$$

It is noteworthy that the total number of elements in the matrix is proportional to  $s$ , which should (for practical applications) scale polynomially with  $N$ . Thus, the size of  $R$  is not exponential in  $N$ , in contrast to the matrix representation of the Hamiltonian. Additionally, storing the sequential outputs is not necessary as each bitstring can be processed as soon as it is measured (see section 5.5). However, for simplicity of explanation, the data is represented by the matrix in the rest of the section.

To compute the expectation value, we need to calculate the unknown non-zero coefficients of the wavefunction given in Eq. (2.5). It is assumed that the collected bitstrings have been measured in the relevant Pauli basis prescribed in the Hamiltonian. Recall that all the Pauli operators  $\sigma^\alpha$  have the eigenvalues 1 and  $-1$ . We adopt the convention  $\sigma^\alpha |1\rangle = -|1\rangle$ . We compute the parity of each bitstring given by each row of  $R$  by multiplying the eigenvalues of each individual bit in the string corresponding to the operator in the term. To do so, it is relevant to count the number of 1s in each bitstring against the presence of a Pauli operator in the Hamiltonian term. The conditional product giving the parity of the bitstring is given by

$$P_s = \prod_{i=0}^{N-1} \delta_s(r_i, h_i), \tag{3.38}$$

where the index  $i$  enumerates each bit of the bitstring  $r$  and Hamiltonian string  $[h]$ , and  $\delta(r_i, h_i) = -1$  only if  $h_i \in \{\sigma^x, \sigma^y, \sigma^z\}$  and  $r_i = 1$ , and 1 otherwise. The list of possibilities are tabulated below.

$h_i$	$r_i$	Parity
$\sigma^\alpha$	1	-1
$\sigma^\alpha$	0	1
$\sigma^I$	1	1
$\sigma^I$	0	1

By enumerating through each bit in the string, the parity of an individual bitstring is given

by Eq. (3.38). For example, the parity counting operation on  $r_0$  given in Eq. (3.37), assuming a Hamiltonian term  $\sigma_0^x \sigma_1^x$ , gives

$$\begin{aligned} P_{r_0} &= \sigma^{\otimes N} |010\dots 01\rangle = \sigma_{N-1}^I |0\rangle \sigma_{N-2}^I |1\rangle \sigma_{N-3}^I |0\rangle \dots \sigma_1^x |0\rangle \sigma_0^x |1\rangle \\ &= (+1) \times (+1) \times (+1) \times \dots \times (+1) \times (-1) \\ &= -1. \end{aligned} \quad (3.39)$$

Thus, the parity of an individual bitstring is given by the product of parities of the individual bits in the string as calculated for the given Hamiltonian term operators. The parities of  $s$  bitstrings are collected in the vector

$$\mathcal{P} = P_0, \dots, P_{s-1}. \quad (3.40)$$

By enumerating all bitstrings  $r_k$  for  $k = 0, \dots, s-1$  taken from the quantum computer, we can write the expectation value of one term  $h_j$  of  $H$  in terms of the measured frequencies (bitstrings) such that

$$\begin{aligned} \langle c_j h_j \rangle &= c_j \frac{\langle \Psi | h_j | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \frac{c_j (\langle r_0 | \sigma^{\otimes N} | r_0 \rangle + \langle r_1 | \sigma^{\otimes N} | r_1 \rangle + \dots + \langle r_{s-1} | \sigma^{\otimes N} | r_{s-1} \rangle)}{s} \\ &= \frac{c_j (\mathcal{P}_0 + \mathcal{P}_1 + \dots + \mathcal{P}_{s-1})}{s} \\ &= \frac{c_j (\sum_{i=0}^{s'-1} \mathcal{P}_i \mathcal{G}(r_i, R))}{s}, \end{aligned} \quad (3.41)$$

where  $\mathcal{G}(x, y)$  is the frequency of reoccurrences of bitstring  $x$  in the rows of matrix  $y$ ,  $s' \leq s$  is the number of unique bitstrings in  $R$ , and  $c$  is the coefficient described in Eq. (3.34). Note that by blindly enumerating  $s'$  from 0 to  $2^{N-1}$  (all possible states) for the purpose of counting, we add an exponential overhead which should be avoided by enumerating each row (all measured states) in  $R$  instead which has only a polynomial number of entries.

The procedure until now can calculate the energy of a single term. Let  $\mathcal{R} = R_1, R_2, \dots, R_m$  be the list of matrices corresponding to each term  $j$ . The expectation value of the entire Hamiltonian is found by adding those of the individual ones,

$$\begin{aligned} \langle H \rangle &= \langle c_1 h_1 \rangle + \langle c_2 h_2 \rangle + \dots + \langle c_m h_m \rangle \\ &= \sum_{j=1}^m c_j \langle h_j \rangle \\ &= \sum_{j=1}^m \frac{c_j \sum_{i=0}^{s'_j-1} \mathcal{P}_{i,j} \times \mathcal{G}(r_{i,j}, R_j)}{s_j}. \end{aligned} \quad (3.42)$$

### 3.3.5 Error resilience

Since variational quantum algorithms are designed for quantum computers that will be prone to errors, it is expected that they are tolerant of them to some extent. In this section, we discuss briefly how such algorithms have an inherent ability to be resilient against certain types of errors.

The first defence of variational methods against errors is the construction of a special ansatz for particular types of problems. Such construction requires previous knowledge about the problem. For example, problems comprising spins on a lattice or electrons in a molecule

often conserve the total number of spins or electrons. Due to the conservation of a certain property of the system under study, it is possible to design an ansatz that respects the conservation. To preserve the total number of particles in a system, a so-called *number preserving* ansatz [Con+22; Kir+22] has been proposed. The number of particles in such an ansatz is conserved for all possible values of  $\theta$ . The preservation is not necessarily limited to particle numbers but can be extended to other symmetries [Gar+20] and has been demonstrated in prototype applications [Omi+22]. A bit flip error can be detected since the total number of  $|0\rangle$ s and  $|1\rangle$ s in the state  $|\psi\rangle$  after the end of the computation will not match with those at the beginning. Thus, all final bitstrings output by the quantum computer that do not conserve an assumed symmetry can be discarded.

The error resilience of variational algorithms is also through the parametrisation. Let an ansatz of an arbitrary product of unitary operators be given by  $\mathbb{U}(\theta)$  act (in theory) on an initial state  $|\psi_0\rangle$ . Assume that an error prone quantum computer instead performs  $\mathbb{U}'(\theta)|\psi_0\rangle$ , where  $\mathbb{U}'(\theta) \neq \mathbb{U}(\theta)$ . Such an error is variationally suppressible if there exists a correction vector  $\beta$  such that

$$\|\mathbb{U}(\theta) - \mathbb{U}'(\theta')\| < \epsilon, \quad \theta' = \theta + \beta, \quad (3.43)$$

for a given  $\epsilon > 0$  and if the corrected vector  $\theta' = \theta + \beta$  corresponds to an optimum on the optimisation landscape [McC+16]. No additional knowledge about an Hamiltonian or an ansatz is required if the condition given in Eq. (3.43) is met. In some troublesome cases the error resilience through parametrisation may not work (see [McC+16] for examples and remedies).

Lastly, even if Eq. (3.43) does not work, it is still possible to approximate the ground state energy. Inherent to the variational methods is the ability to use any ansatz to find the ground state energy. Even if the quantum computer performs  $\mathbb{U}'(\theta)$  instead of  $\mathbb{U}(\theta)$ , it is a valid ansatz in itself albeit not the intended one. Therefore, although a mapping from  $\mathbb{U}(\theta) \rightarrow \mathbb{U}'(\theta')$  may remain unknown, a quantum computer can still be used to approximate the ground state energy.

## 3.4 Classical optimisation

An integral part of the variational methods is the parameter optimisation that is carried out on a classical computer. All the methods we discuss in this section are run on classical computers because they are not hard or intractable for them. We explore the types of optimisation algorithms that are suitable for the hybrid variational methods described until now. Although the majority of the work in this dissertation is concerned with using the energy scalar that is input to the optimisation algorithms, we keep the discussion general by using a function  $f$  as the scalar generator. Parts of this section are based on optimisation algorithms as discussed in [NW06].

### 3.4.1 Mathematical foundations

Optimisation consists of minimising or maximising a given  $f$ . We assume that

- $x$  is a real vector of parameters – corresponding to each unique parameter in the quantum circuit; and
- $f$  is the objective function with the argument  $x$ , i.e.  $f(x)$ , which outputs a scalar quantity to be minimised or maximised.

Often a third assumption is made that defines the constraints on  $x$ . However, such an assumption is not necessary for the work considered in this dissertation. The optimisation

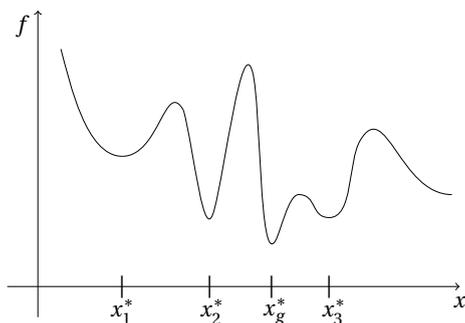


FIGURE 3.4: An illustrative one-dimensional optimisation landscape. There are three local minima, namely  $x_1^*$ ,  $x_2^*$  and  $x_3^*$ , and one global minimum  $x_g^*$ .

problem in the above notation is written as

$$f_{\min} = \min_{x \in \mathbb{R}^m} f(x), \quad (3.44)$$

where  $x$  is a real vector in  $\mathbb{R}$  and  $f: \mathbb{R}^m \rightarrow \mathbb{R}$  is assumed to be a smooth function. The optimisation to maximise  $f$  can be done by minimising  $-f$ , and vice versa. In this work, the main interest would be finding the ground state energy of various physical models and, therefore, the minimisation of  $f$ .

To achieve an optimisation objective, the algorithm should yield a *global optimum* of  $f$  which is a point  $x_g^*$  such that

$$f(x_g^*) \leq f(x) \text{ for all } x,$$

where the domain of  $x$  is  $\mathbb{R}^m$ . In practice, however, finding a global optimum is a difficult problem since the knowledge of  $f$  is usually only local. Generally, in practice one is also limited by the total number of function evaluations that can be performed. Most algorithms find a *local optimum* which is  $x^*$  such that

$$f(x^*) \leq f(x) \text{ for all } x \in \mathcal{N}(x), \quad (3.45)$$

where  $\mathcal{N}(x)$  is the neighbourhood of  $x^*$ . For example, Fig. 3.4 illustrates a difficult single parameter case for global optimisation. There are three local minima and one global minimum. An algorithm that only goes downhill and does not start in the neighbourhood of  $x_g^*$  will not find  $x_g^*$ . The local minima illustrated in Fig. 3.4 are a problem for algorithms in general. For variational algorithms, the knowledge of a suitable initial state combined with initial parameters can allow starting in  $\mathcal{N}(x)_g$ . In general, however, such knowledge may not be readily available or computationally feasible. Even if some information is available, it is in general difficult to ascertain if  $\mathcal{N}(x)$  is  $\mathcal{N}(x)_g$  in multi-dimensional landscapes. Functions for which no prior knowledge about their landscape can be made available are called *black-box* functions. In general, functions in variational methods are black-box type.

### Taylor's theorem to recognise an optimum

Given a black-box function, it might appear that the only way to ascertain that a given  $x$  is a local minimum  $x^*$  is to check the value of  $f(x)$  for each  $x$  in the neighbourhood  $\mathcal{N}(x)$ . Only when the Eq. (3.45) is satisfied for all  $x \in \mathcal{N}(x)$  can we make the claim that  $x^*$  is a local minimum. In general, this process requires enormous computational effort and quickly

becomes intractable if  $f$  is continuous and/or difficult to compute. However, if we assume that  $f$  is *smooth*, there are more efficient ways to determine a local minimum than checking for every possible point. This section outlines Taylor's theorem, which will help us achieve that efficiently.

**Theorem 1** *Taylor's theorem.*

Let  $f: \mathbb{R}^m \rightarrow \mathbb{R}$  be continuously differentiable and let  $p \in \mathbb{R}^m$ . Then we may write

$$f(x+p) = f(x) + \nabla f(x+tp)^T p, \quad (3.46)$$

for some  $t \in (0,1)$  where the superscript  $T$  in  $A^T$  implies the transpose of  $A$ . If  $f$  is further assumed to be twice continuously differentiable, we have that

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp) p dt, \quad (3.47)$$

and that

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+tp) p. \quad (3.48)$$

Thus, the Taylor series truncated at the second term allows us to approximate a given black-box function with a quadratic polynomial. From Eq. (3.48), it is immediately apparent that at the points  $x$ , by substituting  $p = \mathbf{0}$  we get the original value of the function, and the quadratic term approximates  $f$  for a small perturbation around  $x$ . The perturbation will define the neighbourhood  $\mathcal{N}(x)$ , and as we will see, there are efficient ways to recognise the local minimum of  $\mathcal{N}(x)$  for this approximation. Thus, if our approximation to the function is reasonable, we would have efficiently recognised the local minimum of the black-box function around  $x^*$ .

The necessary and sufficient conditions to recognise  $x^*$  as a local minimum are derived by assuming that the point is a local minimum and then proving facts about the first and second-order derivatives of  $f$ . We look at the following theorems to find the local minimum of the quadratic approximation. Here we only state the theorems; the proofs can be found in [NW06].

**Theorem 2** *First-order necessary condition: If  $x^*$  is a local minimum and  $f$  is continuously differentiable in a neighbourhood  $\mathcal{N}(x^*)$ , then  $\nabla f(x^*) = 0$ .*

**Theorem 3** *Second-order necessary condition: If  $x^*$  is a local minimum and  $\nabla^2 f$  exists and is continuous in  $\mathcal{N}(x^*)$ , then  $\nabla f(x^*) = 0$  and  $\nabla^2 f(x^*)$  is positive semidefinite.*

**Theorem 4** *Second-order sufficient condition: If  $\nabla^2 f$  is continuous in  $\mathcal{N}(x^*)$ ,  $\nabla f(x^*) = 0$  and  $\nabla^2 f(x^*)$  is positive definite, then  $x^*$  is a strict local minimum of  $f$ .*

We use these theorems for our purpose as follows. Theorem 1 tells us that we can replace our unknown  $f$  with a quadratic approximation. By doing so, we avoid having to check each point in  $\mathcal{N}(x)$ . Theorem 2 tells us that if  $f$  is smooth, the first-order derivatives of a local minimum  $x^*$  vanish. Theorem 3 goes further and tells us that if the second-order derivative exists, it is positive semidefinite. Theorem 4 tells us that if all the conditions stated in the previous theorems are met, then  $x^*$  is a strict local minimum of  $f$ . A minimum is called 'strict' when the equality sign is removed from Eq. (3.45). Thus, the process of finding the local minimum of a black-box function involves assuming that the given function is smooth and approximating it with a quadratic polynomial. The local minimum of this approximated

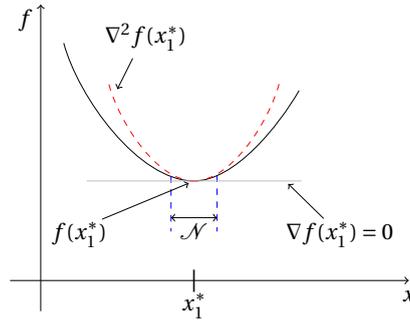


FIGURE 3.5: A quadratic Taylor approximation to the arbitrary function around the local minimum at  $x_1^*$  shown in Fig. 3.4. The dashed red curve showing the approximated function is in good agreement with  $f$  in the region  $\mathcal{N}(x)$ . The dashed vertical blue lines show the neighbourhood  $\mathcal{N}(x)$  around  $x$ .

function is "global" by construction. In other words, the global minimum of the Taylor's approximation of  $f$  is the local minimum of the black-box function  $f$  in the neighbourhood  $\mathcal{N}(x)$ .

For example, consider the local minimum at  $x_1^*$  from Fig. 3.4 shown again in Fig. 3.5. The three terms on the right hand side of Eq. (3.48) are shown for  $x_1^*$ . The value of  $f$  at  $x_1^*$  is the first term in Eq. (3.48), the first-order derivative of  $f$  is the second term, and the second-order derivative is the last term. The quadratic approximation in the neighbourhood  $\mathcal{N}(x)$  is shown by the dashed red curve. The value  $\nabla f = 0$  only at  $x_1^*$ . For any  $x \neq x_1^*$ ,  $\nabla f \neq 0$ . Thus, for this example, the local minimum is found when  $\nabla f = 0$ . The idea is readily extended to more than one dimension.

In summary, one way or another, any optimisation algorithm seeks to find that value of  $f$  where  $\nabla f(x) = 0$ . This explains how gradient based algorithms get their name; as they seek to find a zero gradient. It was assumed that the function is smooth; however, if it is not smooth, this technique will not work. An introduction to methods for non-smooth functions can be found in [HUL93a; HUL93b].

### 3.4.2 Gradient based algorithms

Optimisation algorithms come in all forms and flavours and it is difficult to fit them all in a few categories. A large number of algorithms can still be categorised broadly in terms of gradient based or gradient free. Our first aim is to find out if variational algorithms can work for ideal quantum computers emulated by a classical computer. For the optimisation part, we will rely heavily on gradient based methods as they offer a quicker convergence for noise-free function evaluations. Only if the variational algorithm works for this ideal quantum computer, it makes sense to move onto mimicking real hardware. On an actual quantum computer, the function evaluation is stochastic which makes computing the gradient unreliable with the usual numerical formulas. We will discuss techniques to compute gradients when such functions are involved if certain conditions about the ansatz are met. In general, however, gradients cannot always be reliably computed for such functions; different gradient free methods for them are reviewed in [RS13; LMW19].

### The process

All optimisation algorithms require an initial point  $x_0$ . If some information about  $f$  is available, then  $x_0$  can greatly help in reaching the global minimum when  $\{x_0, x_g^*\} \in \mathcal{N}(x)_g$ . Since for black-box functions no prior information is available,  $x_0$  is often initialised at random. Beginning at  $x_0$ , algorithms move forward in a sequence of iterates at points  $x_k$  for  $k = 0, 1, 2, \dots, \infty$  that terminates either when no more progress is made or when a certain threshold for the number of iterations is reached. To decide the next iterate  $x_{k+1}$  given a previous iterate  $x_k$ , an algorithm uses the information about  $f$  at  $x_k$ . Some algorithms might also use the information from all or some previous iterates  $x_0, x_1, \dots, x_{k-1}$ , e.g. Bayesian optimisation [Den+20]. The sole purpose of an algorithm is to find those next iterates that decrease the value of  $f$  as the value of  $k$  increases. Thus, an algorithm proceeds systematically to find a new iterate  $x_{k+1}$  such that  $f(x_{k+1}) < f(x_k)$ .

### Line search and trust region

Once it is known that we seek to find  $x$  such that  $\nabla f(x) = 0$  starting from  $x_0$ , the next question arises naturally: how do optimisation processes proceed to find that  $x$ ? There are two broad categories of gradient based algorithms that answer this question: *line search* and *trust region*. In line search, an algorithm has to choose a direction of search  $p_k$  and then search along that direction until a new iterate  $x_{k+1}$  is found that lowers the value of  $f$ . The distance to move in the direction  $p_k$  is found by approximately solving to find a step length  $\alpha$ :

$$\min_{\alpha > 0} f(x_k + \alpha p_k). \quad (3.49)$$

The exact solution of Eq. (3.49) would give the maximum benefit from direction  $p_k$ , however, an exact optimisation may be prohibitively expensive. Instead, a limited number of trial step lengths may be chosen until the algorithm finds one that roughly approximates the minimum of Eq. (3.49). The choice may depend on the current iterate, e.g.  $\alpha = \alpha(k)$ . At the next iterate, a new search direction is chosen and step length is calculated. The process is repeated until convergence ( $\nabla f \rightarrow 0$ ).

In the trust region category, information about  $f$  is used to construct a model function  $m_k$  which behaves similar to  $f$  in a certain 'trust region' around  $x_k$ . As the quality of  $m_k$  to mimic  $f$  deteriorates when  $x$  is far from the current iterate  $x_k$ , the search for a minimiser of  $m_k$  is restricted to some smaller region around  $x_k$ . The algorithm approximately solves the following subproblem to find the best candidate step  $l$ :

$$\min_l m_k(x_k + l), \quad (3.50)$$

where  $x_k + l$  lies inside the trust region. If a candidate solution produces a sufficient decrease in  $f$ , the solution is accepted and the next iterate is found.

### The next iterate

Using the line search method, given a current iterate  $x_k$ , the next iterative  $x_{k+1}$  is given by

$$x_{k+1} = x_k + \alpha_k p_k. \quad (3.51)$$

The first task is to find a direction  $\mathbf{p}_k$  that will decrease  $f$ , or a *descent* direction. The most commonly used descent directions can be categorised as

$$\mathbf{p}_k = \begin{cases} -I\nabla f_k & \text{gradient,} \\ -(\nabla^2 f_k)^{-1}\nabla f_k & \text{Newton,} \\ -B_k^{-1}\nabla f_k & \text{quasi-Newton,} \end{cases} \quad (3.52)$$

where  $I$  is the identity matrix,  $\nabla^2 f_k$  is the Hessian matrix, and  $B_k^{-1}$  is a matrix approximating the Hessian. While the gradient direction is the most cost effective to calculate, it is also the slowest: offering only a linear convergence. On the other hand, the Newton direction offers quadratic convergence (see [NW06] for proof). The disadvantages of Newton's direction are that it requires (a) very large number of (potentially costly) function evaluations and (b) inverting the Hessian matrix, at every iteration. The quasi-Newton direction meets in the middle by offering a convergence better than gradient descent but requiring only the first order gradient information. These quasi-Newton methods iteratively update the Hessian inverse approximation  $B_k^{-1}$  as the optimiser traverses the landscape to obtain the information about the curvature beyond a linear approximation. Quasi-Newton methods are the optimal choice for variational methods due to their low number of function evaluations as well as usage of second order information.

After the selection of a direction, a suitable step size  $\alpha_k$  is required at the iterate  $x_k$ . A trivial choice is  $\alpha_k = 1$ , but it may not always work if  $x_k$  is far away from the minimum. In contrast, the trivial choice may actually increase the value of  $f$ . To solve the problem we *backtrack* along the direction by decreasing the value of  $\alpha_k$  until we find a sufficient decrease in  $f$  [Pre+07; Wol69]. A potential direction of research is to find a method for inexact line search so that the expensive to calculate the gradient for variational methods is used most optimally.

### 3.4.3 Calculating gradients

We learnt from the previous section that calculating the gradient is an important task for optimisation algorithms. There are two broad ways of calculating gradients, as described below.

#### Numerical gradients

When an analytical form of the function is not available (black-box), the most commonly used method is the finite difference approximation. Different types of finite difference formulas are available for numerical differentiation. While the finite difference formulas work well for noise-free functions, they often fail on quantum computers due to the stochastic nature of the function. We will discuss remedies to this problem later. The calculation of gradients is carried out numerically using forward, backward, or central finite difference formulas. Below we describe the gradient computation for a single parameter but it is readily generalised by replacing  $x_k$  with a vector  $\mathbf{x}_k$ . The equation for the gradient at  $x_k$  is given by definition

$$\nabla f(x_k) = \lim_{h \rightarrow 0} \frac{f(x_k + h) - f(x_k)}{h}. \quad (3.53)$$

Since  $f$  is a black-box function, the above expression is not available analytically. The gradient is then computed using the finite differences formula

$$\nabla f(x_k) = \frac{f(x_k + h) - f(x_k)}{h} + C, \quad (3.54)$$

where  $C$  is the difference between the true value of the gradient and the value given by the approximation of the finite differences formula. For a smooth function,  $C$  can be made arbitrarily small by decreasing  $h$ . Assuming that  $C$  is sufficiently small, the finite differences formula approximates the gradient as

$$\nabla f(x_k) \approx \frac{f(x_k + h) - f(x_k)}{h}, \quad (3.55)$$

where for our numerical purposes  $h = 10^{-8}$  is sufficiently small. While the central difference formula approximates the gradient more accurately, we do not use it due to the overhead in function evaluations. Using the forward difference formula the number of functional evaluations per iteration is equal to the number of parameters in the circuit plus one.

When a function is evaluated on an emulator of a quantum computer which uses the wavefunction, the output energy is numerically *exact*. In other words, the underlying function is a proper mathematical function. For two sets  $\mathcal{A}$  and  $\mathcal{B}$ ,  $f$  is defined by a set of ordered pairs  $(a, b)$  where  $a \in \mathcal{A}$  and  $b \in \mathcal{B}$ , and every element of  $\mathcal{A}$  maps to exactly one element in  $\mathcal{B}$ . In this case, the gradient can be reliably computed using Eq. (3.55).

When a function is evaluated on a quantum computer or an emulator of a quantum computer (by sampling), even though the underlying function is proper, due to sampling from it the output energy is *not* exact. Such a computer does not output the wavefunction but measurables or samples (bitstrings). For a finite number of samples, the apparent underlying function is then not a proper mathematical function but a stochastic function  $f'$  such that

$$f'(x) = f(x) \pm \epsilon(x), \quad (3.56)$$

where  $\epsilon(x)$  is a stochasticity variable. The energy from the function  $f'$  becomes exact in the asymptotic limit of infinite measurables, where  $\epsilon(x) \rightarrow 0$ . In practice, however, only a limited number of measurables are taken from a computer and  $\epsilon(x) \neq 0$ . In this case, in general, the gradient cannot be reliably computed using Eq. (3.55). However, in our case, since we can reduce  $\epsilon$  by increasing the number of samples, it is possible to estimate the gradient to some extent.

The stochastic nature of  $f'$  implies that Eq. (3.55) is affected by an additional stochastic variable  $\mathcal{E}$  as

$$\nabla f'(x_k) \approx \frac{f(x_k + h) - f(x_k) \pm \mathcal{E}(k)}{h}, \quad (3.57)$$

where  $\mathcal{E}(k) > 0$  allows for the case that the variable may take different values at each iteration. Let us define

$$\Delta f_k = |f(x_k + h) - f(x_k)|. \quad (3.58)$$

To make use of the finite difference formula in this case, it is required (see also appendix of [Kan+17]) that

$$\Delta f_k \gg \mathcal{E}(k). \quad (3.59)$$

We know for our cases that  $\mathcal{E}$  is inversely proportional to the number of samples. It is not required that  $\mathcal{E}$  vanishes, but only that it reduces in value such that Eq. (3.59) becomes valid. If the left hand side of Eq. (3.59) is much larger than the right hand side, we get the finite difference approximation, leading to a descent direction for the gradient descend. Otherwise, if the right hand side is too large ( $\Delta f_k \ll \mathcal{E}$ ) or comparable ( $\Delta f_k \approx \mathcal{E}$ ), the gradient computation will fail to give a good approximation and the search direction will obey the stochastic fluctuations of  $\mathcal{E}$ , leading to no progress in the optimisation. A drawback is that a very large number of samples may be required to fulfil Eq. (3.59) when  $\Delta f_k \rightarrow 0$ , which is the case near

a local minimum.

### Analytical gradients

Consider a general approach where each unitary operator has the form

$$U_i(\theta_i) = \exp(-i\theta_i A_i) \quad \text{for } i = 1, \dots, m, \quad (3.60)$$

where  $A_i$  is some Pauli operator. We know that the energy at each iteration  $x_k$  is

$$E_{x_k} = \langle \psi_0 | U_1^\dagger(\theta_1) \dots U_m^\dagger(\theta_m) H U_m(\theta_m) \dots U_1(\theta_1) | \psi_0 \rangle, \quad (3.61)$$

where  $x = \{\theta_1, \theta_2, \dots, \theta_m\}$ . The task is to compute the gradient of  $E_{x_k}$  in each direction  $e_i$  for  $i = 1, 2, \dots, m$ . The partial derivative in a certain direction  $e_i$  is

$$\begin{aligned} \frac{\partial}{\partial \theta_i} E(\theta_1, \dots, \theta_m) &= -i \langle \psi_0 | U_1^\dagger(\theta_1) \dots U_m^\dagger(\theta_m) H U_m(\theta_m) \dots A_i U_i(\theta_i) \dots U_1(\theta_1) | \psi_0 \rangle \\ &\quad + i \langle \psi_0 | U_1^\dagger(\theta_1) \dots U_i^\dagger(\theta_i) A_i \dots U_m^\dagger(\theta_m) H U_m(\theta_m) \dots U_1(\theta_1) | \psi_0 \rangle \\ &= -2\text{Im} \langle \psi_0 | U_1^\dagger(\theta_1) \dots U_m^\dagger(\theta_m) H U_m(\theta_m) \dots A_i U_i(\theta_i) \dots U_1(\theta_1) | \psi_0 \rangle. \end{aligned} \quad (3.62)$$

It is assumed that  $A_k = A_k^\dagger$ . We can use an auxiliary qubit to calculate the expression in Eq. (3.62). The gradient in a certain direction can be computed by making repeated measurements on the auxiliary qubit where the circuit depth is  $2m + 2$  as described in [FN18]. Avoiding use of an auxiliary qubit is desirable for emulators because using an additional qubit doubles the memory requirements for a classical computer. Analytical gradients will be helpful where numerical gradients cannot be reliably calculated.

### 3.4.4 Summary

We are interested in the application of the optimisation algorithms to the variational methods. It is generally agreed that gradient based methods converge much faster than gradient free methods when the function is not stochastic. This has also been demonstrated in benchmarking simulations of small molecules [Cla+20]. However, the challenge arising from stochastic functions has regenerated interest and efforts in optimisation algorithms for quantum computing. The main problem facing gradient based methods when the functions are stochastic is the gradient computation. In the presence of even a small amount of stochasticity, gradient based methods fail [Lav+20]. To counter the problem, novel methods have been proposed to calculate the gradient in NISQ computing, e.g. the parameter shift rule [Sch+19], quantum natural gradient [Sto+20], and quantum analytic descent [KB22]. Various alternatives and improvements for gradient based and gradient free methods have been proposed [Swe+20; Ben+19; VCM13; Zhu+19]. It is not clear whether gradient based or gradient free methods are better in the presence of stochasticity.

We discussed the mathematical foundations and working of gradient based optimisation algorithms. Undoubtedly, the most important component of such algorithms is the calculation of accurate gradients at each iteration. We discussed the difficulties to such calculation when the function is stochastic and gave countermeasures. We will apply these measures and some new ideas in the next chapter.



## Chapter 4

# Applications

This chapter is dedicated to the applications of the variational methods discussed in previous chapters. We apply the variational method to three different types of problems: the Heisenberg model, the Hubbard model, and molecules from quantum chemistry. The task is to find the ground state energy of each problem. Each problem is briefly introduced before the simulation details are laid out. Parts of the chapter are based on results published in [Jat+22a; JLM22].

### 4.1 Heisenberg model

#### 4.1.1 Introduction

The literature on the Heisenberg model abounds with exemplary introductions. In this section, we briefly introduce the model as introduced in [Blu01]. Consider two electrons having spatial coordinates  $\vec{r}_1$  and  $\vec{r}_2$ , each described by single electron wavefunctions  $\psi_a(\vec{r}_1)$  and  $\psi_b(\vec{r}_2)$ , respectively. The joint state of the two can be written as the product of the individual wavefunctions as  $\psi_a(\vec{r}_1)\psi_b(\vec{r}_2)$ . However, such a simple wavefunction does not obey exchange symmetry, i.e. by exchanging the two electrons we get  $\psi_a(\vec{r}_2)\psi_b(\vec{r}_1)$  which is not a multiple of  $\psi_a(\vec{r}_1)\psi_b(\vec{r}_2)$ . The requirement that states behave properly under particle exchange operation needs to be incorporated in the wavefunction. Furthermore, the wavefunction needs to be antisymmetric so the spin part must be either an antisymmetric singlet state  $\chi_S$  or a symmetric triplet state  $\chi_T$ . These requirements are fulfilled by

$$\begin{aligned}\Psi_S &= \frac{1}{\sqrt{2}}(\psi_a(\vec{r}_1)\psi_b(\vec{r}_2) + \psi_a(\vec{r}_2)\psi_b(\vec{r}_1))\chi_S \\ \Psi_T &= \frac{1}{\sqrt{2}}(\psi_a(\vec{r}_1)\psi_b(\vec{r}_2) - \psi_a(\vec{r}_2)\psi_b(\vec{r}_1))\chi_T,\end{aligned}\tag{4.1}$$

which include both the spatial and spin parts of the wavefunction. The variational energies of the these two states are given by

$$\begin{aligned}E_S &= \int \Psi_S^* H \Psi_S d\vec{r}_1 d\vec{r}_2 \\ E_T &= \int \Psi_T^* H \Psi_T d\vec{r}_1 d\vec{r}_2,\end{aligned}\tag{4.2}$$

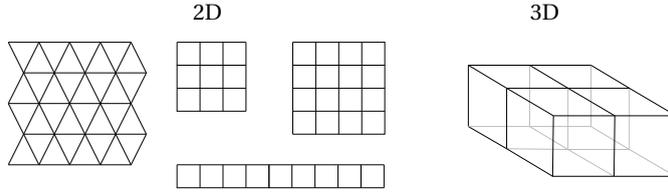


FIGURE 4.1: Illustrations of two and three dimensional lattice structures.

where spin parts of the wavefunction are normalised. The difference between the two energies is

$$\begin{aligned} E_S - E_T &= \int (\Psi_S^* H \Psi_S - \Psi_T^* H \Psi_T) d\vec{r}_1 d\vec{r}_2 \\ &= 2 \int \psi_a^*(\vec{r}_1) \psi_b^*(\vec{r}_2) H \psi_a(\vec{r}_2) \psi_b(\vec{r}_1) d\vec{r}_1 d\vec{r}_2. \end{aligned} \quad (4.3)$$

An effective Hamiltonian [Blu01] can be written as

$$H_{\text{eff}} = -(E_S - E_T) \hat{S}_1 \cdot \hat{S}_2 + C, \quad (4.4)$$

in terms of the spin-1/2 particle operators  $\hat{S}_1$  and  $\hat{S}_2$ , where  $\hat{S} = \hbar \sigma / 2$  (we set  $\hbar = 1$ ), and  $C = 1/4(E_S + 3E_T)$  is a constant term. Let us define  $\mathcal{J} = E_S - E_T$ . Thus, the spin-dependent term in the Hamiltonian becomes

$$H_{\text{spin}} = -\mathcal{J} \hat{S}_1 \cdot \hat{S}_2. \quad (4.5)$$

If  $\mathcal{J} > 0$ ,  $E_S > E_T$  and the triplet state is favoured. If  $\mathcal{J} < 0$ ,  $E_S < E_T$  and the singlet state is favoured. The final step is the assumption that the derivation above is valid for any arbitrary pair of electrons in a material and can be generalised to the entirety of the material. This leads to the Hamiltonian of the **Heisenberg model**:

$$H_s = -\sum_{i>j} \mathcal{J}_{ij} \hat{S}_i \cdot \hat{S}_j = -\frac{1}{2} \sum_{ij} \mathcal{J}_{ij} \hat{S}_i \cdot \hat{S}_j, \quad (4.6)$$

where  $\mathcal{J}_{ij}$  is the *exchange constant* between the  $i^{\text{th}}$  and  $j^{\text{th}}$  spins, and the factor of 1/2 in the right-most term appears because each pair is counted twice. This work will focus on the antiferromagnetic case of  $\mathcal{J} < 0$ .

### Categorisation of the model

The model described by Eq. (4.6) can be categorised based on its use. The model will be called one dimensional if the spins are located on the sites of a one dimensional lattice. Similarly, it can also be used for two or three dimensional lattices. The exchange constant, also called the coupling interaction strength, is chosen to be either unity (isotropic) or random. For lattices with two or more dimensions, the structures can further take various forms. They may be regular or frustrated. Furthermore, the boundary conditions at the ends of each lattice can either be open or periodic. For a visual illustration, we draw some common two and three dimensional lattices in Fig. 4.1. Shown are frustrated, square shaped, and ladder type two dimensional lattices. As an example of a three dimensional lattice we draw a  $3 \times 3 \times 2$  structure. The vertices are the sites where the particles can be accommodated. The edges

between vertices represent the coupling between neighbouring sites. A site can also be connected to a next-to-neighbour site as an appropriate change in the model, and so on.

Periodic boundary conditions are assumed for  $H$ . The energy of  $H$  is given in dimensionless units. The Hamiltonian given in Eq. (4.6) can be converted into string representation once it has been categorised. As an example, consider the ferromagnetic Heisenberg Hamiltonian consisting of spins on a one dimensional lattice of length  $N$  having  $m$  terms, given by

$$H = -J \sum_{i>j} \sigma_i \sigma_j, \quad (4.7)$$

where  $\sigma = \{\sigma^x, \sigma^y, \sigma^z\}$ , we set  $J = 1$ , and  $j = 1, \dots, N-1$  and  $i = j+1$  are the indices of the lattice sites. The spin operators of Eq. (4.6) are taken in the Pauli basis. Other choices of  $i$  and  $j$  indices can be used to describe two or more dimensional models. The string formulation is obtained from the discussion in chapter 3.

### Exact diagonalisation

One way to find the ground state energies of the Hamiltonians is through exact diagonalisation. The matrix is of dimensions  $2^N \times 2^N$ . The string representation can be used to create the matrix as described in section 3.3.2. In the example considered above (Eq. (4.7)), for  $N = 3$  the matrix is

$$M_H = \begin{bmatrix} -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 & -2 & 0 & 0 & 0 \\ 0 & -2 & 1 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -2 & -2 & 0 \\ 0 & -2 & -2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & -2 & 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 \end{bmatrix}_{2^3 \times 2^3}. \quad (4.8)$$

The diagonalisation is performed after the creation of the matrix. A given square matrix  $A$  is said to be diagonalisable if it is *similar* to some diagonal matrix, i.e. if there exists an invertible matrix  $P$  such that

$$A_D = P^{-1}AP, \quad (4.9)$$

is diagonal. The diagonal matrix  $A_D$  contains the eigenvalues of  $A$ , and usually, the lowest of these is of interest since it corresponds to the ground state of a physical system represented by  $A$ . The matrix in Eq. (4.8) can be diagonalised to give the lowest eigenvalue  $-3$ . This example also illustrates the dimension of  $M_H$  as a function of  $N$ , which is of order  $\mathbb{O}(2^{2N})$ . The amount of memory and processing required scale as  $\mathbb{O}(2^{4N})$  and  $\mathbb{O}(2^{6N})$ , respectively. Due to the exponential scaling, it becomes difficult to store and process the matrix at large values of  $N$ . Thus, exact diagonalisation is not suited for finding the ground state of Hamiltonians with large number of spins. This is one of the many reasons why the Heisenberg model is a hard problem to solve.

The resources of quantum computers can be leveraged to find the ground state energy without exact diagonalisation. The remainder of this section is dedicated to the study of the implementation of the Heisenberg model on quantum computers. Since the current computers are error prone and belong to the noisy intermediate scale quantum (NISQ) computers [Pre18], quantum simulators [De +07; De +19; Ale+] that mimic quantum computers will assist in the study.

### Classical memory

Similar to the scaling of the storage space required for storing the  $H$  matrix, the classical random access memory required for storing the state vector scales as  $2^N$  and thus doubles per qubit. Although state vectors for systems requiring up to 30 qubits can be stored on modern personal computers, simulations of several such interesting medium scale applications are rarely encountered. This suggests that the memory requirement cannot be the only hindrance. Contributing factors are a large number of gate operations and thousands of iterations required by the optimisation algorithm, making such applications impractical. To make it practically feasible, we use a massively parallel simulator [De +19; De +07] that accommodates the resource demands on a supercomputer [Juw]. Quantum verification in the supremacy experiment was performed by the same software [Aru+19].

### Verifying the ground state energy

To verify the results in the current case, we compare it with a mean-field model. The main idea behind the mean-field model is to use the average value of all the spins (excluding the one of interest) in a system as a mean value affecting another single spin of interest. There are multiple formulations of mean-field theories; refer to [CL95] for examples. The Hamiltonian of interest for the verification is given as

$$H = -\frac{J}{L} \sum_{\langle i,j \rangle} \sigma_i \cdot \sigma_j - g \sum_{i=1}^L \sigma_i^z, \quad (4.10)$$

where  $J, L$ , and  $g$  are the model parameters [HD00], and  $\langle i, j \rangle$  is the coupling of each spin to every other spin. The Hamiltonian has eigenvalues

$$E_{l,m} = -2Jl(l+1)/L - 2gm + \frac{3}{2}J. \quad (4.11)$$

For simplicity, let us rewrite Eq. (4.10) such that

$$LH = -J \sum_{\langle i,j \rangle} \sigma_i \cdot \sigma_j, \quad (4.12)$$

where setting  $g = 0$  simplifies the calculation. The corresponding eigenvalues are given by

$$E_l = -2Jl(l+1) + \frac{3}{2}JL. \quad (4.13)$$

Two different cases are used to verify the simulation results. First, the antiferromagnetic case. Taking a two-qubit Hamiltonian

$$H_s^{\text{af}} = \sigma_1^x \sigma_2^x + \sigma_1^y \sigma_2^y + \sigma_1^z \sigma_2^z, \quad (4.14)$$

where  $H_s^{\text{af}}$  denotes the antiferromagnetic Hamiltonian, the model parameters are set  $L = 2, J = -1$ , and  $l = 0$ . The energy is then calculated using Eq. (4.13) and is given by

$$E^{\text{af}} = -2 \times -1 \times 0(0+0) + \frac{3}{2} \times -1 \times 2 = -3. \quad (4.15)$$

The simulation also produces the value  $-3$  using  $\sigma_1^x \sigma_2^y$  operators in the ansatz [O'M+16], initial state  $|10\rangle$ , and a single parameter  $\theta = \pi/2$ . Thus, the simulation is verified to work for the

antiferromagnetic case.

Second, the ferromagnetic case. Taking the two qubits Hamiltonian given by

$$H_s^f = -\sigma_1^x \sigma_2^x - \sigma_1^y \sigma_2^y - \sigma_1^z \sigma_2^z, \quad (4.16)$$

where the model parameters are set  $L=2, J=1$ , and  $l=1$ . The energy is then given by

$$E^f = -2 \times 1 \times 1(1+1) + \frac{3}{2} \times 1 \times 2 = -4 + 3 = -1. \quad (4.17)$$

The simulation model produces the value  $-1$  using operators  $\sigma_1^x \sigma_2^y$  in the ansatz, initial state  $|10\rangle$ , and a single parameter  $\theta = -\pi/2$ . In summary, the simulation produces the correct ground state energies for both the ferromagnetic and antiferromagnetic cases. This is also true for Hamiltonians of more than two spins.

### 4.1.2 Finding an ansatz

The choice of an ansatz is critical to the performance of the variational methods. Finding an ansatz for a two qubits problem is relatively easy, but finding an ansatz for large  $N$  is an open problem. One may choose to rely on knowledge from classical theories. For the one dimensional case, the Bethe ansatz [Bet31; Sut85] is known to give the exact solutions. However, its implementation on a quantum computer seems difficult because the gate decomposition involves square roots of complicated functions [Nep20]. Nonetheless, some progress has been made using real-valued solutions [VD+21]. Furthermore, we wish also to find the ground states of two and three dimensional lattices for which no analytical means to solve the problem are known. In this section, we demonstrate how we found a suitable ansatz for the Heisenberg model.

We tune the variational method to optimise the overlap of the trial wavefunction with the ground state instead of the energy. The overlap of a normalised trial wavefunction with the normalised exact eigenfunction(s) of the problem Hamiltonian is defined as

$$O = \sum_{i=1}^d |\langle \psi^T | \psi_{ik}^E \rangle|^2, \quad (4.18)$$

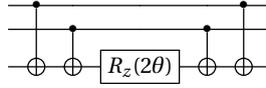
where  $\psi^T$  is the trial wavefunction and  $\psi_{ik}^E$  the exact eigenfunctions of the  $k^{th}$   $d$ -degenerate eigenvalue. We set  $k=0$  since we are interested in the ground state. The value of  $O$  is bounded as  $0 \leq O \leq 1$ .

#### Two and three qubits

We begin with the simplest case of two qubits. Dealing with only two qubits the choice of combinations of the Pauli terms can be completely exhausted by trying all possible combinations. The ansatz is then given by

$$U(\theta) = e^{-i\theta \hat{a}_j}, \quad (4.19)$$

where  $\hat{a}_j$  is a single term with an identifier index  $j$ . In total there are fifteen possible single (e.g.  $\hat{a} = \sigma^x \sigma^j$ ) and double (e.g.  $\hat{a} = \sigma^x \sigma^z$ ) operator products terms. Using only one term at a time, we avoid the need for factorisation of the exponential [DR87] in the case of three or more qubits. Each one can be converted into a circuit as discussed in section 3.3.1. We try all fifteen terms independently and optimise for the maximum wavefunction overlap. Thirteen

FIGURE 4.2: Circuit implementation of  $\exp(-i\theta\sigma^z\sigma^z\sigma^z)$ .

terms give an overlap of 0.5, and two give a complete overlap of 1.0. These two terms were

$$\hat{a}_1 = \sigma_1^x \sigma_2^y \quad \text{and} \quad \hat{a}_2 = \sigma_1^y \sigma_2^x.$$

Using each term and minimising the energy instead also gives the correct energy of  $-3.0$  which matches with the value calculated using exact diagonalisation.

In the case of three qubits, we repeat the same procedure by restricting to single and double operator products where  $j = 1, 2, 3, \dots, 36$ . Thirty-two terms give an overlap of two-thirds, while only four terms give a complete overlap. The terms giving the complete overlap are

$$\hat{a}_1 = \sigma_1^x \sigma_2^y \sigma_3^I, \quad \hat{a}_2 = \sigma_1^y \sigma_2^x \sigma_3^I, \quad \hat{a}_3 = \sigma_1^I \sigma_2^y \sigma_3^x, \quad \text{and} \quad \hat{a}_4 = \sigma_1^I \sigma_2^x \sigma_3^y.$$

Using each term and minimising the energy also gives the correct ground state energy  $-3.0$ , which matches the result from exact diagonalisation.

#### Four or more qubits

The number of single and double product terms scale as  $3^2 N! / 2!(N-2)! + 3N$  for  $N$  qubits. Given this scaling, it quickly becomes difficult to check each term and even harder to check combinations of terms. Checking individual terms is also likely to fail since it is unlikely that the ground state of a problem when  $N > 3$  can be expressed by such a simple ansatz. An ansatz for  $N \geq 4$  can be divided into two categories: neighbouring and non neighbouring. A 'neighbouring' ansatz has combinations of terms that have products of terms on consecutive indices only. A 'non neighbouring' ansatz has both neighbouring terms and terms that have product terms with indices not necessarily consecutive. The cases of two and three qubits hint that the combinations of terms with  $\sigma^x$  and  $\sigma^y$  are more successful at finding the complete overlaps than others. In other words, the ground state seems to be expressed by these terms, provided the optimal parameters can be found. Thus, for  $N \geq 4$ , we proceed with combinations of terms with  $\sigma^x$  and  $\sigma^y$  only.

We observed that placing the parametrised gate only on the last qubit of the quantum circuits led to better results. In the case of  $N \geq 3$ , a third  $\sigma^z$  operator is needed in the ansatz, which always operates on the last qubit. An exemplary circuit is shown in Fig. 4.2. We use such a construction in the remainder of the section.

The task is to construct a variational ansatz that solves for the ground state of the anti-ferromagnetic Heisenberg Hamiltonian. We are interested in an ansatz that contains a set of operators instead of a sum of operators. Such an approach will have a smaller circuit depth. One can start with a large pool of operators; however, this overparametrisation will not necessarily lead to the solution. We wish to have a minimal pool of operators that express the ground state.

To accomplish similar tasks, an adaptive method has been proposed [Gri+19]. The adaptive method has been extended to find excited state energies as well [Yor+21]. For a benchmark of fixed ansatz and the adaptive ansatz methods for small diatomic molecules (i.e.  $\text{H}_2$ ,  $\text{NaH}$ )

Previous works	This work
Used in quantum chemistry (e.g. LiH, BeH <sub>2</sub> )	Used for the Heisenberg model
Minimisation of the energy	Maximisation of the overlap
Best term has the steepest gradient	Best term has the best overlap
Pool has single and double fermionic operator products	Pool has double spin operator products
No variational optimisation in choosing best term	Variational optimisation in choosing best term

TABLE 4.1: Comparison between the implementation of the adaptive variational ansatz preparation method proposed in [Gri+19] and the implementation used in this work.

see [Cla+20]. The idea behind an adaptive method is to grow the number of parametrised operators step-by-step for a given problem until a desired accuracy is achieved. Here we will use it for the Heisenberg model. We implement an adaptive method different from the one proposed in [Gri+19]. The differences in our approach to the original are tabulated in Table 4.1.

The setup we adopt for finding the ansatz is as follows. As shown in Fig. 4.3, a quantum computer is initialised with a state acted upon by a circuit created out of the terms in the ansatz having  $i$  independent parameters. Each parameter belongs to a single term in the ansatz. For the first step,  $i = 0$  and the ansatz is empty. One term from the operator pool is appended to the ansatz and all the parameters are optimised for maximum overlap. Since each term in the pool is independent, this step can be done in parallel. The optimised outputs are arranged  $\bar{O}_m = O_1, O_2, \dots, O_m$ , where  $m$  is the number of terms in the operator pool. Then, the highest overlap in  $\bar{O}_m$  is checked against a threshold  $O_{th}$ , and if it meets the threshold, the procedure is stopped. If the threshold is not met, the term with the highest overlap is appended permanently to the ansatz and the number of parameters is increased by one. Then, we begin the next cycle and the loop continues to append the terms with the highest overlap until a certain threshold is met. Since we are interested in the ground state of the Heisenberg model,  $|\psi_0^T\rangle \rightarrow |\psi_0^E\rangle$  when  $O \rightarrow 1$ . The value of  $O$  can be chosen to be arbitrary close to 1 depending on the precision required for the overlap.

Techniques have been developed to reduce the size of the operator pool, which reduces the computational resources required, e.g. mutual information assistance [Zha+21]. Similar ideas for improving such adaptive techniques and reducing the computational resources have been studied [Tan+21; Yao+21]. Since our operator pool is already heavily reduced, we do not apply such techniques.

We apply the adaptive method to antiferromagnetic Heisenberg chains of length  $N = 4, 5, 6$ , and 7. The results for neighbouring and non neighbouring pools are shown in Fig. 4.4. The overlap precision threshold was set to  $O_{th} = 0.9999$ . We choose an arbitrary value for the threshold but it can be problem dependent, e.g. chemical accuracy in quantum chemistry. The operator pools were non-depletive, meaning that a term once used is not removed from the pool and could be reused. The opposite is also possible but not used. We observe that for  $N = 4$ , the minimum number of terms required to reach  $O_{th}$  are the same. However, this changes as  $N$  is increased. The difference becomes visible for  $N \geq 5$  where the number of terms required to overpass the threshold is fewer from the non neighbouring pool than the neighbouring pool. The difference increases as  $N$  increases; thus, it can be hypothesised that

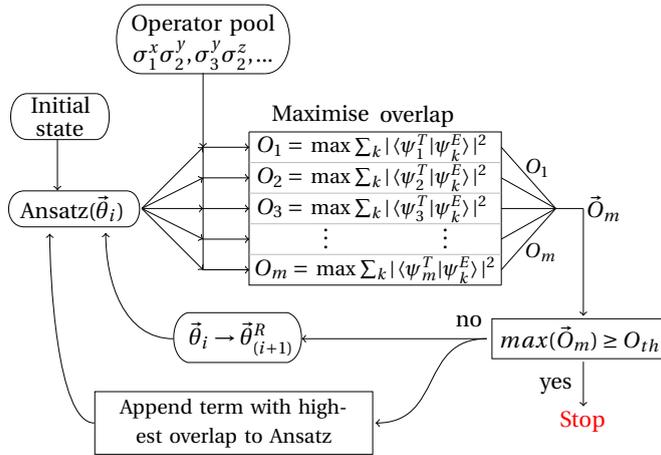


FIGURE 4.3: Schematic diagram of the working of an adaptive variational ansatz selection method. One operator is picked from the operator pool and is added to the ansatz. The overlap of the ansatz is then maximised and the term with the highest overlap is permanently appended to the ansatz. The process is repeated until a threshold is reached.

it will increase further for  $N > 7$ . It is also interesting to note that an overlap of 0.9 or ninety per cent is reached within a few terms, but the threshold accuracy is reached with significantly more terms. For example, for  $N = 7$  a 90 per cent overlap is reached at the twelfth term for both neighbouring and non neighbouring cases, but 99.99 per cent overlap is reached at 72 and 59 terms, respectively. The non neighbouring pool appears to be advantageous if it is important to keep the number of terms (and therefore the number of parameters) to a minimum. However, current quantum computers do not offer connectivities between the qubits suitable for this and swap gates increase the circuit depth significantly. The neighbouring pool appears to be advantageous if the task is to implement the ansatz on current quantum computers, which mostly offer nearest neighbour connectivities without requiring swap gates. Furthermore, the fact that in all cases the adaptive method is able to converge adds confidence that the choice of a reduced form of the two product operators pool, namely only the  $\sigma^x$  and  $\sigma^y$  terms, was reasonable.

## Summary

Table 4.2 shows the energies and overlaps obtained from the adaptive ansatz selection method. The adaptive method did not optimise for the energy but the overlap; however, optimising the energies also leads to similar results. The results may have been different because the optimisation landscapes of energy and overlap are not necessarily the same. It was assumed that the overlap maximisation always finds the global maximum of the corresponding landscape. This assumption leads to the inference that the term appended to the ansatz was the best term. However, this is not necessarily true. Finding global minima is itself a hard problem [BK21]. Despite this drawback, we observe that the adaptive method was able to converge for the small sized problems.

The adaptive method used here uses the wavefunction to compute the overlap that is maximised. The wavefunction is not accessible on a quantum computer; therefore, this method

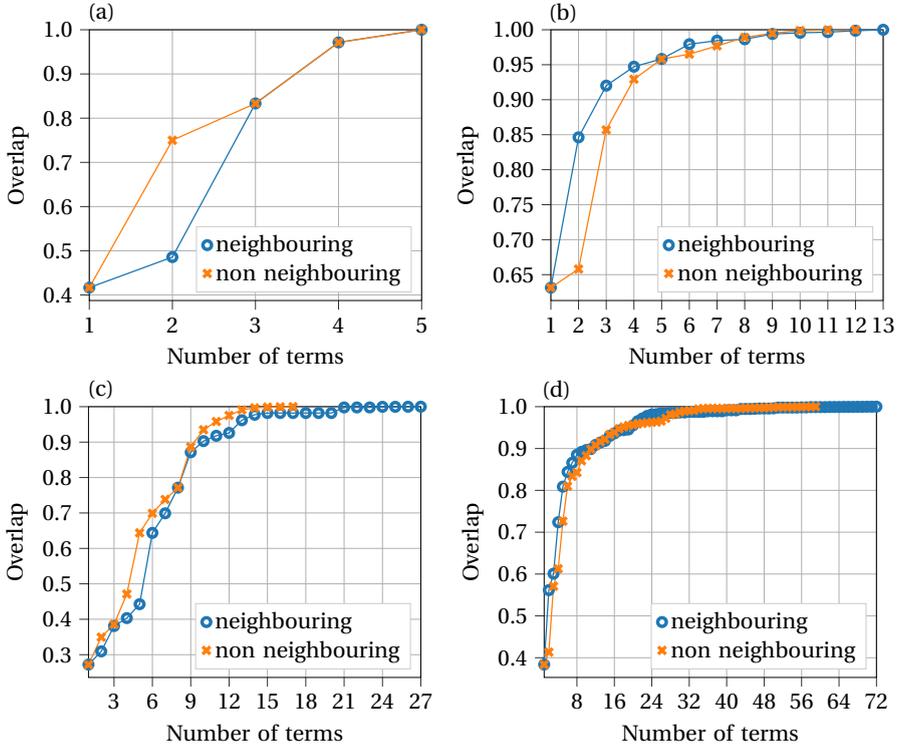


FIGURE 4.4: The overlap of the optimised trial wavefunction with the ground state of the isotropic antiferromagnetic Heisenberg model as a function of the number of operators added to the ansatz. The overlap threshold is set to  $9.999 \times 10^{-1}$ . Shown are problem requiring (a) 4 qubits, (b) 5 qubits, (c) 6 qubits, (d) 7 qubits.

Qubits	Pool	Ground state energy	Variational energy	Overlap
4	Type 1	-8.00000	-8.00000	1.00000
4	Type 2	-8.00000	-8.00000	1.00000
5	Type 1	-7.47214	-7.47210	0.99999
5	Type 2	-7.47214	-7.47182	0.99997
6	Type 1	-11.21110	-11.21089	0.99998
6	Type 2	-11.21110	-11.20999	0.99992
7	Type 1	-11.42072	-11.42072	1.00000
7	Type 2	-11.42072	-11.42071	1.00000

TABLE 4.2: Energies and overlaps calculated at the termination of the adaptive variational ansatz selection method. Type 1 is neighbouring and type 2 is non neighbouring. The ground state energies are calculated by diagonalising the Hamiltonian matrix. The variational energies are calculated using the parameters obtained from the optimisation of the corresponding overlaps.

is not suitable for solving practical problems on the computer. Additionally, the resource requirements of solving a variational calculation for each term make it impractical. However, its application to the small scale prototype problems has given us significant hints as to how to build an ansatz for large scale problems. Namely, that products of two Pauli operators may be sufficient to estimate the ground state energy. We observed that by using an operator pool consisting of only products of two Pauli spin matrices, one is able to find the ground state, when the overlap approaches unity. This motivates an ansatz we introduce in the following section.

### 4.1.3 The XY ansatz

The set of operators that was found to give a good overlap with the ground state consists of the form

$$U(\theta) = e^{-i\theta\sigma^x\sigma^y\sigma_N^z}. \quad (4.20)$$

This knowledge helps us guess the set of operators whose parametrisation is likely to approximate the ground state energy. In general, even if the set of operators is identified, their ordering is still important. The same operators but ordered differently can produce variational energies different from one another [Gri+20; Tra+19]. A change in order may also alter the optimisation landscape. We order the operators as follows. The unitary operation as a product of all exponentiated operators is given by

$$U(\theta) = \prod_{l=N-1}^1 \prod_{k=N}^{l+1} U_{lk}(\theta_{lk}) \prod_{l=N-1}^1 \prod_{k=N}^{l+1} U_{kl}(\theta_{kl}), \quad (4.21)$$

where

$$U_{kl}(\theta_{kl}) = \begin{cases} e^{-i\theta_{kl}\sigma_k^y\sigma_l^x} & \text{if } k = N \text{ or } l = N, \\ e^{-i\theta_{kl}\sigma_k^y\sigma_l^z} & \text{otherwise.} \end{cases} \quad (4.22)$$

The parameters  $\theta_{kl}$  affect the phase of the  $N^{\text{th}}$  qubit. There are  $N(N-1)$  unitary operators in Eq. (4.21). The ansatz expressed by Eq. (4.22) is termed the XY ansatz. Simple tests with the XY ansatz showed that the arrangement used for simulation in this dissertation performs better than 100 random shufflings of the terms (tested for 7 qubits). We do not rule out better ordering than the one we found. Heuristics have been proposed to improve the ordering of the terms and tested through simulations [Tra+19]. We do not implement these heuristics since the ordering presented in this dissertation performs well. Finding the most optimal ordering for each  $N$  is beyond the scope of this dissertation.

### 4.1.4 Initial state

The ground state of a two-spin system with an antiferromagnetic coupling in classical mechanics is the Néel state. This knowledge can be extended to a multiparticle system which maintains the Néel state as its ground state. In quantum mechanics, the Néel state no longer describes the ground state of the system. The ground state of the one dimensional isotropic antiferromagnetic Heisenberg model is located in the sector of total spin zero. The Néel state, for an even number of spins, is also in this sector. Therefore, for the variational approach, it can be expected that a system in the Néel state is likely closer to the quantum mechanical ground state than any other arbitrary state chosen at random.

We choose to initialise the quantum computer in the Néel state to solve for the ground state energy of the antiferromagnetic Heisenberg model. Additionally, we use the Néel state for all types of lattices. These include two and three dimensional lattices, frustrated lattices,

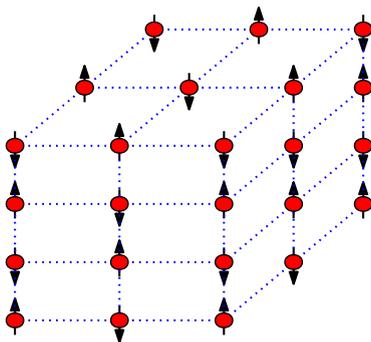


FIGURE 4.5: Picture of the Néel state on a  $3 \times 3 \times 4$  bipartite spin lattice. Observe that the neighbour of any given spin is antiparallel to it.

etc. Half of the lattice spins for the frustrated lattices would be initialised antiparallel to the other half, regardless of their location on the lattice. An example configuration is shown in Fig. 4.5. To initialise the Néel state as the initial state of a quantum computer, the qubit indices are first mapped to the lattice indices. Then, the parallel and antiparallel spins are assigned the computational states  $|0\rangle$  or  $|1\rangle$ , respectively.

#### 4.1.5 Results - Part I

All the basic ingredients required for finding the ground state energy of the Heisenberg model have been prepared. We have verified that the simulation produces results matching those of exact diagonalisation. There are two different ways to proceed. First, through emulation of a perfect quantum computer on a classical computer where the wavefunction is accessible. These results are presented in this section. Second, on an actual quantum computer or through sampling of the wavefunction that mimics a quantum computer. These results are presented in the next section.

An emulator directly manipulates the underlying state vector describing the quantum system. Since the Hilbert space grows exponentially, and therefore so does the state vector, emulators are restricted to a limited number of qubits. However, testing even on a small number of qubits helps to benchmark methods and to predict if they will work when they are run on a quantum computer larger than what can be classically emulated. In this section, the emulator is used to test various methods for finding the ground state energy of the Heisenberg model. Various options affecting the simulations were tried which include different ansätze, different initialisations of the variational parameters, optimisation algorithms, and initial states. The aim was to find the best set of options that are able to solve for the ground state energy within reasonable accuracy. The variational energies will be compared to ground state energies computed using exact diagonalisation or Lanczos method.

#### Isotropic Hamiltonians

The Hamiltonian described an isotropic one-dimensional lattice with periodic boundary conditions. The ground state energies up to 12 qubits were calculated using exact diagonalisation, and for all others using the Lanczos method. The XY ansatz was initialised with all parameters set to zero in all cases. The optimisation algorithm used was Sequential Least Squares Quadratic Programming (SLSQP) [Kra88] in the SciPy package [JOP+01]. The Néel

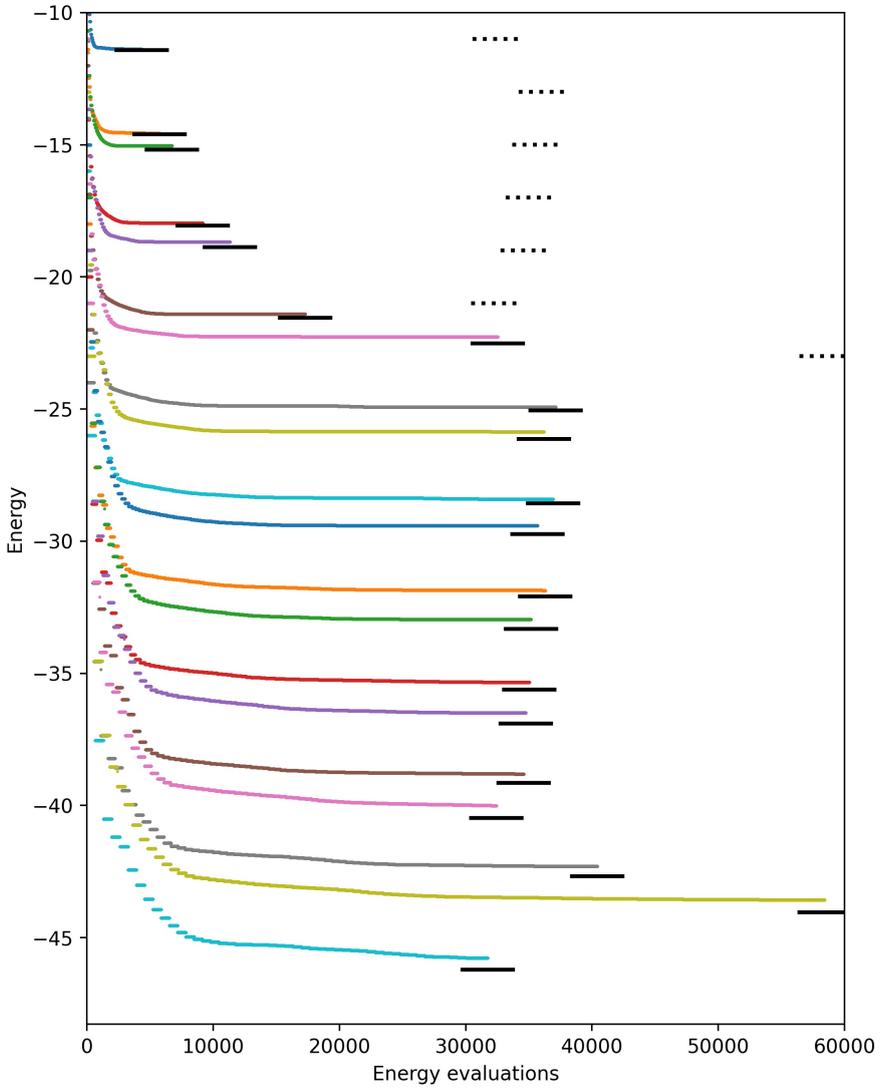


FIGURE 4.6: Variationally optimised energies of the antiferromagnetic Heisenberg rings of size 7 to 26 (top to bottom) as a function of the number of energy evaluations. The solid (dotted) black lines below (above) the end of each energy curve correspond to the ground (Néel) state energy of the corresponding curve. The Néel state energies are only shown for selected odd rings of length 13 to 25.

state energies for selected odd ring sizes show the difference between the initial and ground state energies. The results for the simulations are shown in Fig. 4.6. The following observations are drawn from the results shown in Fig. 4.6.

- The approximation to the ground state energy using the XY ansatz converges rapidly at first and then slows down. This can be observed by the fact that the curves approach the ground state energies as the optimisation proceeds. We observe this trend in all cases.
- The staircase like progress of the energy curves is due to the working of the gradient based optimisation algorithms. At each iteration, the computation of the gradient involves  $m + 1$  function evaluations, where  $m$  is the number of parameters. Since the change in  $\theta$  for gradient computation using the finite differences approximation is very small, the change in energy is also very small compared to the scale on the y-axis. Thus, these small deviations are not visible, and the  $m + 1$  points appear as horizontal lines or "staircase steps". The length of these lines increases for increasing number of qubits (downwards in the plot) since the number of parameters scales as  $N(N - 1)$ .
- The optimisation algorithm did not reach convergence for all cases starting from 14 qubits. Note that the convergence criterion is met when the energy cannot be improved to a threshold of at least  $10^{-6}$  per iteration. Since all these cases were performed on the supercomputer [Juw], time restrictions for submitted jobs stop the simulations after a period of 24hrs. The optimisation process could be restarted from the last parameters, as shown in the case of 25 qubits.

To observe the trend of the larger lattices, we restrict their total energy evaluations between 30,000 and 40,000. The differences between the ground state and variational energies,  $|E_0 - E_f|$ , are plotted in Fig. 4.7 in blue circles. The difference begins to show starting from 7 qubits and increases linearly with the number of qubits. The difference is lower for an even number of qubits than for an odd number. The cause for this trend is unclear. One reason could be that the systems with an even number of qubits have non-degenerate ground states while systems with an odd number of qubits have at least a two-fold degeneracy. In all cases, the parameters were initialised from zeros. However, this is not necessarily the best possible choice to find the global minimum. We do not know of a better choice, and finding it numerically is a hard problem itself. For systems with a larger number of qubits the optimisation process did not terminate. Restarting the optimisation algorithm can potentially lead to lower but unlikely substantially lower energy than shown in the plot. We learn from the case of 25 qubits that progress remains relatively slow beyond this range. The data highlights the importance of starting from low initial energy, preferably as close to the ground state energy as possible.

### Random couplings

We investigated the performance of the XY ansatz for the case of random coupling coefficients in the model. We change the Hamiltonian such that  $J_{ij} \in (0, 1]$ , the case of the random coupling. The Néel state served as the initial state. The results are shown in Fig. 4.8(a). We observe that due to the random couplings, the ground state energies are no longer monotonically decreasing in proportion to the ring size. The relatively large drops initially were also seen for the random case, as well as a step like optimisation curve. The gap between the variationally found energy and the ground state energy grows in proportion to the ring size. The ground state energy was approximated reasonably well by the XY ansatz. Using the same

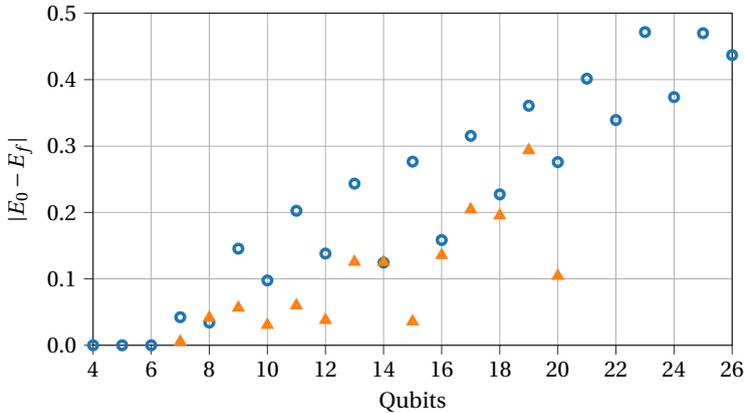


FIGURE 4.7: Difference between the ground state energies and the optimised variational energies obtained using the XY ansatz initialised from zeros as a function of the system size in terms of the number of qubits. Blue circles represent the isotropic case and orange triangles the random coupling case.

ansatz, we get a figure similar to Fig. 4.6, for problem sizes of up to 20 qubits. It is more relevant to compare the isotropic and random coupling cases than the absolute values for each case. Therefore, the difference between the ground state and variational energies,  $|E_0 - E_f|$ , is plotted in Fig. 4.7. The blue dots are the isotropic case (also shown in Fig. 4.6) and the orange dots are for the random couplings case (Fig.4.8(a)).

Also for the random coupling case, the difference between variational and ground state energies is visible starting from 7 qubits. The difference up to 20 qubits is shown in the orange dots of Fig. 4.7. From the given data, the XY ansatz for the random coupling case seems to work well. Not all reported values correspond to the cases where the optimisation algorithm signalled convergence. The values do not show a structure visible for the isotropic case since the values  $J_{ij} \in (0, 1)$  are random and this changes the exact energy of the system as well as the energy landscape to be optimised. In conclusion, the difference between the ground state and variational energies for the random coupling case appears to be lower than in the corresponding isotropic case when the same simulation settings are used. A more concrete analysis requires averages of many more sets of random coupling coefficients. However, such simulations are computationally prohibitive even on supercomputers.

The variational method can be extended further to simulate two and three-dimensional models using all the mechanisms previously discussed. We tried the XY ansatz with zeros as initial parameters on isotropic ladders that were mapped on 6 to 26 qubits. The optimisation progress for the ladder cases is shown in Fig. 4.9. The dotted lines show the Néel state energy which is the initial energy to start the optimisation. The cases of 16 and more qubits were restarted from the previously saved parameters at 20,000 function evaluations. The energy continues to decrease albeit slowly even at 40,000 function evaluations suggesting that the optimiser did not reach a minimum. Continuing the optimisation process is again possible but not pursued due to limitations on the supercomputer budget. Even with the final variational energies at unconverged optimisation, the gap between the ground state and variational energies seems to increase linearly with the ladder size.

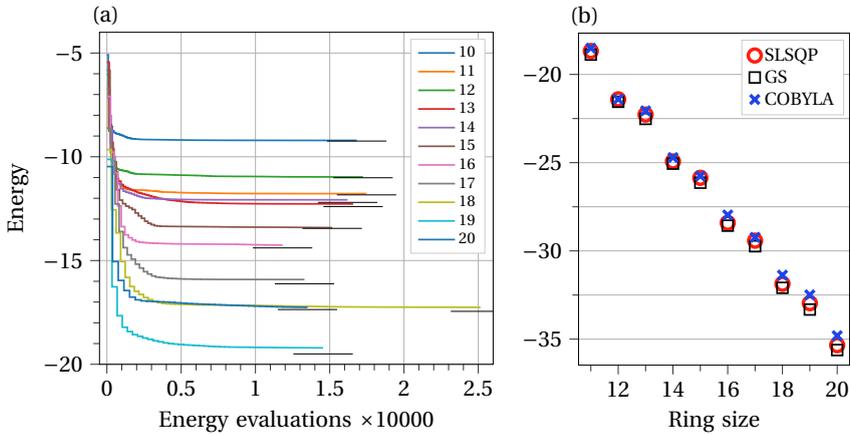


FIGURE 4.8: (a): Same as Fig. 4.6 except that different rings (see legend) had random coupling interactions. (b): Comparison between the energies obtained at the end of the variational optimization using different optimisation algorithms and the ground state (GS) energies. The couplings were isotropic.

### Two- and three-dimensional lattices

Finding the ground state energy of square lattice of dimensions  $\sqrt{N} \times \sqrt{N}$  was investigated for  $N = 16, 25, 36$ . The XY ansatz was used in all cases. For  $N = 16$  and open boundary conditions, the optimiser converged to  $-35.492$  after 48000 energy evaluations. The ground state energy is  $-36.757$ . Similarly, for  $N = 25$  and open boundary conditions, although the optimiser did not converge, the progress at 52000 energy evaluations dropped to less than  $10^{-3}$ . The variational energy was found to be  $-57.454$  as compared to the ground state energy  $-58.785$ . The largest case investigated was  $N = 36$  with periodic boundary conditions. Due to the large size of the lattice and a total of 1260 independent parameters, no more than a single iteration could be performed on the supercomputer in one run. However, restarting is possible. Despite the large size of the lattice, due to the XY ansatz, the total number of gate operations were below 4100. After eight iterations, the energy dropped from  $-72.000$  to  $-94.694$ . The ground state energy is  $-97.758$  [Fen].

Only a select few three-dimensional lattices are possible due to the memory limits. Even a relatively small  $4 \times 4 \times 4$  lattice is beyond what can be currently emulated using variational methods. Nonetheless, we used the variational algorithms to find the ground state energies of a few isotropic lattices. Using the XY ansatz and zeros as initial parameters, the minimum energy of a  $3 \times 3 \times 2$  lattice was  $E_f/N = -2.457$ . The ground state energy is  $-2.617$  [Fen]. We then implemented an ansatz such that we doubled the number of the operators in the XY-ansatz, thus making the active parameter space  $2N(N-1)$ . This 'new' ansatz gave  $E_f/N = -2.592$ , improving the energy estimate. We used the XY-ansatz for  $3 \times 3 \times 3$  and  $3 \times 3 \times 4$  lattices to obtain  $E_f/N = -2.607$  and  $E_f/N = -2.644$ , respectively. The ground state energies are  $-2.720$  and  $-2.797$ , respectively.

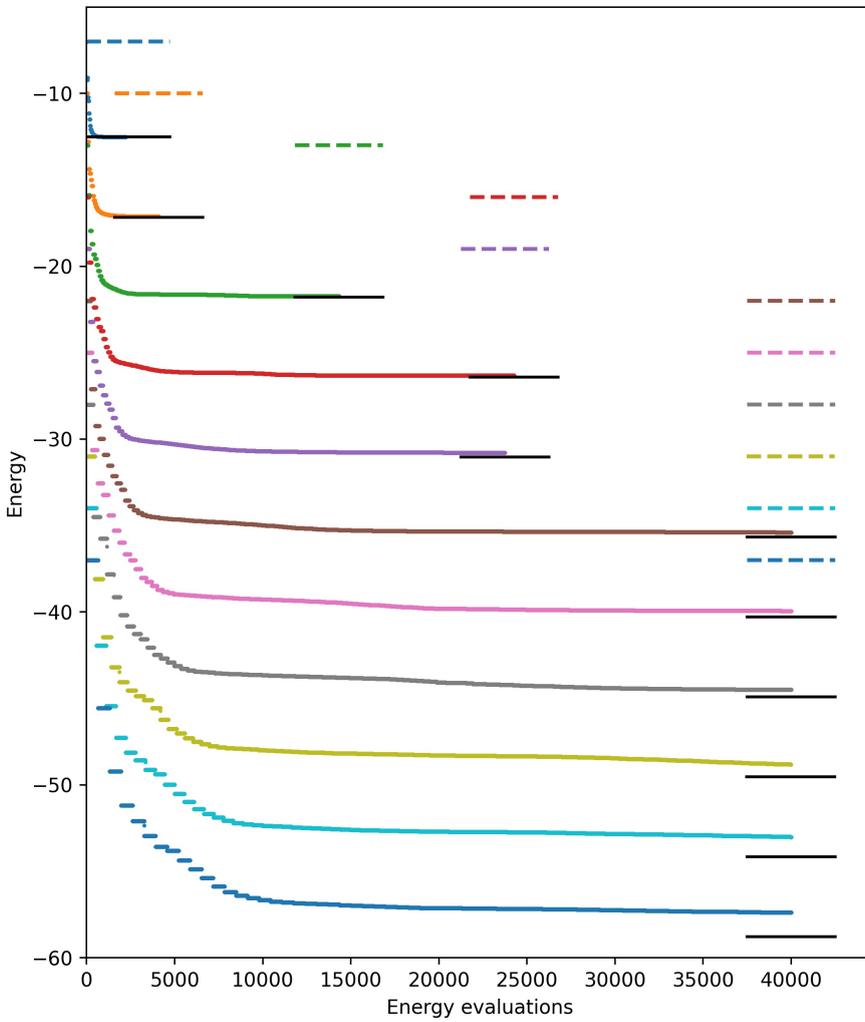


FIGURE 4.9: Same as Fig. 4.6 except that the lattices under consideration were isotropic antiferromagnetic Heisenberg ladders with 6 to 26 qubits. The Néel state energies are shown for all cases in the same colour as the energy curve.

### Different optimisers

We investigated the use of two different types of optimisers. Candidates for the gradient based and gradient free categories were SLSQP and constrained optimisation by linear approximation (COBYLA) [Pow94], respectively. They were tasked to start from the same initial parameters and to find the ground state energies of the isotropic rings. The results are shown in Fig. 4.8(b). The SLSQP optimiser converged and self terminated, requiring less than  $4 \times 10^4$  energy evaluations. The COBYLA optimiser was manually stopped at about  $5 \times 10^4$  energy evaluations. Despite the fewer number of evaluations, SLSQP was able to approximate the ground state energy better than COBYLA, although both performed reasonably well. The difference between the energies obtained by the two increases with increasing ring size, as shown in Fig. 4.8(b). These results confirm the commonly held belief that if functions are smooth and gradient calculation can be expected to succeed, gradient based methods converge faster [NFT20].

### The gradients

The gradient calculations may not always succeed, for example, in the case of a stochastic function. Calculating the energy by sampling an actual quantum computer can be thought of in terms of a stochastic function. In this and the next subsection, we discuss how to calculate gradients in such cases. Consider an ansatz having  $m$  independent parameters acting on an initial state,

$$U(\boldsymbol{\theta})|\psi_0\rangle = U_m(\theta_m)U_{m-1}(\theta_{m-1})\dots U_1(\theta_1)|\psi_0\rangle, \quad (4.23)$$

where  $\boldsymbol{\theta} = \theta_1, \dots, \theta_m$ . In the specific case of the XY ansatz, we know that each term in the product is of the form

$$U_i(\theta_i) = \exp(-i\theta_i \hat{a}_i); \quad \hat{a}_i \in \{\sigma \otimes \sigma, \sigma \otimes \sigma \otimes \sigma_N^z\}, \quad (4.24)$$

for  $i = 1, 2, \dots, m$ . To compute the gradient in a direction  $e_i$ , the corresponding  $\theta_i$  is changed according to a forward differences approximation. For example, for finite forward differences the function is computed at  $f(\theta_i)$  and  $f(\theta_i + h)$ , which helps to calculate the gradient  $\nabla f_i$  in the direction  $e_i$ . Gradient-based methods require the gradient in each direction to be represented by the number of parameters in the ansatz. Therefore, we need

$$\nabla \mathbf{f} = \{\nabla f_1, \nabla f_2, \dots, \nabla f_m\}, \quad (4.25)$$

which contains the gradients for  $m$  directions or parameters. Important is the fact that the computation of a gradient in the direction  $e_i$  is done by changing only  $\theta_i$  while all the other parameters are kept constant. This allows the possibility of parallel processing. Combining this with the fact that each  $U_i$  is unitary, we can expect a periodic behaviour.

### Sinusoidal fitting

It is not always possible to use the finite difference approximation to accurately calculate the gradients. Here, we explore an alternative way to compute gradients given what we know about our ansatz. Given the periodic behaviour, it is possible to vary  $\theta_i$  from 0 to  $2\pi$  at equal discrete steps and fit the data points to a sinusoidal curve assuming analytical forms of the functions at a given iterate  $x_k$ :

$$f_i = a_i \sin(\theta_i + \phi_i) + c_i, \quad (4.26)$$

where  $a$  is the amplitude of the variation,  $\phi$  is the phase, and  $c$  is the offset constant. Similar ideas have been implemented [VT18; OGB21]. Having obtained an analytical form  $f_i$  due to

a change in a certain parameter  $\theta_i$ , the corresponding gradient is then given by

$$\nabla f_i(\theta) = a_i \cos(\theta_i + \phi_i), \quad (4.27)$$

where  $\theta_i$  is taken from the current iterate  $x_k$  in the direction  $e_i$ . The process is repeated  $m$  times to get the corresponding gradient for each operator in the ansatz. Thus, Eq. (4.25) becomes

$$\nabla f = \{a_1 \cos(\theta_1 + \phi_1), a_2 \cos(\theta_2 + \phi_2), \dots, a_m \cos(\theta_m + \phi_m)\}, \quad (4.28)$$

where  $\theta_i$  for  $i = 1, \dots, m$  are given by the current iterate

$$x_k = \{\theta_1, \theta_2, \dots, \theta_m\}, \quad (4.29)$$

and  $a, \phi$  are determined by fitting equally spaced points in the interval  $[0, 2\pi]$  using Eq. (4.26).

Since we consider the function evaluation to be expensive, the aim is to use the least number of evaluations without compromising on the quality of the gradients. To fit the parameters in Eq. (4.26) in the least amount of evaluations, the periodicity of  $2\pi$  for our choice of ansatz is used; see the Appendix of [OGB21] for a proof. Then,  $f(\theta_i)$  is uniquely determined if the energies at three different  $\theta_i$  are measured at equally spaced points in the interval  $[0, 2\pi]$ . Secondly, due to the periodicity, the value of  $f(\theta_i = 0)$  can be reused for  $f(\theta_i = 2\pi)$ . For a stochastic function, there will be different values. Since the proposed method fits the data to a sinusoidal curve and uses it to compute analytical gradients, stochastic functions no longer pose a problem in the gradient computation. The gradients calculated from the above mentioned method were tested for some of the problems already tested using the state vector simulator. These functions are not stochastic. The gradients and the variational energies were in perfect agreement with the results already found using finite difference methods.

### Demonstration

The following model for the noise or stochasticity was introduced to test the method. In a hybrid quantum-classical simulation, errors can occur at the various steps of the computation, e.g. during the quantum subroutine, during data transfer to the cloud, or even at the classical computer collecting the bitstrings to compute the energy. To take all these possible sources of error as well as keep a simple model for the errors, we add a random variable to the energy such that at the iterate  $x_k$  it is then given by

$$E_{x_k} = E_{x_k}(1 + R(-\epsilon, \epsilon)), \quad (4.30)$$

where  $E$  is the energy from the state vector at  $x_k$ ,  $R(a, b)$  is a random number between  $a$  and  $b$ , and  $\epsilon$  is the parameter deciding the largest amount of noise to introduce. For example, by setting  $\epsilon = 0.01$ , the energy  $E_{x_k}$  has less than one percent random error at most. Note that as the energy drops, the maximum possible variation increases. The value of  $R$  is larger than  $\hbar$  in the forward differences formula in all cases. We test the method for  $\epsilon = 10^{-1}$  to  $10^{-4}$  and for two different numbers of parameters. For the gradient based optimiser we use SLSQP and for the gradient-free optimisers there are three choices: (1) covariance matrix adaptation (CMA) [HMK03], which has been used in recent quantum variational applications [Del+22; Eba+22; Ost+21; Chi+20]; (2) COBYLA used in benchmarking [Lav+20] and for a scheduling problem [AGV20]; (3) implicit filtering (IMFIL) [Kel11] which has shown some robustness to noise as well as insensitivity to initial parameters [Lav+20].

Results for a 5-qubits model with XY ansatz having 20 parameters are shown in Fig. 4.10. The plots show the optimisation progress of the four different optimisers. The optimisers easily get stuck in local minima for  $\epsilon = 10^{-1}$ . As  $\epsilon$  is decreased, all four optimisers are able

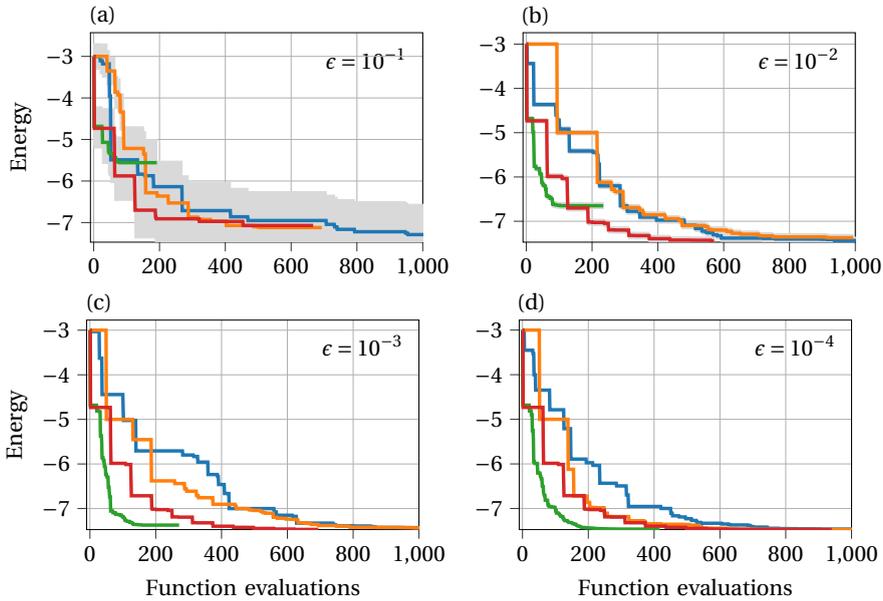


FIGURE 4.10: Optimisation progress in a 5-qubits model with different amounts of noise  $\epsilon$ . Blue curves are for CMA, orange for IMFIL, green for COBYLA, and red for SLSQP, respectively. The shaded grey area shows the maximum allowed level of noise in the energy at each function evaluation.

to make more progress towards the ground state energy. For the case with the least amount of noise,  $\epsilon = 10^{-4}$ , the fastest progress is made by COBYLA; however, the best estimate of the ground state energy is found by SLSQP. In summary, for a small number of parameters and a small amount of noise, all optimisers work, and there is no significant difference in the total number of function evaluations either.

The results for a case with a slightly more number of parameters, the 8-qubits isotropic Heisenberg model with 56 parameters is shown in Fig. 4.11. For large  $\epsilon$ , none of the optimisers found the ground state energy. As  $\epsilon$  is decreased, SLSQP and COBYLA both make quicker progress than the other two; however, COBYLA gets stuck in a local minimum. The gradient based method is able to make a significant drop in energy at the first step in all cases, which gives it a large lead. While SLSQP converges faster, it does not always reach the lowest energy, as is seen for  $\epsilon = 10^{-3}$ . For  $\epsilon = 10^{-4}$ , SLSQP produces the best results. Since 56 parameters are still small in number, i.e. when compared to the 50-qubits case which will have 2450 parameters, it is already evident that gradient based methods will converge much faster if they can be provided with good quality gradients despite the presence of noise. In conclusion, an overhead of at least three function evaluations per parameter will still use a much lower total number of function evaluations for the entire optimisation process as the problems are scaled up.

### Parameter shift rule

The overhead of at least three function evaluations per parameter per iteration can be further reduced to two if certain conditions about the ansatz are met. In Appendix B we show the

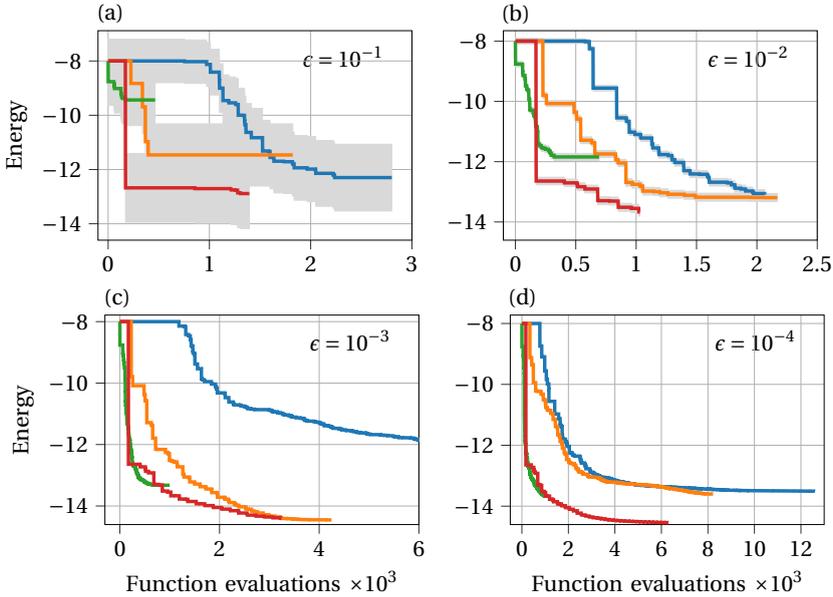


FIGURE 4.11: Same as Fig. 4.10 except that an 8-qubits model was used.

equivalence of this rule to our previous approach. To evaluate the gradient, the parameter shift rule [Sch+19; Cro19] can be used. If each operator in the ansatz satisfied Eq. (4.26), then the gradient is evaluated using the fact that if the curve fits to the sin function, then the derivative is calculable directly without any fitting. This has additional benefits. First, no ancilla qubits are required. This can allow for all the qubits to be used for the variational application. Second, a suitable change in the input parameters allows for the gradient to be computed. The number of samples required remains the same. We outline the process as follows.

Instead of fitting points to sin and using the cos with the same fitted values, it is more efficient to use

$$\nabla f_i = \frac{f(\theta_i + \frac{\pi}{2}) - f(\theta_i - \frac{\pi}{2})}{2}, \quad (4.31)$$

where only two evaluations per parameter are sufficient to get the gradient. Due to the fact that evaluation of  $\nabla f$  is possible by *shifting* the parameters while using the same circuit, Eq. (4.31) is named as the parameter shift rule. The gradient is then

$$\nabla \mathbf{f} = \left\{ \frac{f(\theta_1 + \frac{\pi}{2}) - f(\theta_1 - \frac{\pi}{2})}{2}, \dots, \frac{f(\theta_m + \frac{\pi}{2}) - f(\theta_m - \frac{\pi}{2})}{2} \right\}. \quad (4.32)$$

We tested the rule laid out in Eq. (4.32) and it did give accurate gradients for stochastic energy evaluations. The parameter shift rule allows an efficient evaluation of the gradient given that some conditions are fulfilled by the ansatz. In other terms, if some knowledge about the ansatz is available, through which one can ascertain that the rule is applicable, the variational optimisation can be sped up. This teaches us that additional information helps when using variational methods.

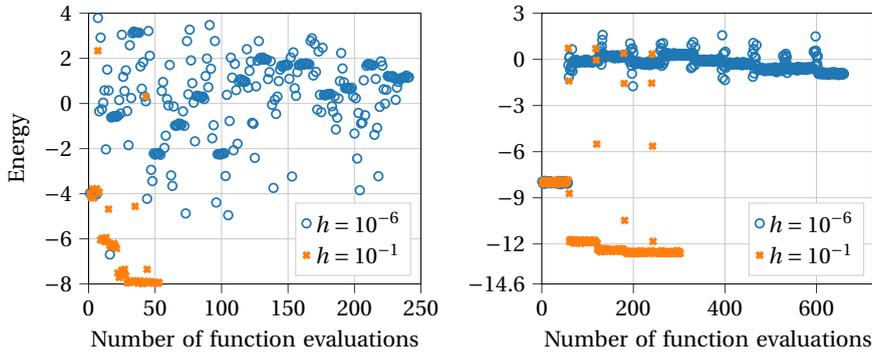


FIGURE 4.12: Energy optimisation progress against the number of evaluations using gradient based algorithms with two different step sizes  $h$ . **Left:** four qubits problem with 5 parameters. **Right:** eight qubits problem with 56 parameters.

#### 4.1.6 Results - Part II

On a quantum computer, the given  $f$  to be minimised becomes stochastic. The fundamental change in the optimisation problem occurs due to a change in the optimisation landscape. In other words,  $f$  is no longer a function in a strict mathematical sense since the domain  $x$  is not uniquely mapped onto the codomain  $f(x)$ , but instead to  $f(x) + \epsilon$ , where  $\epsilon \rightarrow 0$  only when the number of samples approaches infinity. This leads to problems in the gradient calculations if a finite difference formula is used. To better study the case, we can use an emulator to mimic a quantum computer by sampling from the underlying state vector.

We demonstrate that gradient based methods used until now would fail on a quantum computer. Figure 4.12 shows the optimisation progress of the 4 and 8 qubits isotropic Heisenberg antiferromagnetic Hamiltonians each having a parameterised ansatz containing 5 and 56 parameters, respectively. Each plot shows the energy evaluated at the  $k^{\text{th}}$  iteration using two different values of  $h$ . The  $h$  used in this section is a scalar and should not be confused with subterms of a Hamiltonian. The energies calculated using exact diagonalisation are  $-8.000$  and  $-14.604$ , respectively. The parameters in all four cases are initialised to zero. The dots in blue colour correspond to the case where  $\Delta f_k < \mathcal{E}$ , whereas the orange ones correspond to  $\Delta f_k > \mathcal{E}$  (see Eq. (3.59)). The optimisation progress for these two cases is substantially different. Using  $h = 10^{-6}$  gives  $\Delta f_k < \mathcal{E}$  and the gradient calculation fails to give the correct direction. This leads to no progress. Using  $h = 10^{-1}$  gives  $\Delta f_k > \mathcal{E}$  and the gradient calculation succeeds in finding a descend direction. The optimisation process for orange dots stopped itself, but the one with blue dots needed to be stopped manually since the optimiser was unable to decide on a stopping condition. Figure 4.12 demonstrates the importance of choosing the correct  $h$  for optimisation to progress towards a direction that decreases energy. However, this is not a solution to the problem because taking the step size large enough does not necessarily improve the gradient in general. Furthermore, the accuracy of the final energy is affected negatively.

At a given iterate  $x_k$ , the value of  $\Delta f_k$  is controlled using the finite difference  $h > 0$ . Thus, to use Eq. (3.59), we need to find  $h$  such that  $\Delta f_k(h) \gg \mathcal{E}$ . For the purposes of finding the gradient at iterate  $x_k$ , it suffices  $\Delta f_k$  to be an order of magnitude larger than  $\mathcal{E}$  due to the fact that a small change in the true value of the gradients still corresponds to a descend direction. Therefore, the task is to find

$$\min_h \Delta f_k(h) \geq g\mathcal{E}, \quad (4.33)$$

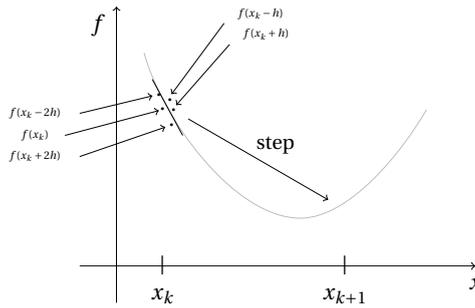


FIGURE 4.13: The curve represents the unknown function  $f$  and the dots are evaluations of  $f$  at and around the current iterate  $x_k$ . The least squares fit approximates the gradient for the evaluations of  $f$ . The fit is used as the gradient to take the step reaching the new iterate  $x_{k+1}$  where the process is repeated. Observably, using only  $f(x_k + h)$  and  $f(x_k)$  would give an inexact gradient.

which gives a good approximation to the gradient for  $g = 10$  and reduces  $f$  at the next iterate  $x_{k+1}$ . The accuracy of the gradients can be improved by demanding  $g \gg 1$ , but it should be noted that due to  $f$  being unknown, it cannot be guaranteed that  $\Delta f_k$  exists for  $g \gg 1$ . Equation (4.33) can be solved by starting at a very low value of  $h$  and increasing it at certain intervals, i.e. by an order of magnitude. While exactly solving Eq. (4.33) will give the most benefit, it is also to be considered that not a lot of computational effort is spent on it. Upon finding a suitable  $h$ , it can be kept constant until the optimisation stops making any progress. The optimisation process will stop making progress for the following two reasons. First, the assumption that  $\mathcal{E}_k = \mathcal{E}$  for all  $k$  breaks down. Second, Eq. (4.33) is no longer valid at the stopping iterate  $x_k$ . To proceed further, Eq. (4.33) can be solved again and the process can be restarted.

In the case of the Heisenberg model, gradient based optimisation algorithms make slow or no progress near the local minimum due to the fact that as  $x_k$  approaches a local minimiser, the gradient becomes smaller. Further progress requires smaller  $h$  which leads to smaller values of  $\Delta f_k$  which contradicts the assumption that  $\Delta f_k > \mathcal{E}$ . The two examples shown in Fig. 4.12 demonstrate that as long as the change in energy is much greater than  $\mathcal{E}$ , the gradients will be accurate, and each new iterate  $x_{k+1}$  will minimise the energy. When an optimiser reaches close to a minimum, where the difference in energy is closer to zero, the optimiser will stop making progress. While one choice of  $h$  makes the algorithms perform better than the other, it still does not lead to the ground state energy. Thus, it appears that no progress is possible without modifications to the current algorithm beyond a certain iterate.

The problem is the use of the forward differences formula, which depends on only two function evaluations. To access the true gradient of a stochastic  $f$  is the task that needs to be addressed. Such a task may be addressed by increasing evaluations of  $f$  to multiple points instead of two. Taking multiple function values around the iterate  $x_k$  and performing a least squares fit yields a better approximation of the gradient. The gradient at  $x_k$  using the linear fitting method evaluates the  $f$  at the points

$$f_r = f(x_k + rh), \quad (4.34)$$

where  $r \in [-n, n]$ , for  $n \in \mathbb{Z}$ . As an example, consider  $N = 2$  which takes five evaluations of  $f$

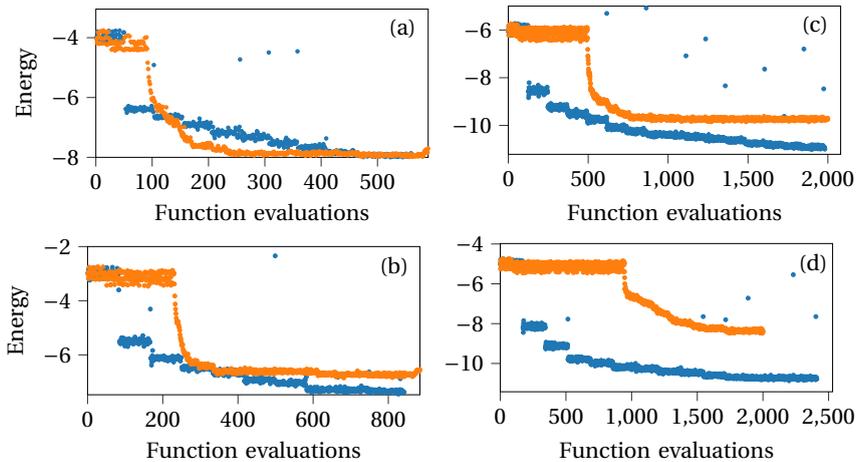


FIGURE 4.14: Optimisation progress of finding the ground state energy of the Heisenberg model using two different types of optimisation algorithms. SLSQP is shown in blue and BOBYQA in orange. Panels (a), (b), (c), and (d) correspond to 4, 5, 6, and 7 qubits problem sizes, respectively.

around an iterate  $x_k$ , as shown in Fig. 4.13. Clearly, the evaluated points will not necessarily lie on  $f$ . If the gradient was calculated using the finite difference formula, which uses evaluations  $f(x_k + h)$  and  $f(x_k)$ , the gradient would be almost zero, and the optimisation process would make no progress. However, the least squares fit to the five points closely approximates the true gradient and this helps in choosing the descend direction. After taking a step, the process can be repeated at the new iterate  $x_{k+1}$  until convergence is reached. The accuracy of the computed gradient can be improved at the cost of more evaluations of  $f$ . The drawback is that significantly more function evaluations are required. It is important to note that this technique will only work in the domain where Eq. (4.33) is valid.

Using the technique mentioned above, we compare the gradient based algorithm SLSQP to yet another gradient free algorithm called BOBYQA in Fig. 4.14. In BOBYQA, each iteration employs a quadratic approximation to the true (unknown) function [Pow09]. The panels show the energy evaluated at each function evaluation for Heisenberg rings of sizes 4, 5, 6, and 7 qubits. The problem Hamiltonian was isotropic with periodic boundary conditions. The XY ansatz was initialised with zeros as initial parameters. The gradient for SLSQP was approximated at each iterate using a 5-point linear fit around  $x_k$ . The XY ansatz in panel (a) contains 12 parameters, and due to the small parameter space, BOBYQA quickly overtakes SLSQP and reaches the minimum faster. However, in panel (b), as the parameters increase to 20, BOBYQA requires more evaluations of  $f$  at the initial point to proceed. As the number of parameters is further increased in panels (c) and (d), where there are 30 and 42 parameters, respectively, BOBYQA needs even more function evaluations at the initial point to lower the energy. Requiring a large number of evaluations is detrimental to the aim of reaching the minimum in the least amount of evaluations. The plot clearly shows that finding a quadratic approximation to  $f$  at each  $x_k$  is already very expensive at small problem sizes. Thus, if gradient based methods can be modified to be used with noisy functions, they are preferable to methods that use quadratic approximations of the function. It is important to note that practical applications of variational algorithms require significantly more parameters than demonstrated here in small prototype problems.

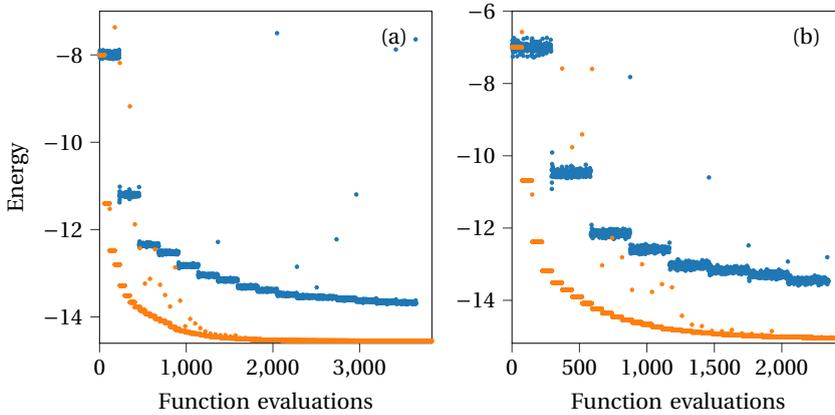


FIGURE 4.15: Comparison between the optimisation progress curves for the energies obtained using bitstrings (blue) and state vector (orange). (a) 8 qubits and (b) 9 qubits antiferromagnetic Heisenberg models.

Next we compare the SLSQP results from Fig. 4.14 with the corresponding results using the state vector. These are shown in Fig. 4.15. Panels (a) and (b) compare the optimisation progress from the quantum computer emulator (in blue) and the state vector simulator (in orange) for 8 and 9 qubits isotropic Heisenberg models, respectively. In both cases of the model, a five point linear fit reflects directly in the number of function evaluations required to go to the next iterate  $x_{k+1}$ . In the 8 qubits case shown in panel (a), the number of parameters were 56, and therefore, the gradient for the state vector case is calculated using at least  $m + 1$  (57) function evaluations. However, for the same case, the emulator requires at least  $4m + 1$  (225) function evaluations. Thus, the blue "staircase steps" in panel (a) are four times the length of the orange ones in (b). A similar argument is valid for panel (b) where the number of parameters were 72. In both the 8 and 9 qubits cases, it is observable that the first few iterations have an almost same drop in energy. Additionally, we observe that some energies do not lie on the staircase-like steps due to the gradient based optimisation algorithm seeking to find a suitable step size. Given that the same optimiser was used, the same drop in energy highlights that the gradient computation using the five points linear fit is approximately accurate to the gradient computed using the state vector simulator. The optimisation curves would look the same if the gradient was exactly the same at each step. It is also observed that the drop in energy is less in magnitude for the quantum computer after the first few iterations. One reason for this slow progress is as follows. As the parameters approach a minimum, the relative change of the parameters between two consecutive iterations decreases. Further progress then may require more accurate gradient computations which can be achieved using more points for the linear fit and further reducing  $h$ . In conclusion, good approximations to the gradient are essential for the optimisation progress of gradient based algorithms and in the case of stochastic functions, such gradients may be computed at the cost of increasing the number of function evaluations.

### 4.1.7 Discussion

#### Relevant literature

Beyond what has been presented in this chapter, work has been done in implementing the Heisenberg model on quantum computers and emulators. While some simulations exploit symmetries in the Hamiltonian [SSY20], others use e.g. a hardware efficient ansatz [Kan+17]. Qubit efficient architectures [Hug+19] have been used to reduce the number of qubits required and used for a  $4 \times 4$  square lattice with 6 qubits and about 97% fidelity which would otherwise require 16 qubits [Liu+19]. Effects of noise on the simulations of the model have been studied [Zen+21]. The Heisenberg model has been used as a benchmarking test bed for various methods [KB22; OGB21].

#### Classical competitors to quantum variational methods

While we have focussed on variational methods for quantum computers, many methods implemented on classical computers have been successful in simulating the Heisenberg model. We briefly mention some of them. A Green's-function Monte Carlo (GFMC) method [Kal62] has been used to compute the staggered magnetization [Run92b] and ground state energy [Run92a] in a two dimensional antiferromagnet on a  $12 \times 12$  lattice.

Several numerical results from various methods are available for one-dimensional lattices with and without periodic boundary conditions for comparison. For example, a chain of  $N = 20$  spins with periodic boundary conditions using variational Monte Carlo (with  $10^{10}$  iterations) gives  $E/N = -1.694916(12)$  [PF10] as compared to the variational energy using the XY ansatz  $E_f/N = -1.7671$ . A method closely related to GFMC reported  $E/N = -1.7796(8)$  [BD88]. The ground state energy is  $E_0/N = -1.7809$  [Fen]. The 20 spins case is itself a reasonable approximation to the infinite lattice limit  $-1.7726$  given by the Bethe ansatz. Monte Carlo methods suffer from the well known sign problem for two and three dimensional frustrated models [De 93] but are nevertheless useful [DDL84].

Simulations of the ladder lattices using the density-matrix renormalisation group technique [Whi92] have been performed for lattices of size up to  $160 \times 2$  [ABN19] but have had only limited success in two dimensions [PF10]. Quantum Monte Carlo simulations have been done for ladders of sizes up to  $200 \times 2$  [Wes+17]. It is not currently possible to use the variational methods discussed in this dissertation on emulators for such large lattices. However, a ladder of size  $10 \times 2$  with open boundary conditions has been simulated to give  $E_f/N = -2.231$  and compared to the ground state energy of  $-2.246$  found using the Lanczos method [Fen].

Two dimensional lattices pose a tougher challenge as the analytical results are unknown and the numerical methods are difficult to scale up. Exact diagonalisation has been done for up to 40 spins [RS10] a decade ago. A variant of Monte Carlo type simulations used in [BS88] for the  $4 \times 4$  and  $6 \times 6$  lattices with periodic boundary conditions give  $E/N = -2.8100(24)$  and  $E/N = -2.7260(28)$ , respectively, as compared to variational energies per spin obtained in this chapter  $E_f/N = -2.6403$  and  $E_f/N = -2.6304$ , respectively.

## 4.2 Quantum chemistry

### 4.2.1 Introduction

This section explores ways to use quantum computers to solve chemical problems. This involves describing the problem, converting the problem to be used in an algorithm designed for quantum computers, and running that algorithm to extract the desired output. The extent to which quantum computers will help solve chemical problems has been extensively

reviewed in [Cao+19]. Prototype applications of finding ground state energies of a chain of twelve hydrogen atoms have been demonstrated on superconductivity based quantum computers [Aru+20], among others like water [Nam+20] or small molecules [Hem+18; McC+19].

In the following sections, we revisit some of the basic principles involved in finding the ground state energies of Hamiltonians for chemical problems on quantum computers. We demonstrate the working by finding the ground state energy of the water molecule and study the effect of introducing errors in the computation of it. We present a prototype use case for finding the dissociation energy of Li-Li. Note that the energy of the Hamiltonians within this section is given in the Hartree units.

### Describing the problem

The formulation of a problem to be solved begins with knowing the physical configuration of an atom or a molecule of interest. This configuration includes information regarding the atom(s), the charge, the spin, physical units to be used, the basis set, interatomic distances, etc. Once the configuration is known, it becomes possible to calculate properties related to the molecule, e.g. nuclear repulsion energy, number of orbitals, nuclear dipole moment, orbital energies, etc., and are classically tractable. These calculations are not performed on a quantum computer but are stored and often recombined with calculations performed on the computer.

The relative coordinates of the atoms or molecules can be used for calculating the following properties. One- and two-body integrals in the molecular orbital basis, dipole integrals, molecular orbital coefficients, Hartree-Fock energy, Nuclear repulsion energy, etc. Preprogrammed packages are available to perform these calculations. In this dissertation, we used the Python-based simulations of chemistry framework (PySCF) package [Sun+18]. Alternatives are available [Fri+16; Par+17]. Some other properties besides the five mentioned above, for example, the total number of orbitals, number of active, frozen, or empty orbitals, are used together for the next step.

### Converting the problem

After the problem description and some elementary calculations, the next step is to make the problem quantum computer compatible. In the literature, this step is also termed as *translation* or *mapping*. The description of the problem is originally in terms of the annihilation and creation operators, also called second quantisation or the fermionic language. The task is to convert the second quantised Hamiltonian into a spin Hamiltonian. There are many ways to do this, e.g. by using the Jordan-Wigner transformation [JW28], Bravyi-Kitaev transformation [BK02; SRL12], Parity transformation [Bra+17], Bravyi-Kitaev Superfast (BKSF) algorithm [SW18], and various others [HTW17; VC05]. One conversion method may be better for one molecule and worse for another molecule. For example, one study estimates that the Bravyi-Kitaev method would have a reduced quantum computational cost than the Jordan-Wigner method for systems with more than 32 spin orbitals [Tra+18]. We briefly introduce the Jordan Wigner transformation below.

### Jordan Wigner transformation

We describe the Jordan Wigner transformation in part as discussed in [SRL12]. The transformation involves two basic tasks. First, the information pertaining to the number of fermions is stored on the lattice. The task is based on what is termed as *occupational number* basis or representation, where, as the name suggests, we let the state of a qubit store the occupational

number of a corresponding orbital. Second, the fermionic operators are mapped to spin operators. Once the states are stored, we need to convert the fermionic creation and annihilation operators to spin operators that will act on qubits in the same way that fermionic operators would have acted on the fermions. These let us encode fermionic states onto qubits.

Here we outline how the second task can be carried out. Assume that we wish to have a set of operations that perform the operations

$$\begin{aligned} Q^+ |0\rangle &= |1\rangle & Q^+ |1\rangle &= 0 \\ Q^- |0\rangle &= 0 & Q^- |1\rangle &= |0\rangle, \end{aligned} \quad (4.35)$$

where  $Q$ 's are the qubit operators we seek to represent in terms of fermionic operators. We need to make sure that  $Q$ 's also obey the fermionic anticommutation relations

$$\{a_i^\dagger, a_j^\dagger\} = \{a_i, a_j\} = 0 \quad \text{and} \quad \{a_i^\dagger, a_j\} = \delta_{ij} I. \quad (4.36)$$

The set of Pauli operators

$$\begin{aligned} \sigma^+ &= \frac{1}{2}(\sigma^x + i\sigma^y) \\ \sigma^- &= \frac{1}{2}(\sigma^x - i\sigma^y), \end{aligned} \quad (4.37)$$

fulfil that criterion when we represent them as

$$Q^+ = \sigma^- \quad \text{and} \quad Q^- = \sigma^+. \quad (4.38)$$

It is easy to recognise that  $Q^\pm$  anti-commutes with  $\sigma^z$ . If we represent the action of the fermionic operators with  $Q_j^\pm$  in Eq. (4.38) and of  $\sigma^z$  on all qubits with index less than  $j$ , the qubit operators will obey the relations of Eq. (4.36). In other words, the states of our qubits will have the same phases when acted upon by qubit operators as do the electronic states of a molecule when acted upon by the corresponding creation or annihilation operators [SRL12]. The effect of the action of  $\sigma^z$  on qubits is to handle the parity associated with the fermions by introducing a phase change of  $-1$  if the parity of the set of qubits having index  $< j$  is 1 (odd), or  $+1$  (no change) if it is 0 (even). The parity of a set of qubits is defined as the sum (modulo 2) of numbers that describe the states they are in. The fermionic operators are then given in terms of qubit operators as

$$a_j^\dagger = 1^{\otimes n-j-1} \otimes Q^+ \otimes [\sigma^{z \otimes j}] \quad \text{and} \quad a_j = 1^{\otimes n-j-1} \otimes Q^- \otimes [\sigma^{z \otimes j}]. \quad (4.39)$$

### Preparation timings

Before a quantum chemistry problem can be solved on a quantum computer, it needs to be formulated in machine readable format and transferred to the computer. Additionally, part of the computational effort can be performed on a classical computer by using approximations such as frozen orbitals. There exist computational packages [Sun+18] that take as input the geometrical structure of the atom or molecule and output the corresponding Hamiltonian in a suitable basis. The geometrical coordinates are input along with the desired basis to PySCE.

Table 4.3 show timings for different calculations involved. For each molecule, we use close to minimal bases that are useful for formulating prototype problems. It is these bases that decide the number of qubits required for the task. The circuit obtained from the unitary coupled cluster ansatz gives the number of parameters. The time taken to perform all the tractable classical calculations for each molecule are given under the Transpile column. These

Molecule	Basis	Qubits	Parameters	Transpile (min)	OpenMP (min)	MPI (min)
OHOHO	STO-6G	24	678	5.8	15.3	3.5
NCHOH-	STO-6G	24	1427	12.0	76.6	11.0
Li <sub>2</sub>	631G	28	63	3.9	421.8	24.9
Be <sub>2</sub>	631G	28	262	5.7	457.6	25.9
N <sub>2</sub>	631G	28	1083	14.2		28.6
O <sub>2</sub> -	631G	28	1419	17.9		30.7
CN-	631G	29	2127	24.3		114.5
NC-CN	STO-6G	29	2815	30.6		124.4
OHO-	631G	32	4201	33.3		845.9

TABLE 4.3: Preparation and runtime estimates for molecules.

calculations include, e.g. generating the Hamiltonian, calculating nuclear repulsion energy, creating the circuit, etc. Next, all of the ingredients required to perform a variational calculation are combined to calculate the energy of the Hamiltonian once. These numeric values will later be controlled by the optimisation algorithm in the variational method, which is not performed here. The time taken to calculate the energy once, i.e. for one set of numeric parameters, is done on either one compute node using OpenMP [DM98] and multiple compute nodes using MPI [For94] protocols and is shown in their respective columns. The energy is calculated using JUQCS on 24 cores using only one node (OpenMP) and using 1536 cores on multiple nodes (MPI) on the JURECA supercomputer [Jur]. This data has been previously published in [Mic20].

The data in the table shows that the classical part of the calculations remains tractable (see Transpile column). In contrast, the time taken to calculate the energy even once grows significantly with the size of the molecule. This was expected and it confirms why we need quantum computers. Additionally, the standard unitary coupled cluster ansatz parametrises the circuit with a growing number of parameters. These two combinations make the prospect of emulation to find the ground state energy of these molecules infeasible. The gradient based quasi-Newton algorithms would require energy evaluations equal to one plus the number of parameters to complete one iteration. Gradient free methods would most likely require more. Several thousand iterations can be expected to be required to find a ground state energy within chemical accuracy. Therefore, although the number of qubits required to represent such molecules is within the memory capabilities of current supercomputers, the number of energy evaluations remains prohibitively large. Furthermore, the bases used in the table are not necessarily the most accurate for estimating the ground state energies measured in experiments. These bases are used to demonstrate prototype applications. More accurate energies can be obtained by increasing the basis sizes, thus leading to even more qubits, parameters, and total time required.

## 4.2.2 Water molecule

We consider the geometry of the water molecule as shown in Fig. 4.17. We use the basis set "STO-6G" to calculate its ground state energy [HSP69]. The STO is an acronym for Slater type orbitals. An extensive list of basis sets and their usages is found in [You01]. We generate the unitary coupled cluster singles and doubles (UCCSD) ansatz [Ana+22], which contains 30 parameters using Qiskit. The water molecule in the selected basis is mapped into a problem requiring eight qubits. The initial state is prepared according to the Hartree-Fock state. All the parameters were initialised to zero to take advantage of this initial state. We observe

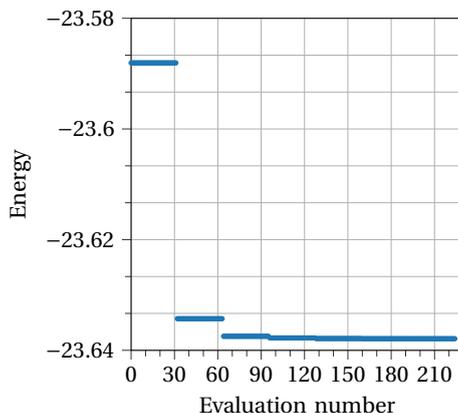


FIGURE 4.16: Optimisation progress of finding the ground state energy of the water molecule.

that due to this initial state, the optimisation algorithm only requires a few iterations before convergence.

The progress of the optimisation is shown in Fig. 4.16. The final energy after the optimiser signalled convergence was found to be  $-23.6379$ . Note that this is the variationally optimised energy of the active orbital space. The ground state energy found using exact diagonalisation is  $-23.63799$ . The total energy  $-75.729$  is found by adding to the final energy the frozen orbital energy ( $-61.2846$ ) and the nuclear repulsion energy ( $9.1939$ ), respectively. The ground state energies of molecules are always calculated in a prescribed basis. It is difficult to compare such energies across different bases since two bases are rarely on an equal footing. For example, even with the STO basis there can be further characterisations such as *double zeta* or *triple zeta with two polarisation functions*, etc. The full configuration interaction or exact diagonalisation of the water molecule in an STO basis of double zeta polarised basis gives  $-76.07$  [VDN94] which is clearly different from STO-6G.

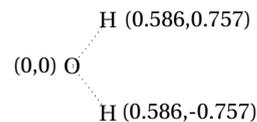


FIGURE 4.17: Configuration of the water molecule given in  $(x, y)$  coordinates.

### Simulations with errors

We investigate the effects of introducing certain errors on the variational quantum eigen-solver. The errors are introduced by adding different gates to the original circuit. In quantum computers, errors are broadly categorised into the following categories: bit flip and phase flip channels, depolarisation channel, overrotations, amplitude damping, and phase damping [NC10]. In this section, bit flip and phase flip errors and overrotation errors have been implemented. Some of the error categories are challenging to implement (e.g. depolarisation channel) due to the large number of circuits involved. Due to the large number of computing resources required, we have instead focused on what was implementable in a reasonable amount of computing time, which restricts us to an average of only a few cases of the instead of a very large number of them.

The errors are introduced in the quantum circuit used to calculate the ground state energy of the water molecule. The quantum circuit can find the ground state energy within chemical

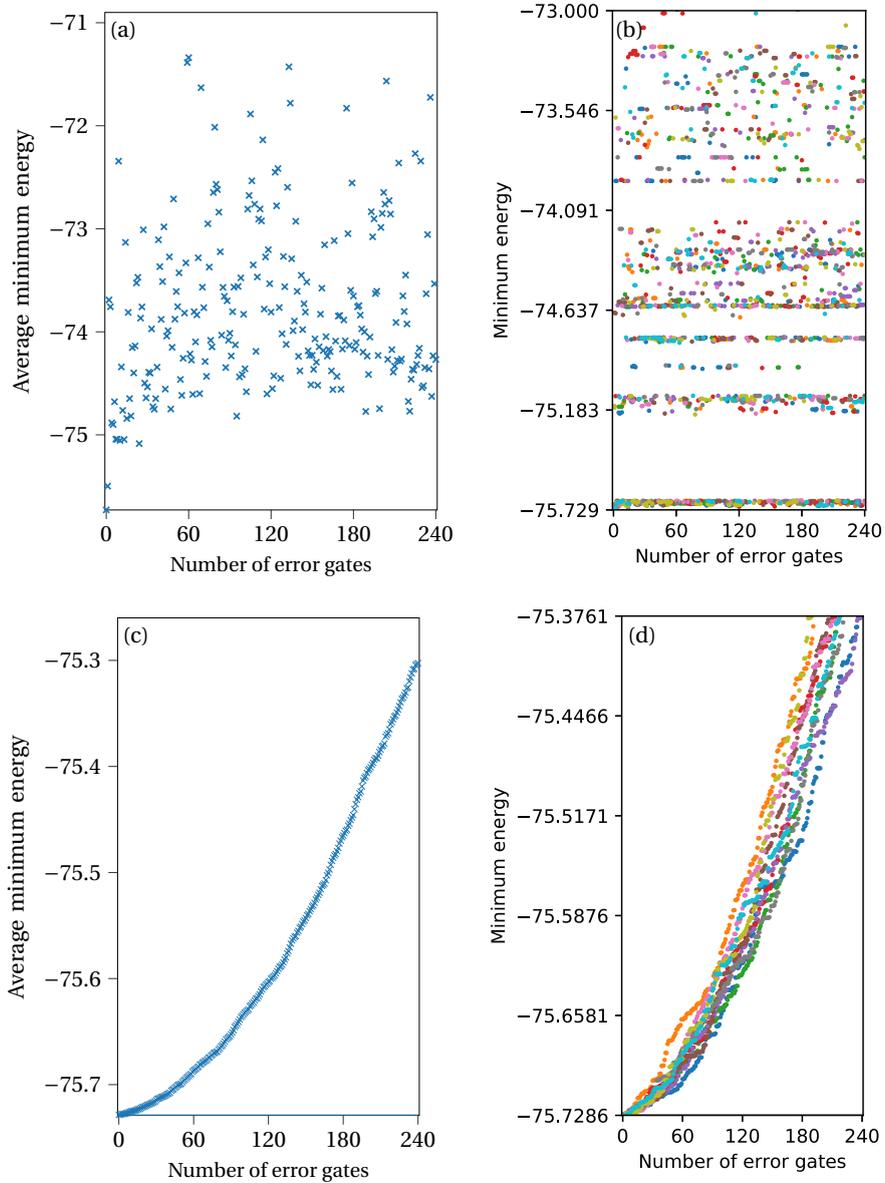


FIGURE 4.18: Final variational energies as a function of increasing number of gate errors added to the quantum circuit. (a) Average of energies obtained in ten runs after randomly adding  $X$ ,  $Y$ , and  $Z$  gate errors. (b) All final energies in the ten cases from panel (a) are shown and colour coded. Energy gaps are observed. (c) Average of energies obtained in ten runs as a function of increasing overrotation errors; (d) All final energies in the ten cases from panel (c) are shown and colour coded.

accuracy when no errors are introduced. To introduce errors, we added additional gates as follows. For the bit flip errors, the flip gate  $X$  is randomly added to the circuit at uniform intervals. Similarly, we add the gate  $Z$  for the phase flip error, and for the bit and phase flip error, we add the  $Y$  gate. All these  $X$ ,  $Y$ , and  $Z$  gates are randomly chosen, each with a probability of one-thirds and added at a random location in the quantum circuit. For overrotations, the extra gates added are  $R_x$ ,  $R_y$ , and  $R_z$ , or namely, the rotation along the  $X$ -axis,  $Y$ -axis, and  $Z$ -axis, respectively, with a fixed (but configurable) small angle (e.g. one degree) of rotation. In an error prone simulation, in principle, errors may appear at any time during the execution of a circuit; however, owing to simplicity and limited computational resources, we restrict the errors to appearing at only certain depths of the circuit at uniform intervals over the circuit. The number of errors appearing in a circuit is controlled, but the placement in the circuit is kept random. The seed for the randomness generator is changed ten times leading to ten different placements each time.

## Results

Results for the simulations when introducing the errors are shown in Fig. 4.18. Panels (a) and (b) show the minimum energies obtained as a function of the total number of  $X$ ,  $Y$ , and  $Z$  gate errors added to the quantum circuit. Panel (a) shows the average, while panel (b) shows each case in ten different error gate configurations each denoted by a different colour marker. The lowest point of the  $y$ -axis is the ground state energy calculated using exact diagonalisation. We observe the emergence of levels where the finally obtained energies appear to be accumulated. Certain energies are never reached when the optimisation algorithm converges to a minimum. The left plots show the average of each of these ten random seeds. Interestingly, even though a large number of points appear at the bottom in the right plot, the calculated average over ten different runs shows that such cases are rare given that the average is much higher. The minimum energies obtained were between  $-71.339$  and  $-75.729$ . The best result is obtained when no error gates are introduced. Even a few error gates increase the minimum obtained significantly.

In panels (c) and (d) in Fig. 4.18, we show the results when introducing the overrotation gates as errors. Panel (c) is the average of each of the ten runs shown and colour coded in panel (d). Each overrotation gate involved an angle of 1 degree or approx. 0.017 radians. We observe a completely different effect than that visible in panels (a) and (b). The minimum energy obtained appears to be monotonically increasing as the number of error gates is increased. The levels of energy obtained in the previous case have disappeared.

We conclude that while variational methods are able to tolerate errors, the impact on the accuracy sensitive results may be undesirable.

### 4.2.3 Dissociation of Li-Li

One measure of the bond strength of atoms in a molecule is the dissociation energy. Thus, a prototype application is to calculate the dissociation energy of molecules. We investigate the dissociation of Li-Li. In this section, we go through the ingredients to set up the problem. Often in quantum chemistry, approximations are performed in the formulation of a problem. One such approximation is called core freezing. In core freezing, electrons at the core of the atom or molecule are left out of the calculation due to insignificant contributions. In essence, it is the electron orbitals which are "frozen". Undertaking such an approximation can significantly simplify the problem's difficulty without a significant effect on the description of the properties of the system.

We used the PySCF package to automate the generation of a Hamiltonian as well as core freezing. For small systems like the Li atom, it remains possible to perform calculations with and without core freezing. When freezing the core of the Li atom, the active orbital space is mapped to a problem requiring 13 qubits. The energy of the frozen core is then  $-7.2355$ . After using the UCCSD ansatz, the optimised variational energy was found to be  $-0.1958$ . The ground state energy of the Li atom is given by

$$E_0^{\text{Li}} = -7.2355 - 0.1958 = -7.4312. \quad (4.40)$$

We found the same energy without freezing the core, which was mapped into a problem requiring 15 qubits and more terms in the Hamiltonian. Since Li-Li will dissociate into two Li atoms each in the ground state energy Eq. (4.40), the total energy of both the atoms will be double that of each individual atom. The Li-Li Hamiltonian described a 28-qubits system. The frozen orbital energy of the molecule is  $-16.0546$ . In addition to the frozen orbital energy, the molecule has a classically efficiently computable nuclear repulsion energy of  $1.7817$ . Using the UCCSD ansatz, the ground state energy of active orbitals of Li-Li was found to be  $-0.5483$ . The total ground state energy is given by

$$E_0^{\text{Li-Li}} = -14.8211. \quad (4.41)$$

### Dissociation energy

Using the ground state energies of Li-Li and Li one can calculate the bond dissociation energy by subtracting the energies of the individual atoms from the energy of the molecule,

$$E_d = E_0^{\text{Li-Li}} - 2E_0^{\text{Li}} = 0.0413. \quad (4.42)$$

Experiments have found the dissociation energy to be close to the above value, e.g.  $0.0377 \pm 0.0002$  [VOZ69] or  $0.0400$  [Luo07]. The estimates vary depending on the experimental method [Luo07]. It must be noted that we did not verify if both the energies were within chemical accuracy. Therefore, our estimate may be only accidentally accurate despite the reasonable value of the dissociation energy.

## 4.3 Hubbard model

### 4.3.1 Introduction

The Hubbard Hamiltonian arises naturally by considering a simple description of the motion and interactions of electrons on a lattice. Given a certain lattice, by using the Born-Oppenheimer approximation [BO27] the lattice points describe the stationary atoms. Most of the complexity of the atom is taken away by considering that it has only a single level or orbital. Thus, one site in a lattice accommodates at most two electrons and is described by at most four configurations: empty, one up electron, one down electron, or a pair of up and down electrons. For a solid with only one energy band at the Fermi surface, this simplification with only one orbital is already relevant [Sca16].

Electrons possess kinetic energy. This energy can be modelled by considering a kinetic term  $t$  which makes the electrons hop from one site to another. In the language of creation and annihilation operators, this is the equivalent of the electrons disappearing from one site and appearing on another. For simplicity, the Hubbard Hamiltonian only considers hopping between the nearest sites on the lattice. Furthermore, the electrons in the lattice interact via the repulsive Coulomb interaction. For simplicity, the Hubbard Hamiltonian only considers

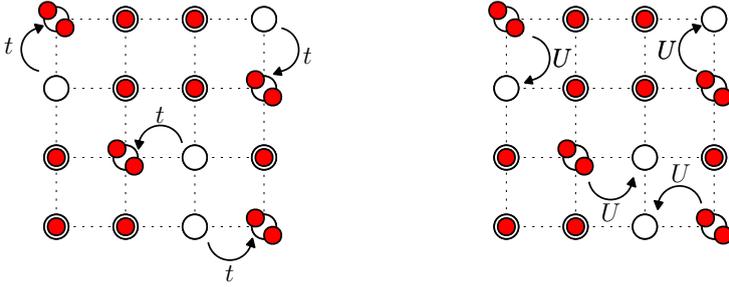


FIGURE 4.19: Pictorial representation of the kinetic and Coulomb potential terms on a lattice half-filled with electrons. The left picture depicts how  $t$  enables electrons to hop from one site to another. The right picture shows how  $U$  is turned on when there is double occupation on a site.

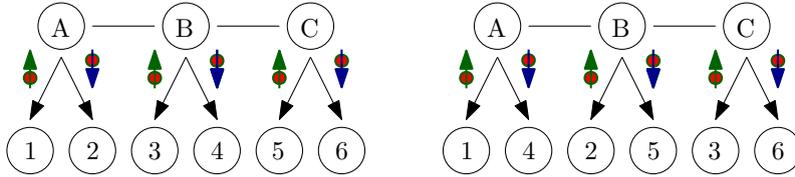


FIGURE 4.20: Mapping of fermionic lattice sites A,B,C to qubit indices for (left) odd-even scheme and (right) up-down scheme.

this repulsion if the electrons are on the same site. This interaction is modelled by an on-site repulsion term  $U$ . Thus,  $U$  is zero if the site is empty or has only one electron, and it is positive if the site is doubly occupied. Although this is the simplest case, extensions of the model also include interaction between electrons of the neighbouring atom and so on (see Ref. [LH87; EN07; MZ93]). A pictorial representation of both the terms is given in Fig. 4.19.

Formally, the Hamiltonian is given by

$$H = -t \sum_{\langle i,j \rangle, \sigma} (a_{i\sigma}^\dagger a_{j\sigma} + a_{j\sigma}^\dagger a_{i\sigma}) + U \sum_i n_{i\uparrow} n_{i\downarrow}, \quad (4.43)$$

where  $n = a_{i\uparrow}^\dagger a_{i\uparrow}$  is the number operator and the symbol  $\langle i,j \rangle$  indicates that the hopping takes place over the nearest neighbours only. The mutual interplay of different values of  $t$  and  $U$  can give rise to a number of interesting physical phenomena.

### Translation

Schemes to transform a fermionic Hamiltonian to a spin Hamiltonian were described earlier. Before the transformations can be used, it is necessary to identify a layout of the mapping. In all schemes, the total number of qubits required is twice the number of sites on the fermionic lattice. Since each fermionic site requires two qubits, two mapping layouts are common for the Hubbard model. We name them the *odd-even* and the *up-down* schemes. We use the OpenFermion package to implement both [McC+17b]. They are briefly described below.

- **Odd-even:** Assume that the fermionic lattice sites are enumerated  $i = 1, 2, \dots, n$ , and the qubits are enumerated  $j = 1, 2, \dots, 2n$ . Then the odd-even scheme refers to the mapping

such that each  $i^{\text{th}}$  fermionic site is mapped to qubits with the indices  $j = 2i$  and  $j = 2i - 1$ . Since each fermionic site occupies an odd and even indexed qubit, the scheme is named odd-even. It is a matter of choice if the spin-up term is given the odd or the even index.

- **Up-down:** Assume again that the fermionic lattice sites are enumerated  $i = 1, 2, \dots, n$ , and the qubits are enumerated  $j = 1, 2, \dots, 2n$ . Then the up-down scheme refers to the mapping such that each  $i^{\text{th}}$  fermionic site with the spin-up term is mapped to qubits with the indices  $j = i$  and with the spin-down term with  $j = n + i$ , or vice versa, as per chosen convention. Since each fermionic site occupies the qubit indices based on spin, the scheme is named up-down. Both the schemes are illustrated in Fig. 4.20.

### 4.3.2 Ansatz

In finding the ground state energy of the Hubbard model, certain symmetries need to be respected. One of them is particle conservation symmetry. The task in solving the Hubbard model is often to find the ground state energy given a certain filling of the lattice, which remains conserved at all times. Most often, the model is considered for half filling. The XY ansatz is unsuitable as it does not necessarily respect the particle conservation symmetry of the problem. This section outlines an ansatz suited for the Hubbard model.

#### Variational Hamiltonian Ansatz

The adaptive method used to build the ansatz for the Heisenberg model was met with limited success when used for the Hubbard model. An ansatz inspired by the Hamiltonian itself [WHT15] has shown reasonable success under certain conditions [Cad+20]. It has since been widely used, e.g. in solving the Hubbard model [MS20] or the Heisenberg model on the kagome lattice [KW21; BM22], as well as observed for having fewer local minima upon overparametrisation [Wie+20]. We introduce the variational Hamiltonian Ansatz below.

As described in Eq. (3.34), the Hamiltonian is often a sum of terms

$$H = \sum_j c_j h_j, \quad (4.44)$$

and not all terms necessarily commute amongst each other. For simplicity, and without loss of generality, assume that all terms in  $H$  are non-commuting. Then a  $p$ -depth variational Hamiltonian Ansatz is given by

$$U(\theta) = \prod_{l=1}^p \left( \prod_j e^{-i\theta_{j,l} h_j} \right), \quad (4.45)$$

which is again a set of operators defined by each  $h$ . This ansatz is inspired by adiabatic evolution. When operators commute, sets of commuting operators replace  $h$  and are assigned a corresponding single parameter. Given the set of operators, this ansatz can be used to construct circuits similar to those previously simulated for the Heisenberg model. Note that the parameter is not always placed on the last qubit in this case.

The sets of commuting terms depend on the configuration of the lattice structure. On two dimensional square lattices, often, five sets are sufficient to accommodate all the operators such that operators in each set all mutually commute [Cad+20; Rei+19]. Four of these represent the hopping term and the fifth is the on-site interaction. We investigate periodic one

dimensional lattices where five sets are also sufficient. An advantage of the variational Hamiltonian Ansatz is the small number of parameters per step.

### Number preserving ansatz

While the variational Hamiltonian Ansatz intrinsically conserves the occupation number or the particle conservation symmetry, one can imagine other operators not present in the Hamiltonian that also conserve it [Cad+20]. As the name suggests, a number preserving ansatz is any ansatz that respects the particle symmetry, or any other symmetry that may be relevant [Gar+20]. A number preserving ansatz can be considered a general case of the variational Hamiltonian Ansatz.

#### 4.3.3 Initial state

Part of the reason for the success of finding the ground state energy of the Heisenberg model was the ability to start from a low initial energy state, the Néel state. In the Hubbard model, one can take advantage of the interplay of the kinetic or on-site interaction terms by setting either one to zero. The state of the resulting Hamiltonian can then be prepared on a quantum computer. One can intuitively expect doing so to be advantageous over a random initialisation of parameters.

In the context of the Hubbard model, the initial state does not only refer to the initial computational states of the qubits. If interpreted in the same way as in the Heisenberg model, the initial state is fixed and expected to be specified in the problem itself. For example, a potential task might be to find the ground state energy when the Hubbard lattice is half filled. The computational states would then be set with half of the qubits in the state  $|1\rangle$  or  $|0\rangle$ , as per chosen convention, to "fill" the lattice. The filling will not change during the entire course of the variational optimisation. In contrast to the Heisenberg model, the initial state can also refer to some gate operations applied to the computational state. These are not part of an ansatz. Such operations, in combination with the initial computational states, are expected to deliver the same effect as the Néel state in the Heisenberg model. Below we discuss two ways to prepare such initial states for the Hubbard model.

#### Only kinetic terms or $U = 0$

The ground state of the non-interacting Hubbard model, or  $U = 0$ , is efficiently preparable using fermionic Fourier transform [VCL09; Jia+18], swap networks [Cad+20], or using Givens rotations [Jia+18]. Although the state is preparable, it is not a straightforward task, in contrast to the Néel state. Furthermore, the overhead of circuit depth [Cad+20] incurred when preparing these states should preferably be avoided in view of the low-depth capabilities of current quantum computers.

#### Only on-site terms or $t = 0$

By removing the kinetic terms, the Hamiltonian effectively describes a collection of single site lattices, whose ground state is also efficiently preparable. Similar to the Néel state, the occupational number of the Hamiltonian can be prepared by suitably setting (or filling) the computational initial states. In this case, the initial parameters of an ansatz may not always be set to zero as the initial point may be a local minimum. To avoid the minimum, the parameters may be set close to zero or to any other combination.

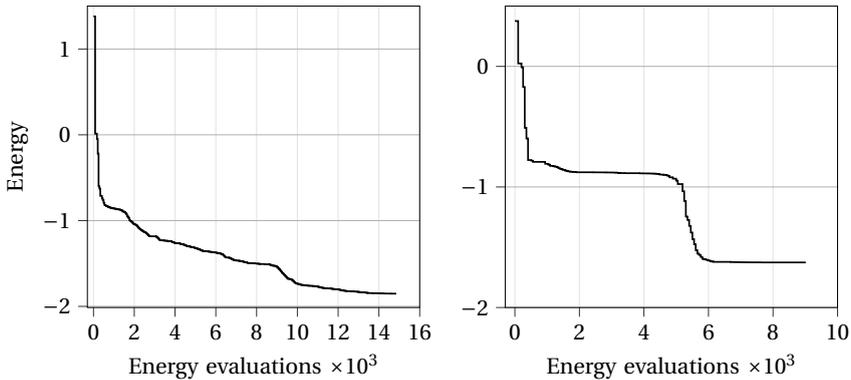


FIGURE 4.21: Energy progress curves for different lattices of the Hubbard model. The lattice sizes were  $6 \times 1$  (left) and  $16 \times 1$  (right). The ground state energies are not shown.

#### 4.3.4 Results

We used the variational Hamiltonian Ansatz to find the ground state energy of a one dimensional fermionic ring of length sixteen with periodic boundary conditions. The problem maps to a system with 32 qubits. We study the case of  $U = 4$  and  $t = 1$ . Using the Jordan Wigner transformation and an odd-even scheme, the converted Hamiltonian has 113 terms. The ansatz consists of ten steps,  $p = 10$ , where each step contributes five independent parameters. Thus, the total number of parameters was 50. Three of these five parameters in each step are attributed to the hopping terms, divided such that all terms in a set commute. The fourth and fifth parameters are given to the periodic boundary hopping terms and the on-site interaction terms, respectively. The initial state is set to fill eight spin up and eight spin down fermions without any additional gate operations, which is the  $t = 0$  case. We initialise all parameters with the value 0.025. These initial parameters give an initial energy of 0.376. The optimiser stopped making significant progress after a variational energy of  $-1.626$ , which can be thought of as a local minimum but cannot be strictly established as such. This estimate is far from the ground state energy, which is close to  $-7.64$  [Han].

We also investigated finding the ground state energy of a ring of length six with periodic boundary conditions. We use the variational Hamiltonian Ansatz with four parameters per step and a total of ten steps. Unlike the sixteen site lattice, we place the periodic boundary hopping terms with one of the other sets of hopping terms. Clearly, there is some freedom in using the variational Hamiltonian Ansatz. The parameters were initialised with the value 0.05. The final energy after the optimiser signalled convergence was  $-1.851$ . The ground state energy is  $-3.669$  [Han].

We observe that the variationally obtained energy in both cases is not a good approximation to the ground state energy. The variational method fails to find the ground state. We plot the energy progress curves for both the cases in Fig. 4.21. Although the energy initially drops quickly, similar to the Heisenberg model, we observe that the curves have some differences from the ones shown for the Heisenberg model. For the  $6 \times 1$  case, we observe several slowdowns and quick drops of the energy, and the most progress is no longer made initially, in contrast to the Heisenberg cases. The contrast is more visible for the  $16 \times 1$  case, where the optimiser appears to make relatively no progress from 2000 to 5000 energy evaluations, and thereafter the energy drops significantly. These observations give credence to the intuition

that the energy landscape may be non-convex and difficult to navigate.

### 4.3.5 Discussion

The results show that variational methods were not able to find the ground state energy for the Hubbard model, at least not in the configuration used. Here we discuss reasons and potential configurations that may find better results. The first reason is that the initial energy was much higher than the ground state energy. The larger the difference between the initial and the ground state energies, the more the traversing distance for the optimiser to find the global minimum. Traversing more distance on the landscape has more chances of getting stuck in a local minimum. One way to avoid this problem is to start from an initial energy relatively closer to the ground state energy. The initial state with  $U = 0$  offers a potential solution by providing a lower initial energy [Cad+20].

The second reason is the initialisation of parameters in conjunction with the initial state. Similar to the Néel state in the Heisenberg model, a random initialisation will not be successful. Since the variational Hamiltonian Ansatz is inspired by adiabatic evolution, the initial parameters have also been proposed to be initialised in a way that mimics it [Rei+19]. For example, by setting  $\theta = 1/t$  for the hopping terms and  $\theta = 1/pt$  for the interaction terms.

## 4.4 Conclusion

### Heisenberg model

We calculated ground state energies of the Heisenberg model using variational methods and compared them to the results known from other methods. We learnt that although simulating the model is theoretically possible for any variational ansatz and settings if good results are expected, several contributing factors need to be considered. The first factor is the choice of a suitable ansatz. An ansatz should not only have a good overlap with the ground state but should also produce an energy landscape that is navigable through optimisation algorithms. The aim is to avoid getting stuck in a local minimum where the energy is much higher than the ground state energy. Not only should the landscape have the value of the energy at the global minimum reasonably close to the ground state of the Hamiltonian, but it should also be possible to reach (close to) the global minimum within a reasonable number of energy evaluations. An ansatz inspired by applying an adaptive ansatz selection method to a small number of lattice sizes was invented and used for the model. This ansatz allowed us to find reasonable approximations to the ground state energies. One disadvantage of the ansatz is that it requires all-to-one qubit connectivity, which is difficult to achieve in practice on superconductivity based quantum computers; however, trapped ion computers offer this feature.

We observed that the limiting factor to finding the ground state energies of the Heisenberg model was not always the memory of the classical computer emulating the quantum computer. For the following reasons, even supercomputers cannot efficiently emulate quantum computers for variational methods. Large circuit depths and thousands of iterations required to find a good approximation to the ground state energy demand considerable resources. Additionally, although the number of parameters in the XY ansatz grows polynomially, the evaluation of the energy even once for lattices of size thirty five or more is already very expensive. This implies that only a few energy evaluations are possible within 24 hours, the time limit per job on the supercomputer [Juw]. Since quasi-Newton methods improve their approximation to the Hessian at each iteration, having very few iterations per job leads to worse performance. While this is a serious disadvantage for quasi-Newton methods, it can be fixed

by saving the internal parameters of the optimisation processes. This is a possible track for future projects.

## Quantum chemistry

The accurate description of a molecule depends on the basis selected to describe its orbitals. Finding the ground state energies of molecular Hamiltonians on a quantum computer requires a transformation of the second quantised Hamiltonian to the spin Hamiltonian. This task can be carried out using the Jordan Wigner transformation. The process of obtaining the ground state energy consists of two parts: tractable and intractable. The former includes e.g. frozen orbitals and the latter e.g. active orbitals. The former is calculated on the classical computer, and the latter is intended for the quantum computer. Our primary focus was on the latter. Since current quantum computers are not good enough, we performed the calculations on emulators. We used minimal or close to minimal bases for the prototype problems.

We analysed the time required to perform variational calculations for some molecules that required between 24 to 32 qubits. As expected, the tractable parts of the calculations could be performed in a reasonable amount of time. The intractable parts, e.g. the emulation of a quantum computer, required up to several hours for a single energy evaluation. Since the variational methods will require thousands of energy evaluations before converging to the ground state energy, these were not feasible and, therefore, not performed.

We demonstrated the use case of the water molecule. After calculating its ground state energy, we investigated the effect of gate errors introduced to the simulation. We found that although variational methods are resistant to errors, the accuracy of the energy is lost. For example, by adding gate errors, the obtained final energy was not within chemical accuracy. Finally, we demonstrated another use case by calculating the dissociation energy of the Li-Li bond.

## Hubbard model

Similar to the Heisenberg model and molecules, the Hubbard model is also a hard problem to solve. We investigated the use of quantum computers through an emulator and the challenges of calculating the ground state energy of the model. Similar to the molecular Hamiltonians, the Hubbard model is first converted to the spin Hamiltonian using the Jordan Wigner transformation. Each fermion requires two qubits. We discussed two choices: odd-even and up-down. Due to the considerations of particle symmetries, the ansatz had to be suitably chosen. We implemented the variational Hamiltonian Ansatz, which respects particle conservation on the lattice. Furthermore, there are two promising ways to set up the initial state which should provide a better initial energy to start the variational optimisation, in contrast to the initialisation of the parameters at random.

Our results showed that the variational configurations we used were unable to find the ground state energy. We discussed recent literature which highlights better configurations which may yield better results. We concluded that the variational setup needs to be fine tuned for each problem to have any reasonable chance of finding the ground state energy.

## Chapter 5

# Quasi dynamical evolution

### 5.1 Introduction

Variational methods have been one of the first promising prototype quantum computer applications. In the previous chapters, we demonstrated their use beyond the small scale. We also encountered potential problems. In this chapter, we highlight those problems and develop a *quasi dynamical evolution* heuristic to remedy them. The chapter is structured as follows. First, we numerically demonstrate the presence of local minima and barren plateaus in the energy landscape of selected Hamiltonians. Second, we propose the heuristic, which is intended to find the ground state energy of a given Hamiltonian systematically. We also discuss the connection of the heuristic to quantum annealing. Third, we present the results of implementing the heuristic on different Hamiltonians. Last, we discuss issues related to the computation of parities of the measured bitstrings and the computation time for an emulator against an ideal quantum computer. This chapter is based on results already published in [Jat+22b].

### 5.2 Problems in variational methods

#### 5.2.1 Local minima

Even prototype problems of relatively small size that require no more than a few qubits can exhibit multiple local minima around the global minimum in their multidimensional energy landscapes. This is a problem for variational methods. To numerically demonstrate the presence of such local minima, we implement and test two different problems. First, we find the ground state energy of a collection of one dimensional isotropic antiferromagnetic rings using the XY-ansatz. Second, we compute the ground state energy of the water molecule using different ansätze.

#### Isotropic antiferromagnetic Heisenberg model

Figure 5.1(a) compares the energies found using the variational quantum eigensolver against the ground state energies as a function of the problem size. The energies are found using the BFGS minimisation algorithm, which is restarted 100 times after it reports convergence. The parameters were assigned randomly in the interval  $[0, 2\pi)$  upon each restart. The ground state energies are calculated using exact diagonalisation, which is possible due to the small size of the systems. The error bars show the maximum and minimum values obtained in all 100 restarts. Interestingly, the even numbered lattices have wider error bars. The initial energies, obtained by setting the initial state to the Néel state, are shown in dotted lines. Overall, we observe that the variational energies drop below the initial Néel state energies and converge

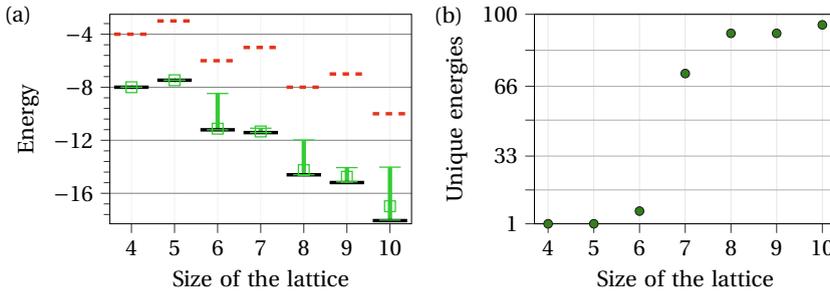


FIGURE 5.1: (a) Minimum energies found using the variational quantum eigensolver for the Heisenberg model averaged over 100 restarts (green squares) and their maximum and minimum values (error bars). Solid (dashed) black (red) lines are the ground (Néel) state energies for each lattice. (b) Percentage of unique energies found for different lattice sizes.

towards the ground state energies. The variationally obtained energies are, on average, close to the ground state energies.

Figure 5.1(b) shows the number of unique values of energy per 100 trials. We count the number of local minima by enumerating the unique energy values. The underlying assumption is that the unique values of the energy represent unique local minima. To accommodate for the fact that a local minimum may be a valley (in multidimensions) the energy values are rounded off to  $10^{-3}$ . Using our analysis, there is a possible undercounting of local minima as there may be different minima having the same energy. However, as the data shows, in the larger lattices a large number of local minima are present even when undercounting them. There is a sharp increase in the number of local minima for  $N \geq 7$ . Plot (b) shows that local minima pose a problem when finding the ground state energy of the Heisenberg model even for small size, proof of concept lattices.

### Water molecule

The energies found using the variational quantum eigensolver when restarting a thousand times using random initial parameters are shown in Fig. 5.2. Refer to section 4.2.2 for details of the Hamiltonian. The ground state energy of the active orbital space found using exact diagonalisation is  $-23.638$  Hartree, which is approximately  $-643.223$  eV. Figure 5.2(a) shows the data when using the unitary coupled cluster singles and doubles (UCCSD) and Fig. 5.2(b) when using a TwoLocal ansatz inbuilt in Qiskit [Ale+]. We observe in both cases the existence of unique values of energy that represent local minima. A large number of cases fall into the same local minima while others get stuck elsewhere. In Fig. 5.2(a), we observe the presence of a local minimum at  $-23.591$  Hartree in which a large number of trials get trapped. The energies of the rest of the local minima in the UCCSD ansatz are closer to the ground state energy. In Fig. 5.2(b), there are also many different local minima but they are not concentrated closer to the ground state energy. The obtained energies were not within chemical accuracy in any of the experiments using either ansatz. The aim of the results presented here was only to show the presence of local minima.

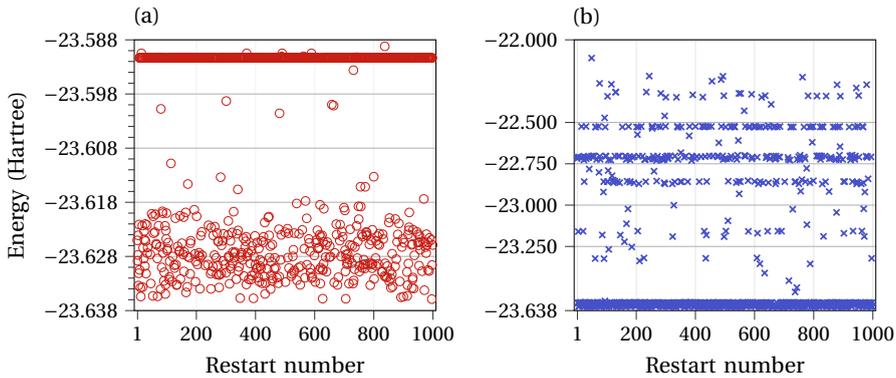


FIGURE 5.2: (a) Final energies obtained after 1000 restarts using the UCCSD ansatz. (b) Same as (a) except using TwoLocal ansatz.

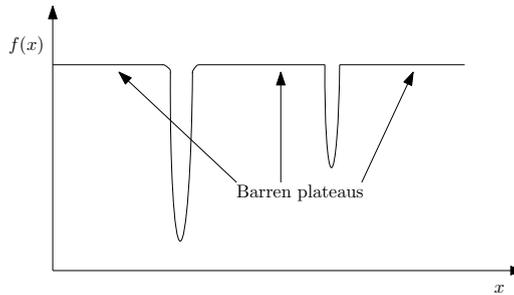


FIGURE 5.3: Picture showing emergence of barren plateaus in a one dimensional optimisation landscape.

## Summary

Figures 5.1 and 5.2 together show that multiple local minima exist even for prototype, small scale problems. Also, when the local minima exist, they often surround the global minimum. Therefore, it is vital that variational methods are able to escape or avoid local minima. Relevant research in this direction is limited and proposed improvements are resource expensive [WGK20].

### 5.2.2 Barren plateaus

An important factor for the success of a variational calculation is the starting point of the optimiser in conjunction with an initial state of the quantum computer. In general, it may not be possible to bring about a suitable combination of the two and one must rely on randomly choosing an initial state as well as parameters. It has been shown that such attempts will be detrimental for the success of variational methods as they will encounter what is known as barren plateaus [McC+18; ZG21; CNB21; Pat+21; Hol+22].

Barren plateaus refers to the phenomenon of gradients approaching zero on the optimisation landscape. When a gradient based optimiser is provided with an initial point and it

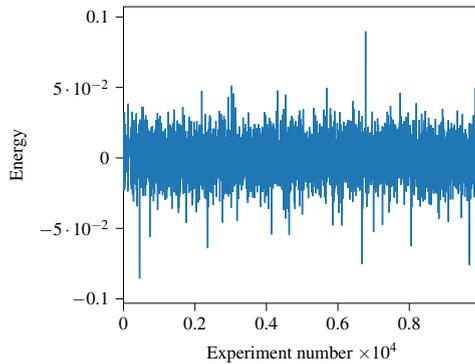


FIGURE 5.4: Energies for the  $3 \times 3 \times 3$  Heisenberg model obtained using  $10^4$  sets of random initial parameters.

tries to find the gradient at the point in order to seek a downhill or descend direction, it encounters a zero or close to zero gradient. A pictorial representation is drawn in Fig. 5.3 which shows a global minimum and a local minimum surrounded by barren plateaus in one dimension. A random point on the landscape will likely encounter the barren plateau. The gradient for such plateaus is zero which is the same as it would be when a minimum is reached, thus the algorithm cannot progress as the neighbourhood of the point leads to no drop in the function value. Additionally, it has been shown that gradient free optimisers will not solve the problem either [Arr+21].

We show a numerical example of an occurrence of the phenomenon in the Heisenberg model in Fig. 5.4. The energy landscape is 702 dimensional and represents a  $3 \times 3 \times 3$  lattice of the isotropic Heisenberg model. We do not measure the gradients in each experiment but the energy found using those parameters. The energy values show the initial energies of  $10^4$  different random settings of the initial parameters. For comparison, note that the ground state energy is  $-73.452$  and the Néel state energy is  $-54$ . The fact that the energies shown in the figure are far above the Néel state energy shows that randomly initialising the parameters does not provide an optimal initial energy even when trying a large number of times. Although the quantum computer was prepared in the Néel initial state, choosing the parameters randomly destroys the initial state. The data shown in Fig 5.4 shows an almost flat line (see Fig. 5.3). The fact that random probes of the energy landscape show a flat line suggests the presence of barren plateaus.

Since finding a solution to the problem of barren plateaus is vital for the success of variational methods, it is essential that techniques are developed that help avoid barren plateaus.

### 5.3 Evolution

The dynamics of a quantum mechanical system is described by the operation of the unitary operators on the initial state of the system. Finding the ground state of a given problem, starting from some initial state, can be viewed from the perspective of finding a trajectory of the wavefunction that leads to it. In quantum annealing, this is known and requires the time evolution to be sufficiently slow [BF28; Kat50], which in turn requires depths of quantum circuits not feasible on current quantum computers. No variational calculations need to be involved in such dynamics. Variational algorithms are an alternative offering small depth

circuits where an optimisation algorithm traverses the trajectory. Among several drawbacks of optimisation is the possibility to get stuck in a local minimum. To ameliorate the drawbacks of both annealing and variational methods, we propose an idea that makes use of trajectories of both of them. We propose a kind of *dynamics* that has the same features, namely, the action of unitary operators on the initial state until the ground state is obtained. However, the proposal deviates from the previously mentioned dynamics insofar it introduces variational parameters and optimisation processes. Therefore, we name the process as quasi dynamics.

### 5.3.1 The quasi dynamics

Let the quantum mechanical system under study be a quantum computer. Assume that it is prepared in the initial state  $|\Psi_0\rangle$ . Let  $U(\theta)$  be a parametrised ansatz. Then, the wavefunction describing the state of a quantum computer as a function of  $M$  parametrised unitary operators is

$$|\psi(\theta)\rangle = U(\theta)|\Psi_0\rangle = U_M(\theta_M)\dots U_1(\theta_1)|\Psi_0\rangle. \quad (5.1)$$

Each unitary operator can have a different parameter. To avoid the mixing of notation, let  $\mathbb{U}(\Theta)$  be the unitary operators obtained when the optimisation algorithm signals convergence, where  $\Theta$  denotes the optimised parameters. The parameter optimised final state is given by

$$|\Psi_1\rangle = \mathbb{U}(\Theta)|\Psi_0\rangle. \quad (5.2)$$

Often the energy obtained by sampling  $|\Psi_1\rangle$  is significantly far away from the true ground state energy of the system. This can be due to local minima. One can perform another optimisation on  $|\Psi_1\rangle$  using once again the same ansatz. Thus, the state  $|\Psi_1\rangle$  can serve as the initial state,  $|\Psi_1\rangle \rightarrow |\Psi_0\rangle$ , for another variational optimization. Substituting Eq. (5.2) in Eq. (5.1) gives

$$|\psi(\theta)\rangle = U(\theta)\mathbb{U}(\Theta)|\Psi_0\rangle = U(\theta)|\Psi_1\rangle. \quad (5.3)$$

This process of substitution can be repeated  $p$  times such that the state of the system after  $p^{\text{th}}$  successive repetition is given by

$$|\Psi_p\rangle = \mathbb{U}_p(\Theta_p)\dots \mathbb{U}_1(\Theta_1)|\Psi_0\rangle. \quad (5.4)$$

We view each repetition as one cycle. We name the process represented by Eq. (5.4) as quasi dynamical state evolution, or *evolution* in short. To use  $|\Psi\rangle$  from cycle  $k$  as initial state in cycle  $k+1$ , special care needs to be taken that the parameters in  $U_{p+1}(\theta)$  are appropriately chosen. Otherwise, the progress can be insignificant. We demonstrate this point numerically using the following examples.

### 5.3.2 Demonstration

We consider four different lattice sizes of the one dimensional Heisenberg model to illustrate that there is little progress when appropriate parameters are not chosen for each cycle. To demonstrate the case, we choose new sets of random values for initialising each cycle's parameters. The results are shown in Fig. 5.5. Cycle number zero represents the final energy obtained using the standard variational quantum eigensolver. The initial parameters for cycle zero were zeros. Thereafter, the evolution heuristic is used with random parameters. We calculate the energy fidelity which is defined as the ratio of minimum energy obtained using the variational quantum eigensolver and the ground state energy. The results show that the energy fidelity decreases abruptly. The reason for the decrease can be explained from the fact that each cycle the optimiser begins anew and, therefore, may get stuck in a local minima.

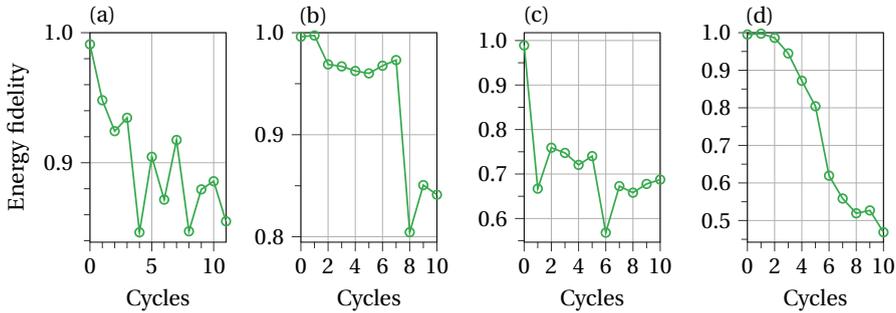


FIGURE 5.5: Energy fidelity for the isotropic one dimensional Heisenberg model as a function of the number of evolution cycles using random parameters for each new evolution cycle. (a) 9 qubits; (b) 10 qubits; (c) 11 qubits; and (d) 12 qubits.

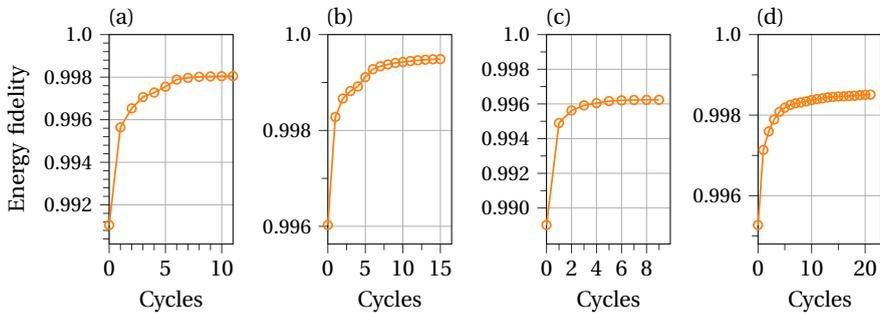


FIGURE 5.6: Same as Fig. 5.5 except for using zeros as parameters for each cycle. (a) 9 qubits; (b) 10 qubits; (c) 11 qubits; and (d) 12 qubits.

When choosing random parameters, each cycle is equivalent to performing a standard variational calculation where the initial states are different for each cycle. No systematic benefit can be guaranteed from such an approach.

In contrast to random parameters, we set all parameters to zero for every evolution cycle,  $\theta = [0, \dots, 0]$ . Starting from zeros is equivalent to initialising an identity circuit [Gra+19]. The results for the same problems are shown in Fig. 5.6. We observe that the energy fidelity only increases. The evolution heuristic is stopped after a certain number of cycles when the improvement in the fidelity is below a small threshold. Additionally, by avoiding an initialization at random places in the energy landscape possible barren plateaus can be avoided [McC+18].

### Operators for each cycle

While writing Eq. (5.4) it was assumed that the  $U(\theta)$  at each cycle is the same unitary operator. We used this assumption for the examples shown in Fig. 5.6. It is by no means necessary to adopt such an assumption. Let for each evolution cycle  $U(\theta)$  there be either a different set of unitary operators or a different combination of the same operators (if they do not commute). At present it is unclear how to choose a different  $U(\theta)$  at each cycle that allows further improvement in the energy, for a given Hamiltonian. The adaptive method to build an ansatz [Gri+19] solves this problem but is computationally expensive. Using the evolution

heuristic, different unitary operators can be used at each new cycle. This allows for expanding the parameter space at the expense of polynomial increase in the gate operations. Ideas similar to the proposed heuristic have been demonstrated for machine learning [Sko+21] and combinatorial optimization problems [Liu+22].

### 5.3.3 Connection to annealing

We conjecture that certain choices of  $\mathbb{U}(\Theta)$  at each cycle can facilitate finding the ground state  $|\Psi_g\rangle$  of  $H$ , such that in the asymptotic limit of  $p \rightarrow \infty$ ,

$$|\Psi_p\rangle \rightarrow |\Psi_g\rangle. \quad (5.5)$$

The convergence at the asymptotic limit is the worst case scenario. To find the ground state energy  $E_0$  using Eq. (5.5) and the variational principle, we get

$$E = \frac{\langle \Psi_p | H | \Psi_p \rangle}{\langle \Psi_p | \Psi_p \rangle} \rightarrow E_0 \quad \text{as } p \rightarrow \infty. \quad (5.6)$$

Repeated evolution cycles will only lower the energy. The underlying energy function decreases monotonically by construction of the evolution heuristic because an optimizer is designed to accept only those parameters that increase the energy fidelity (or lower the energy) at each iteration. Thus, the energy either improves or stays constant, as observed in the examples shown in Fig. 5.6. To increase the energy fidelity at each cycle it is assumed that suitable  $\mathbb{U}(\Theta)$  are chosen. The energy fidelity cannot be calculated except for exemplary problems, therefore, the decrease in energy should be the measure of improvement. The proposal remains a conjecture because it is unclear what the suitable  $\mathbb{U}(\Theta)$  should be for each cycle and given problem Hamiltonian.

Quantum annealing is based on the adiabatic theorem which guarantees that a system, (say) initially prepared in the ground state, will remain in its instantaneous ground state given that the following conditions are met. First, the change in the system Hamiltonian is slow enough and second, there is a sufficiently large gap between the ground and excited states. When  $p \gg 1$ , the parameters can be confined to some bounds in such a way that the state evolution corresponds to an adiabatic evolution. The quantum approximate optimization algorithm [FGG14] and rapidly quenched quantum annealing [Cal+21] have been developed with similar lines of thought.

## 5.4 Results

We employ the evolution heuristic in two different ways. First, we use it to test if there is an improvement in the ground state energies obtained by the variational quantum eigensolver for the Heisenberg model. Second, we generate random Hamiltonians and verify if the heuristic can improve the energy fidelities.

### 5.4.1 Heisenberg model

We employ the heuristic on antiferromagnetic rings of size  $4 \leq N \leq 12$ . First we test the standard strategy of initialising the parameters randomly a hundred times against using the Néel initial state. Due to the small size of the problems, exact diagonalisation can be used to find the ground state energy. The results are shown in Fig. 5.7(a). We found that rings up to

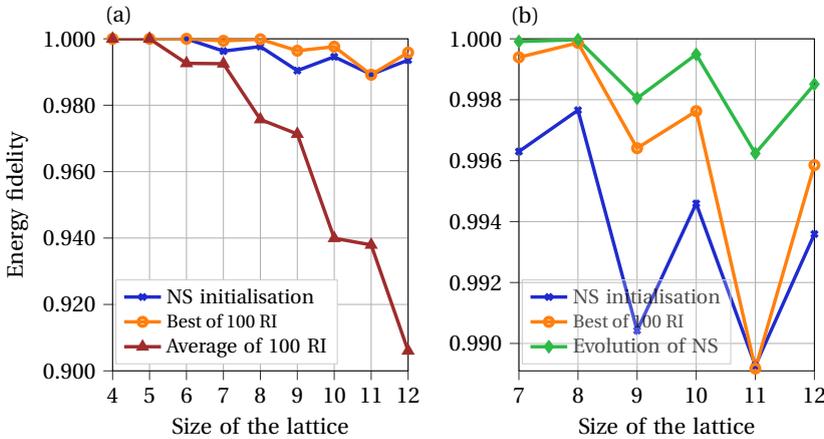


FIGURE 5.7: Energy fidelity as a function of the size of the lattice for the isotropic one dimensional Heisenberg model when using different strategies to find the ground state energy. (a) Comparison between the Néel state initialisation and random initializations (RI) strategies for the antiferromagnetic rings. (b) Same as (a) except that the average of RI runs data is replaced by the evolution data, and the x- and y-axis scales are changed.

$N \leq 5$  are relatively easy problems and both the strategies can find the energy with a fidelity of 1.000.

The average performance of the random initialisation strategy drops significantly as the size of the lattice increases. This behaviour is explained on the basis of the existence of local minima in the energy landscape and was to be expected. The best energy fidelity out of all the random initialisations performs better, however, at the expense of using approximately 100 times more computational time. It is interesting to note that starting the optimisation from the Néel initial state is beneficial for all lattice sizes tested. This is not trivial since we know of no guarantee that the point in the energy landscape for the Néel initial state is itself not a local minimum.

Given the success of the Néel state initialisation, we employ the evolution heuristic to use the final state obtained after the optimiser converges the variational optimisation starting from the Néel state. The same  $U(\theta)$  are used for each cycle and the evolution is stopped when the decrease in the energy is less than  $10^{-4}$  between consecutive cycles. The results are shown in Fig. 5.7(b). We observe that the energy fidelity using evolution exceeds the best of all random initialisations for all lattice sizes. The reason for odd numbered lattices having relatively lower energy fidelities can be thought to be arising from the degeneracy of the ground state. We conclude that the evolution heuristic is able to improve the approximation to the ground state energy systematically.

## 5.4.2 Random Hamiltonians

We study the performance of the evolution heuristic by trying it out on randomly generated Hamiltonians. We first define the total Hamiltonian

$$H_T = \sum_{\langle i,j \rangle} \left( J_{ij}^{xx} \sigma_i^x \cdot \sigma_j^x + J_{ij}^{yy} \sigma_i^y \cdot \sigma_j^y + J_{ij}^{zz} \sigma_i^z \cdot \sigma_j^z \right), \quad (5.7)$$

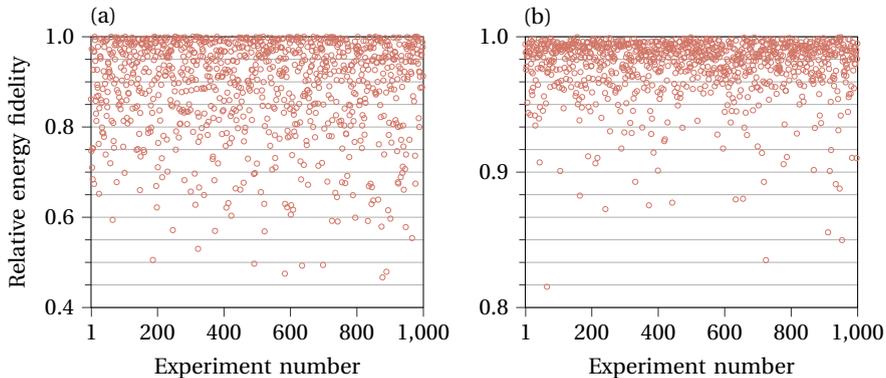


FIGURE 5.8: Relative energy fidelity for  $10^3$  different Hamiltonians generated by random assignment of coefficients and subsets of terms from Eq. (5.7). Lower relative energy fidelity signifies a better improvement of the energy by the evolution. A value of 1.0 signifies no improvement. Data taken for (a)  $N = 6$  and (b)  $N = 8$ .

where the sum  $\langle i, j \rangle$  sums over all pairs of  $N$  lattice sites. A total of one third of all the terms are randomly picked from  $H_T$  and their coefficients are assigned random values in the interval  $J_{ij}^{\alpha\alpha} \in (0, 10)$  for  $\alpha \in \{x, y, z\}$ . Note that Eq. (5.7) is analytically solvable for  $J_{ij}^{xx} = J_{ij}^{yy} = J_{ij}^{zz} = 1$ . The ground state energy for a finite subset of terms in the Hamiltonian defined in Eq. (5.7) with random coefficients needs to be computed numerically. We confine our numerical study to rings of length six and eight. For the former case, we use only half of all the terms in the XY-ansatz while in the latter case we use the complete XY-ansatz. The optimisation parameters are initialised randomly and the evolution is performed on the final state nine times. To study the decrease in the energies, we define a relative energy fidelity which is the ratio of the energy obtained through the standard variational quantum eigensolver and the final energy obtained after nine evolution cycles. The purpose here is not to find the ground state energy as accurately as possible but to find out if evolution can decrease the energy when the standard eigensolver gives suboptimal results.

The results for a thousand different experiments are shown in Fig. 5.8. Lower relative energy fidelity means a better improvement offered by evolution. Figure 5.8(a) shows that the improvement is substantial for a majority of experiments when  $N = 6$ . One reason is that only half of the terms in the XY-ansatz were used. It was known that this modified ansatz does not necessarily find the ground state energy accurately, therefore, it is an ideal case to test the evolution heuristic. In contrast, in Fig. 5.8(b), the complete XY-ansatz was used which we know is able to approximate the ground state energy reasonably well. Therefore, we observe relatively less improvement for the  $N = 8$  case than for the  $N = 6$  case. In all the experiments that show improvement the standard variational quantum eigensolver gets stuck in some local minimum, and using evolution one is able to further improve the energy. Furthermore, even the simplistic approach of using the same  $U(\theta)$  for each cycle appears useful.

### 5.4.3 Large lattices

Results from large lattices have been published in [Jat+22b]. Here we briefly summarise the findings. Under study were the antiferromagnetic Heisenberg rings of size  $13 \leq N \leq 25$ , ladders of size  $6 \times 2 \leq N \leq 13 \times 2$ , and square lattices of size  $4 \times 4 \leq N \leq 6 \times 6$ . Either open or

periodic boundary conditions were considered. The evolution was performed on the final state obtained after the standard variational eigensolver application to the Néel initial state. Due to the time required to complete the evolution for such large lattice the number of cycles were restricted to three for the rings and to one for others. The ground states were calculated numerically using the Lanczos algorithm for calculating the energy fidelity.

All antiferromagnetic rings started with an energy fidelity of less than 0.60 when initialised in the Néel initial state. After the application of the evolution heuristic the fidelities obtained were greater than 0.99. Similarly, for the ladder and square lattices the fidelity was less than 0.75 when initialised in the Néel state. After one evolution cycle the fidelity for all ladder lattices improved at least up to 0.98. There was no noticeable improvement for the square lattices except for the  $6 \times 6$  case. This can be explained since the same  $U(\theta)$  may not be the best choice for each cycle.

## 5.5 Discussion

In this section, we discuss the following two issues. First, once the computer has reached the ground state of the problem, samples from measuring that state need to be stored and manipulated in a classical computer. We highlight some issues relevant for manipulating bitstrings obtained from the measurement. Second, quantum computers are often compared against classical computers. We compare the emulation of a quantum computer on a classical computer against a hypothetical fully functional quantum computer.

### Parity computation considerations

We performed the evolution calculations by manipulating the complete wavefunction. On an actual quantum computer, the wavefunction will not be accessible. Instead, once the computer is prepared in the ground state, the final state will be sampled uncovering the underlying distribution. Evolution, or quantum algorithms in general, can be deemed to be useful if the following assumption about the sampling is satisfied. Finite samples should be sufficient to accurately calculate the expectation value of the Hamiltonian. Additionally, if the number of contributing states in a certain measurement basis do not increase exponentially with  $N$ , their storage is feasible in the matrix form expressed in Eq. (3.37). However, if this is not possible, the expectation value can still be calculated by processing each bitstring after it is measured without storing it. Thus, there are two ways of processing the measurements made on the quantum computer; (I) storage of all the bitstrings before processing or (II) processing the bitstrings as they are measured. In this section, we examine which approach is advantageous.

To visualise some of the results that use the evolution heuristic, we sample the final state  $10^6$  times. The number of samples is motivated by the same number used to demonstrate quantum supremacy in a different experiment [Aru+19]. We plot the unique states obtained from the samples against  $2^N$  for each of the ring and ladder lattice results. The data is plotted in Fig. 5.9. Note that  $10^6$  samples are not necessarily sufficient to accurately extract all the possible unique states when  $2^N$  is more than or comparable to  $10^6$ . For the ring and ladder cases requiring less than 16 qubits, where the total samples are orders of magnitude more than all possible states, we observe that unique states do not overlap with all possible states. This suggests that not all states contribute to the calculation of the expectation value. For  $N > 20$ , the number of samples were less than all possible states, so that no concrete conclusions can be drawn. In general, as the ring and ladder sizes increase, we observe that the number of unique states start to approach the total number of samples. We can expect that as the

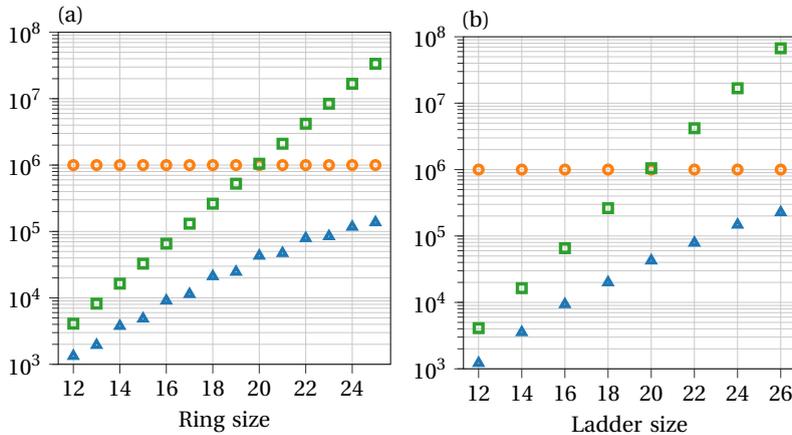


FIGURE 5.9: Comparison between the unique number of states (triangles) and all possible states (squares) found when sampling a million times (circles). Results are shown for (a) isotropic rings and (b) isotropic ladders.

size is increased beyond 26 qubits, more and more unique states will be sampled. It can be expected that for  $N > 50$ , sampling a million times will likely yield only unique samples for the Heisenberg model. This was also the case for the supremacy experiment [Aru+19]. To explain the relevance of this result, we first discuss the following.

A part of the classical computational effort required to evaluate the expectation values lies at calculating the parity (see section 3.3.4) of each measured bitstring corresponding to the terms in the Hamiltonian. The effort to compute the parity is proportional to

$$N \times s \times m, \quad (5.8)$$

where  $N$  is the number of qubits,  $s$  are the total number of samples, and  $m$  is the number of terms in the Hamiltonian. Although  $s$  and  $m$  in Eq. (5.8) are expected to scale polynomially with  $N$ , thus making evaluation of expectation values possible, it is relevant to ask if it will be practically significant. A quantum computer with 60 qubits has recently been sampled  $7 \times 10^7$  times [Zhu+22]. We use these values for the Heisenberg model, which would have 180 terms in the Hamiltonian. The maximum number of comparison operations required to compute the parity of all the bitstrings would be  $60 \times 7 \times 10^7 \times 180 = 7.56 \times 10^{11}$ . This is not an insignificant number of operations but not very large either.

The number of terms in the Heisenberg model scale linearly as  $3N$ . In some quantum chemistry applications, the number of terms scales polynomially rather than linearly. We tabulate information about some molecular Hamiltonians in Table 5.1. Note that the number of terms  $m$  depends on the basis used for the Hamiltonian. The tabulated data used minimal bases *sto6g* or *631g*, whose accuracies are low but serve as good prototypes. The parity operations required for such prototype problems are significant. With improved and more accurate bases, one can expect the number of operations to increase further. As an example, consider that the water molecule in the *sto6g* basis has  $N = 8$  and  $m = 514$ , but in another basis (*631gss*) has  $N = 42$  and  $m = 107382$ . Assuming that a million samples are enough to approximate the ground state energy, the former (latter) would require  $4.11 \times 10^9$  ( $4.51 \times 10^{12}$ ) operations. The molecules considered here are considered prototype problems; as more challenging molecules are considered, the operations can be expected to increase further. While

Molecule	$N$	$m$	$N \times s \times m$
OHOHO	24	9187	$2.20 \times 10^{11}$
NCHOH-	24	24263	$5.82 \times 10^{11}$
$N_2$	28	21489	$6.02 \times 10^{11}$
CN-	29	42897	$1.24 \times 10^{12}$
OHO-	32	34475	$1.10 \times 10^{12}$

TABLE 5.1: Number of operations required to calculate the parities of a million samples ( $s = 10^6$ ) for different molecules.  $N$  is the number of qubits and  $m$  is the number of terms in the Hamiltonian.

they may grow only polynomially, it is still relevant to manage them such that the total time of the entire computation of an algorithm, both classical and quantum, remains practical.

In this section, we predicted that future larger scale simulations of the Heisenberg model will likely sample only unique states. This assumption may also hold for various other interesting Hamiltonians. Additionally, the classical computational effort required to compute the parity of the sampled bitstrings is not insignificant even for prototype problems and will likely increase for future large scale simulations. Combining these two observations, we conclude that it would be beneficial to process the parities of the bitstrings as soon as they are measured instead of storing them and doing it later.

### Timing against an emulator

One core expectation from future quantum computers is the ability to solve some problems significantly quicker than classical computers. Quantum advantage is often shown by using a quantum computer to compute a certain task and estimating the time required by a classical computer to complete the same task. In this section, we perform the task differently. We compare the time required by an emulator to mimic a quantum computer against the potential time required by a fully functional quantum computer. We estimate the latter using the currently available single and two qubit gate operation times. The total number of operations on a quantum computer can be estimated as

$$N_o = N_G \times a T_H \times s, \quad (5.9)$$

where  $N_G$  is the number of gates,  $T_H$  is the number of terms in the Hamiltonian,  $0 < a \leq 1$  accounts for the fact that several terms in the Hamiltonian may be measured simultaneously for some problems, and  $s$  is the number of samples required to reach a certain accuracy.

We make the reasonable assumption that the same gate operation takes the same amount of time on each execution on the actual quantum computer. Ion trap based computers offer relatively slower two qubits gates where the execution time is of the order  $1.6\mu s$  [Sch+18]. The estimates for superconducting type quantum computers vary. Some require single qubit gate execution times of the order of 100 ns and two qubits gates around 500 ns [Tak+17], while others require execution times for both type of gates close to the order of 10 ns [Aru+19]. For the calculation below we take the latter most optimistic values.

The total number of energy evaluations required for the  $N = 24$  spins ring up to three evolution cycles was 84744. A new layer of  $U(\theta)$  was added to the circuit at each cycle. For the purpose of estimating the execution time, relevant is the total number of operations mentioned in Eq. (5.9) that need to be performed sequentially. This number is often lower than

all possible single and two qubits operations. The number of sequential operations were approximately 6400 and 11500, for single and two qubits gates, respectively. Taking optimistic values  $aT_h = 3$  and  $s = 2^{13}$ , the time required to calculate the energy once would be about 44 seconds. The total time required to complete the variational calculation using a quantum computer stands at 43 days. In comparison, running the emulator on a supercomputer required 4 days for the same computation.

For the largest lattice case of a  $20 \times 2$  ladder ( $N = 40$ ), the time taken to calculate the energy using JUQCS once was on average 110 seconds. The number of terms in the XY ansatz were sharply reduced to allow ten iterations per run. The reduced ansatz did not recover the ground state energy. We only discuss the computational time required without considering evolution. The time required by a fully working quantum computer, in comparison, would be about 3.23 seconds. The time for JUQCS is larger partly because of the transfer of the wavefunction between different nodes of the supercomputer is time consuming and partly because the computational effort per node also increases. Thus, as the system size increases, a fully functional computer may perform tasks faster than its emulator on a supercomputer. We note two important things. First, the comparison does not illustrate quantum advantage since we compared an emulator against the quantum computer. There may exist classical algorithms that find the ground state energy quicker than emulated variational algorithms. Any conclusion of potential advantage needs to study the best classical algorithm against variational algorithms on the computer. Second, the ansatz used will expectedly not find the ground state energy, and was only used to show that a large scale simulation is possible by manipulating the state vector. Using the XY ansatz for  $N = 40$  is not yet practical on supercomputers.

## 5.6 Conclusion

We proposed the idea of quasi-dynamical evolution which was built on standard variational methods and systematically lowers the ground state energy of a problem Hamiltonian. The idea, by construction, permits only improvements in the energy. The idea is generally applicable to a broad range of variational methods and is inspired by quantum annealing. We demonstrated its working by improving the standard variational estimates of ground state energies for the Heisenberg model. While evolution is a useful heuristic, it does not guarantee finding the ground state energy for problems in general. The reason for this is the fact that it is not clear what  $U$  are the most suitable for each cycle. Future works should focus on answering this open question.



## Chapter 6

# Error mitigation and benchmarking

Quantum computers built to date are error prone. It is known that error correction can help overcome the challenges of working with an error prone quantum computer [DS96; Sho96; Got98; DiV09]. Error correction is described as a method of protecting quantum computation against errors by encoding redundant information to the computation [NC10]. Error correction can be termed *active* in the sense that the technique is executed simultaneously with the main application that it helps avoid errors. The redundancy requires several additional qubits per logical qubit [DiV+00] to be used for error correction to be implementable. This is a significant disadvantage since the fabrication of large scale quantum computers remain a difficult challenge. We wish to avoid requiring additional hardware and need an alternative method altogether. This chapter is based on the work published in [Jat+20].

### 6.1 Introduction

Error mitigation is one way to overcome the errors in computations on current and near-term quantum computers without requiring additional qubits. The idea is that if a quantum computer can systematically reproduce its errors, then the errors may be mitigatable to some extent. In contrast to error correction, error mitigation protocols are *passive* since they need to be run either before or after the execution of the main application. Such protocols often do not require additional qubits but additional resources from the available qubits, thereby circumventing the main problem faced by error correction.

Several error mitigation techniques have been studied. Examples include mitigation using error extrapolation [TBG17; LB17] (see [Zha+20a] for experimental demonstration), using quasi-probability decomposition [TBG17], using conserved symmetries [BM+18] (see [Sag+19] for experimental demonstration), using a stabilizer-like method [MYB19] for depolarisation errors, using correlated Markovian noise models [Bra+21], using gate set tomography and quasi-probability decomposition [Son+19], using subspace expansion [McC+17a], and using matrix based methods [Jat+20; Nat+21]. Error mitigation has been demonstrated for up to 20 qubits [RGM22]. Despite the progress in error mitigation protocols, drawbacks persist. For example, one protocol works only for small depth circuits [TBG17] or another that mitigates only measurement based errors [Bra+21]. Often error mitigation protocols attenuate the errors but do not completely eliminate them. Therefore, it becomes necessary to weigh the cost of the extra resources required against the benefit obtained. Often the protocols are difficult to implement and may not be attractive to an end user of a quantum computer who lacks knowledge of complicated protocols.

This chapter introduces and demonstrates easy to implement error mitigation protocols on real quantum computers. We introduce criteria that ideal mitigation protocols should strive for, and explore the additional benefit of benchmarking offered by matrix based mitigation protocols. We also implement a technique to achieve scalable error mitigation. Our proposal

has been used in at least two independent studies [Gho+21; TGH22]. We also explore ways to benchmark quantum computers using the techniques used in error mitigation as well as develop other models.

## 6.2 Mitigation criteria

Although several mitigation protocols exist and have also been experimentally demonstrated, a concrete formulation of the requirements that such protocols should aim for has not been formulated. Such requirements can serve as a guideline for developing new and better protocols. Below we discuss the criteria published in [Jat+20] that all *ideal* error mitigation protocols should fulfil.

1. **Result recovery.** The protocol should be able to mitigate errors to a satisfactory accuracy. For protocols where theoretical guarantees are missing, we quantify this in section 6.3.
2. **Depth independence.** The protocol should not depend on circuit depth. An ideal error mitigation protocol should work independently of the depth of the circuit used in the main application.
3. **Error model.** The protocol should take into account all types of errors a quantum computer may be prone to. Additionally, it should not rely on prior information about errors the quantum computer is prone to.
4. **Practically realisable.** The protocol should make use of resources that are practically similar to the resources used by the circuit which is to be mitigated. The total time taken by the mitigation protocol should be reasonable compared to the main application.
5. **No additional hardware.** The protocol should not require (many) additional qubits for being implementable. For example, some protocols may require a single ancilla qubit. If many additional qubits are required, error correction might be a better alternative.
6. **Gate-set independence.** The protocol should take into account and be applicable to all types of quantum gates. It should not be restricted to using a certain set of gates only.
7. **No output knowledge.** The protocol should not make use of any specific knowledge about the output of a given circuit.

We introduce a standard matrix based protocol which is commonly used for mitigation of state preparation and measurement errors. This standard protocol fulfils requirements 2, 5, and 7. We then improve this standard protocol using a simple and easy to implement mechanism. The new, improved protocol satisfies the requirements 2, 3, 5, 6, and 7. While the standard protocol only mitigates state preparation and measurement errors, the improved protocol mitigates all general errors during a quantum circuit execution.

## 6.3 Standard protocol

Two persistent errors in any error prone quantum computer can appear during state preparation and measurement. The standard protocol is helpful if a large part of an erroneous quantum computer output contains these errors. The standard protocol works as follows. Assume that state preparation and measurement errors for a given circuit  $C_g$  will also occur for

other circuit(s)  $C_c$ . To mitigate the errors, measure the effects produced by these errors by designing  $C_c$  such that their outputs are known or easy to compute. Then,  $C_c$  can be used to *calibrate* the output of the quantum computer. We call, therefore,  $C_c$  calibration circuit(s).

Assume that an error prone quantum computer produces some bitstrings with relative frequencies  $v_1, v_2, \dots, v_{2^N}$  that differ from the ideal (exact) expected probabilities  $e_1, e_2, \dots, e_{2^N}$ . Further assume that there exist error mitigated frequencies  $x_1, x_2, \dots, x_{2^N}$  and ideal emulated frequencies  $s_1, s_2, \dots, s_{2^N}$  as well. The column vectors representing these quantities are defined as

$$\mathbf{V} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{2^N} \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{2^N} \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{2^N} \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{2^N} \end{pmatrix}. \quad (6.1)$$

Assume that the vectors in Eq. (6.1) are normalised, i.e.  $\sum v_i = \sum e_i = \sum x_i = \sum s_i = 1$ , where  $i = 1, \dots, 2^N$ . The working of the standard protocol assumes the existence of a  $2^N \times 2^N$  matrix  $M$  such that

$$M\mathbf{E} = \mathbf{V}. \quad (6.2)$$

The matrix  $M$  is sometimes named the complete assignment matrix [Nat+21]. Other names are used in gate set tomography [Gre15]. Although not used in its current form, Eq. (6.2) serves as a good starting point to understand the basic idea of the protocol. Note that  $M$  is the identity matrix when the quantum computer is not error prone. For an error prone computer,  $M$  has non-zero off-diagonal entries. For a computer in which errors dominate significantly, all the matrix entries can be expected to be similar in magnitude, suggesting that the computer was working as a random number generator. These distinctions can serve as a basis for the benchmarking of quantum computers. The standard protocol uses the calibration circuits  $C_c$  to fill the matrix  $M$  and use it for error mitigation.

### Calibration matrix

Determination of the calibration matrix requires  $2^N$  circuits given an application using  $N$  qubits. The procedure is as follows. Prepare different circuits where the qubits are initialised in all possible  $2^N$  (independent) initial states. Each such circuit is a calibration circuit. Measure each circuit. Enter the obtained frequencies from each calibration circuit into columns of  $M$ , where the  $j^{\text{th}}$  column, starting from left, takes output frequencies from the circuit whose state is given by the binary representation of  $j$ , for all  $j = 1, \dots, 2^N$ .

The matrix  $M$  can now be used to mitigate (errors in) the relative frequencies  $\mathbf{V}$  obtained from the primary circuit  $C_g$ . We rewrite Eq. (6.2) in terms of the mitigated and emulated frequencies, such that  $M\mathbf{X} = \mathbf{V}$ , where  $\mathbf{X}$  representing the mitigated frequencies may not always be exact (i.e. equal to  $\mathbf{E}$ ). There is a problem if we proceed to solve for  $\mathbf{X}$  using  $\mathbf{X} = M^{-1}\mathbf{V}$ . Since we are dealing with normalised relative frequencies constrained to be in the interval  $[0, 1]$ , an inverse of  $M$  to solve for  $\mathbf{X}$  can lead to values in  $\mathbf{X}$  outside this interval [MZO20; Gel20]. This is a problem because neither probabilities nor frequencies can be negative. The problem can be circumvented by minimising least squares, also termed as constrained optimal likelihood estimation [Gre15]. The task is changed to finding the minimum of the function

$$f = \sum_{i=1}^{2^N} (v_i - (M \cdot X)_i)^2, \quad (6.3)$$

given the constraints  $0 \leq x_i \leq 1$  and  $\sum x_i = 1$  summed over all  $i = 1, \dots, 2^N$ . The vector  $\mathbf{X}$  is initialised at random and we use the *Sequential Least Squares Quadratic Programming* (SLSQP)

[Kra88] algorithm. Note that while Eq. (6.2) cannot be used in some cases, using Eq. (6.3) is always possible.

### Quantifying improvement

The error mitigated frequencies  $\mathbf{X}$  can be compared to the frequencies  $\mathbf{S}$  obtained from an ideal emulator (see Eq. (6.1)). To test the protocol we define a quantitative test of improvement as follows. Let the root mean square errors between the mitigated and ideal frequencies be given by

$$\Delta X = \sqrt{\sum_{k=1}^{2^N} (x_k - s_k)^2}, \quad (6.4)$$

and between the measured frequencies and the ideal frequencies,

$$\Delta V = \sqrt{\sum_{k=1}^{2^N} (v_k - s_k)^2}. \quad (6.5)$$

By defining  $\Delta_Q = \Delta V - \Delta X$ , we have the possibilities

$$\Delta_Q \begin{cases} > 0, & \text{for positive mitigation.} \\ < 0, & \text{for negative mitigation.} \\ = 0, & \text{for no mitigation.} \end{cases} \quad (6.6)$$

Note that  $\Delta_Q$  only represents the overall error in all the states and does not suggest the individual differences. It is possible to envision that all but one state contributes to a positive or negative value of  $\Delta_Q$ . Despite this drawback,  $\Delta_Q$  is a valuable metric represented by a single scalar value. An ideal error mitigation protocol should obtain large and positive  $\Delta_Q$ . Additionally,  $\Delta_Q$  equal to zero or close to zero is not acceptable since a large number of resources may have been spent in creating  $M$ . In the worst case, it is possible that  $\Delta_Q < 0$  which indicates that the error mitigation protocol worsened the outcome.

### Drawbacks

Most algorithms of practical interest do not use circuits considered by the standard protocol. In general, several quantum gate operations form the basis of the underlying benefits that can be reaped from quantum computers. If several hundred gates are involved, state preparation and measurement errors are likely negligible compared to the overall errors. Since the standard protocol only fulfils requirements 2, 5, and 7, it is of interest to develop an improved protocol that ameliorates the shortcomings of the standard protocol.

## 6.4 Improved protocol

Quantum circuits often contain many gates. The errors in computation will accumulate if the gate operations are prone to errors. Therefore, in general, we wish to mitigate the effects of errors that arise not only due to state preparation and measurement but potentially due to other sources, i.e. erroneous gate operations. Often we do not know *a priori* what error sources will contribute to or dominate in any practical computation. One should include all potential error sources. To this end, a general protocol to mitigate errors in a quantum

circuit has been proposed [Jat+20]. This general protocol has been independently used in mitigating errors in a BB84 quantum key distribution scheme [Gho+21] and in simulating the time evolution of open systems [TGH22].

The general protocol improves the standard protocol using a simple idea. To mitigate errors in a given circuit  $C_g$  with  $N$  qubits of depth  $D$ :

1. Similar to the standard protocol, prepare the calibration circuits in all possible  $2^N$  states, but twice.
2. Consider all the gates of the circuit  $C_g$  up to depth  $D/2$  (if  $D$  is even) or  $(D-1)/2$  (if  $D$  is odd), and add them to the calibration circuits. The reasons for this choice are practical considerations regarding current quantum computers and are discussed in the next section.
3. Add *inverse gates* of the gates added in step 2, in reverse order. Measure the calibration circuits and record the observed frequencies in the calibration matrix  $M_1$ . The entry of the frequencies in the matrix is the same as the standard protocol.
4. Repeat steps 2 and 3 for the remaining half of the gates in the circuit  $C_g$  for the remaining calibration circuits, and name the new matrix  $M_2$ .
5. Calculate the average of the above matrices,  $M = (M_1 + M_2)/2$ .

After  $M$  is obtained, the remaining steps are the same as for the standard protocol. The difference in the improved and the standard protocol originates from considering the gates in the computation as crucial sources of error.

### Calibration circuits

This section discusses the reasons for the choice of step 2 in the improved protocol. To reproduce errors occurring in  $C_g$ , one should ideally use  $C_g$  also as the calibration circuit(s). The circuit depths of both circuits would be approximately the same. We say *approximately* because one often needs an extra gate for the state preparation in  $C_c$ . This approach is not possible because a priori we do not know the outputs of the circuit  $C_g$ . Knowing these outputs would violate requirement 7. In this case, the matrix  $M$  will not necessarily have a diagonal structure, even for an ideal quantum computer. As an alternative, we propose to use a circuit similar to  $C_g$  for which the outputs are always known, which is the identity circuit, e.g.  $C_{2c} \approx C_g^\dagger C_g$  (see [Jat+20] for a pictorial view). The new circuit then has double the depth of  $C_g$ . By using this approach, we approximate the errors produced in  $C_g$  of depth  $D$  with errors produced in  $C_{2c}$  of depth  $2D$ . While working with error prone quantum computers of IBM Q, we found that circuits with different depths exhibit different errors. Therefore, doubling the depth is likely to reduce the protocol's efficacy and include errors unaccounted for by the matrix  $M$ . For this reason, we divide  $C_g$  in half and then take  $C_c \approx C_{g/2}^\dagger C_{g/2}$  to keep the circuit depth of the calibration circuit(s)  $C_c$  and the given circuit  $C_g$  very close. A drawback of the improved protocol is that it requires twice the amount of circuit executions required by the standard protocol. The benefit is that it captures the gate operation errors.

## 6.5 Experimental demonstration

Several random circuits are generated and their outputs mitigated to test the effectiveness of the improved protocol. Some of these were also compared with the standard protocol. To

Applied Gate	Basis Gate
<b>ID</b>	<b>ID</b>
<b>U1(<math>\theta</math>)</b>	<b>U1(<math>\theta</math>)</b>
<b>X</b>	<b>U3(<math>\pi, 0, \pi</math>)</b>
<b>Y</b>	<b>U3(<math>\pi, \pi/2, \pi/2</math>)</b>
<b>Z</b>	<b>U1(<math>\pi</math>)</b>
<b>H</b>	<b>U2(0, <math>\pi</math>)</b>
<b>S</b>	<b>U1(<math>\pi/2</math>)</b>
<b>S<sup>†</sup></b>	<b>U1(<math>-\pi/2</math>)</b>
<b>T</b>	<b>U1(<math>\pi/4</math>)</b>
<b>T<sup>†</sup></b>	<b>U1(<math>-\pi/4</math>)</b>
<b>CNOT(c→t)</b>	<b>CNOT(c→t)</b>

TABLE 6.1: The applied gates in the protocols are the gates that were added to the generated circuits. The basis gates are their corresponding gates actually implemented on the quantum computer.

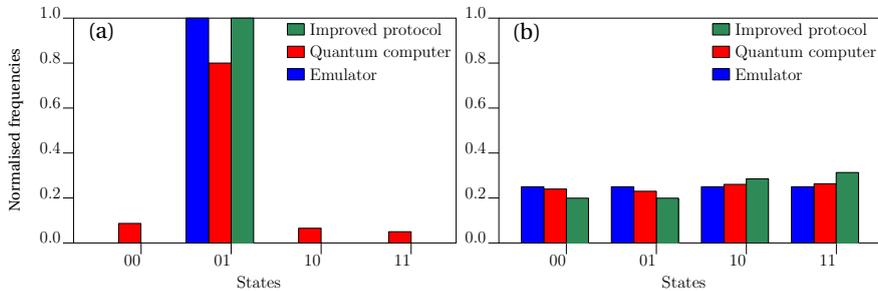


FIGURE 6.1: (a) Normalised frequencies of all the states obtained from an emulator, quantum computer, and after error mitigation using the improved protocol. (b) Same as (a) except for a different circuit.

get a quantitative estimate of the ability of the improved protocol, it is applied to randomly generated circuits for a large number of cases. Although the improved protocol is independent of the set of gates used, we use only a subset of all possible gates for practical reasons. The gates we used form a universal gate set. These are:  $\{\text{Id}, \text{U1}, \text{X}, \text{Y}, \text{Z}, \text{H}, \text{S}, \text{S}^\dagger, \text{T}, \text{T}^\dagger, \text{CNOT}\}$ . These gates are transpiled into quantum computer compatible gates, as supported by different architectures shown in Table 6.1. For our experiments, we used the following IBM Q quantum computers: Burlington [5-qb], Ourense [5-qb], and Armonk [1-q].

The relevant details of the conducted experiments are as follows. Random circuits were created respecting the physical connectivity of the qubits on the device. Each circuit and its mitigation were repeated ten times to observe statistical deviations. The range of circuit depths for each  $N$  was chosen keeping in mind the actual device performance.  $\Delta V$  was used as an approximate indicator of device performance. Total samples for all experiments were kept at  $2^{13}$  per experiment, the maximum supported by all the devices. The emulator results were obtained using  $2^{13} \times 100$  samples. The measurement gate is not included in the circuit depth.

### 6.5.1 Improved protocol

The error mitigation results for two different 2 qubits circuits are shown in Fig. 6.1. These two circuits have been hand picked to show the two possible extremes of error mitigation using the improved protocol. Figure 6.1(a) shows positive mitigation, where the blue bar represents the frequencies obtained using an emulator [Ale+], the red bar shows the output obtained from a quantum computer, and the green bar the error mitigated results using the improved protocol. In this case, the mitigation is positive since the result from the protocol is closer to the true result than the quantum computer. On the contrary, Fig. 6.1(b) shows negative mitigation where the protocol makes the output from the quantum computer worse. A pattern of positive and negative mitigations was observed in a large number of randomised experiments, which shows the trend that the mitigation is always positive if the (ideal) output states are directly represented in the matrix  $M$  (i.e. 01 in Fig. 6.1(a)). The mitigation fails if the ideal output is an equal superposition of all possible computational states.

In Fig. 6.2, we plot the average trend for 100 different experiments each repeated 10 times for different  $N$ . We present the data for the experiments in ascending order of the average  $\Delta V$  for better readability. Recall from Eq. (6.6) that we are interested in  $\Delta_Q > 0$  since this implies positive mitigation. Some of these results can be directly compared to those shown in [Jat+20] where only the  $\Delta X$  and  $\Delta V$  are shown. In Fig. 6.2(a), a large number of experiments numbered up to 40 have  $\Delta_Q < 0$ . By looking at these cases individually, we notice that these experiments belong to the category where the ideal states are similar to the case shown in Fig. 6.1(b). For the plots (a), (c), and (e) in Fig. 6.2, several  $\Delta_Q$  exist which are negative or close to zero. These cases mean the protocol was unable to mitigate the errors. However, most experiments show  $\Delta_Q > 0$  and hence positive error mitigation. For the plots (b), (d), and (f) in Fig. 6.2, the gate set for the generation of the random circuits did not include the **H** gate and, therefore, only those states are observed that exist in  $M$ . Since we know the improved protocol works best in this case, we observe large positive mitigation. However, such cases are unlikely to occur in practical quantum algorithms.

#### Calibration matrix

The matrix  $M$  plays a central role in standard and improved mitigation protocols. Both protocols differ in how the circuits are designed to fill the matrix. This matrix can also offer other insights before it is used for mitigation. The standard and improved protocols can work for arbitrary depth circuits if the quantum computer produces distinguishable column entries in the calibration matrix. By observing if the entries within each column of the matrix are all nearly equal, we can infer that the quantum computer worked completely randomly if the output was (unintentionally) uniform over all states. In such a case, we do not expect positive mitigation. Alternatively, positive error mitigation may be expected if the matrix's columns contain distinguishable (e.g. dominant diagonal) entries.

### 6.5.2 Standard protocol

We also present results for comparison of the standard and improved protocols. We apply the standard protocol to 100 different experiments each repeated 10 times for  $N = 2$  and  $N = 3$ . The results are shown in Fig. 6.3(a-b). We plot the root mean square deviation  $\Delta_Q$  for the averages of  $\Delta V$  and  $\Delta X$  over the 10 repetitions. The data is shown in the increasing values of  $\Delta V$ . The shown results correspond to  $N = 2$  with depths in the range [72,80], and  $N = 3$  with depths in the range [7,11]. By comparing the standard protocol data from Fig. 6.3(a) with the improved protocol data from Fig. 6.2(b), we observe that the  $\Delta_Q$  are much more positive for the improved protocol. This result signifies that the improved protocol can take

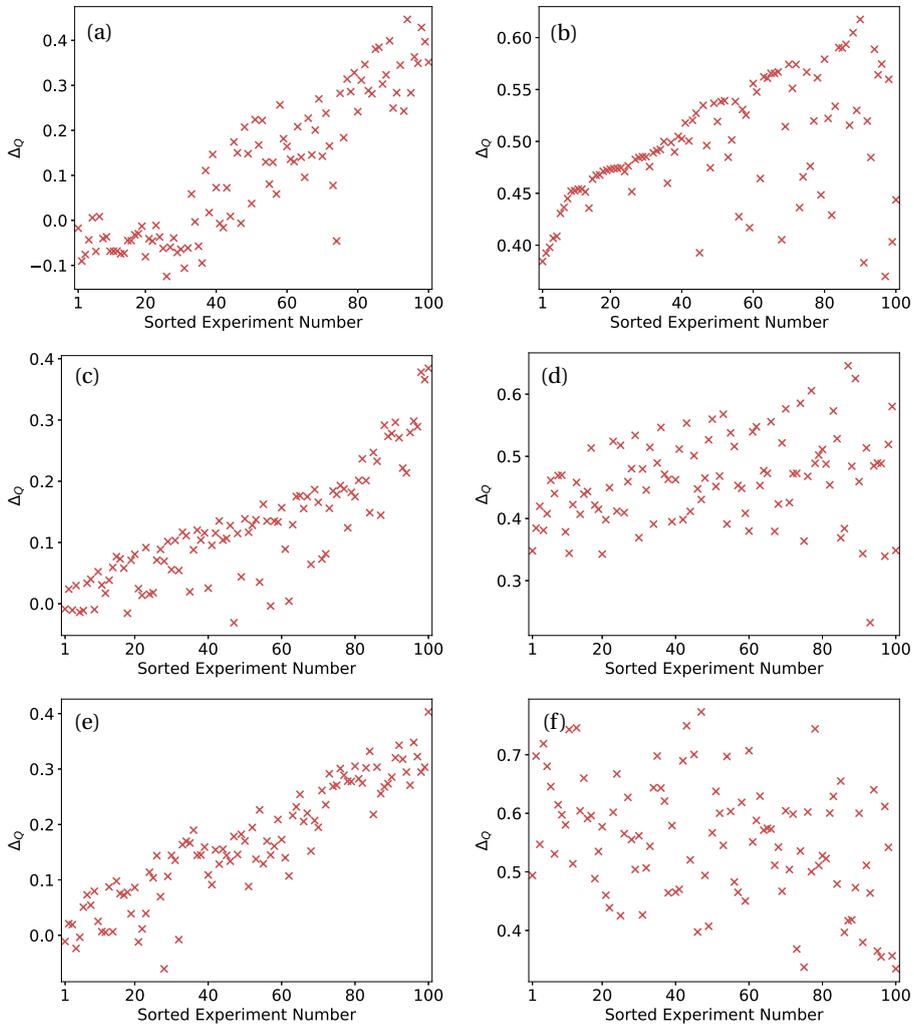


FIGURE 6.2:  $\Delta_Q$  for the improved protocol experiments sorted by the increasing values of  $\Delta V$  (cf. Fig. 6 in [Jat+20]). The data corresponds to (a)  $N=2$  and depths [16, 20]; (b)  $N=2$  and [74, 80]; (c)  $N=3$  and [6, 10]; (d)  $N=3$  and [47, 56]; (e)  $N=4$  and [6, 10]; and (f)  $N=4$  and [53, 66].

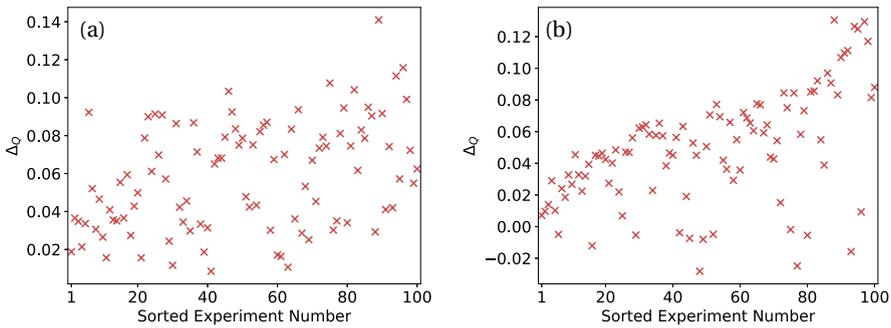


FIGURE 6.3:  $\Delta_Q$  for the standard protocol experiments sorted by the increasing values of  $\Delta V$ . The data corresponds to (a)  $N = 2$  and depths [72, 80], and (b)  $N = 3$  and depths [7, 11].

into account and mitigate errors better than the standard protocol. Similarly, for the 3 qubits case, comparing the standard protocol data from Fig. 6.3(b) with improved protocol data from Fig. 6.2(c), we observe that although there are some negative values of  $\Delta_Q$  for both, successful positive mitigation using the improved protocol mitigates the errors to a greater extent than the standard protocol.

## 6.6 Benchmarking

Testing prototype quantum computers through benchmarking is an important field in computing. Recent developments include comparing the superconducting and ion trap based quantum computers [Lin+17], generating maximally entangled states [CLLG19], application motivated circuit classes [Mil+21], quantum volume as a single number metric [Cro+19], visual benchmarking circuits [CBG21], scalable identity circuit operations [Mic+17; Pro+22], etc. The matrix  $M$  serves the purpose of error mitigation and benchmarking the quantum computers since all circuits used for the error mitigation are identity circuits composed of random gates. Since for an error prone quantum computer  $M$  has non-zero off-diagonal elements, by looking at  $M$  we can make rough estimates of the amount of errors involved in a quantum computer. The diagonal entries of the matrix can help identify a bias or preference towards certain states and the off diagonal entries bit flip or other errors. The matrix  $M$  does not capture phase related errors.

All the conducted experiments use  $2^{13}$  samples for a given number of qubits in a given range of circuit depths. In each experiment, a circuit with random gates is split in half and made into an identity circuit as outlined in the improved mitigation protocol in section 6.4. We start with all possible (computational) initial states of the quantum computer for each experiment and apply an identity circuit. For an error-free quantum computer, we would expect the same initial state to be measured at the end of the computation. However, due to errors, it is possible that other states are also measured. The matrix  $M$ , the average of the two matrices corresponding to each half of the identity circuits for an experiment, shows the other states obtained as non-zero off-diagonal elements. We calculate a matrix  $M'$  which is the average of the matrices from all the different random circuits.

In Fig. 6.4, we plot the matrix  $M'$  for a different number of qubits and range of circuit depths. Plots show  $M'$  in three dimensions where the height of a bar is the value of the corresponding element in the matrix. Shown in Fig. 6.4(a) are averaged results that are a

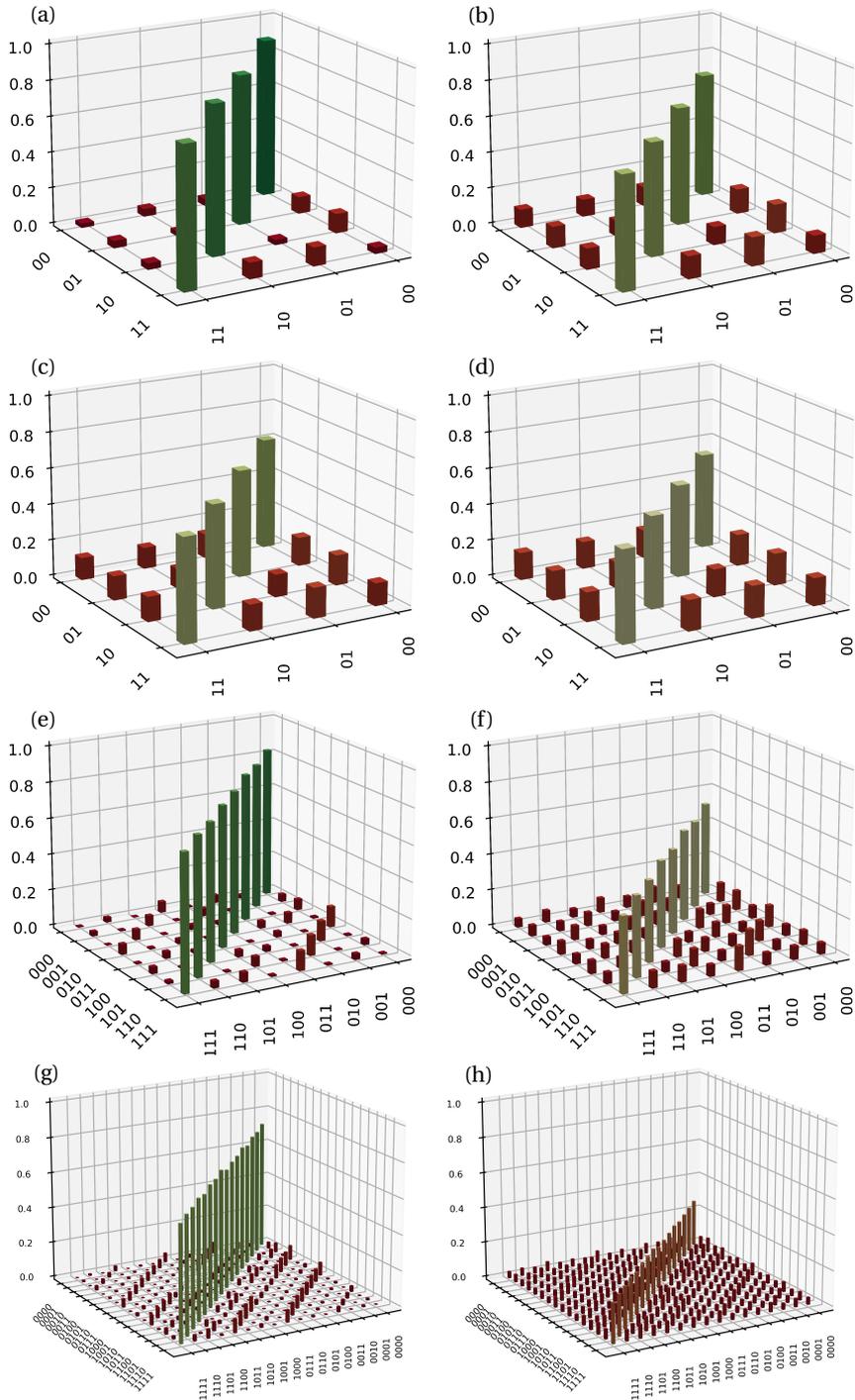


FIGURE 6.4: Normalised frequencies as a function of the states of a quantum computer. Data shows the values in the matrices in a three dimensional plot where the z-axis shows the frequencies. (a)  $N = 2$  and depth  $[8, 10]$ ; (b)  $N = 2$  and depth  $[16, 20]$ ; (c)  $N = 2$  and depth  $[35, 40]$ ; (d)  $N = 2$  and depth  $[74, 80]$ ; (e)  $N = 3$  and depth  $[6, 10]$ ; (f)  $N = 3$  and depth  $[47, 56]$ ; (g)  $N = 4$  and depth  $[6, 10]$ ; (h)  $N = 4$  and depth  $[53, 66]$ .

total of 100 experiments carried out for  $N = 2$  with circuits depths in the range 8 to 10. We observe the presence of small but finite off-diagonal entries. The errors in this depth range are relatively small given the relatively better performance of the quantum computer due to the small depth of the circuits. Observe that the amplitude of the off-diagonal elements is larger when they are one bit-flip away from the diagonal, i.e. 01 is one bit-flip away from 11. Similarly, the amplitude is smaller if a bitstring is two flips away. This behaviour is intuitive because a single bit-flip (error) is more likely than two single bit-flips.

In Fig. 6.4(b-d), we increase the circuit depth ranges (for  $N = 2$ ) in plots (b) 16 to 20, (c) 35 to 40, and (d) 74 to 80, respectively. We carry out a total of 100 experiments and repeat each experiment 10 times each with  $2^{13}$  samples. The  $M'$  is then the average of all these experiments. The amplitude of the off-diagonal elements increases as the depth is increased with a proportional decrease in the amplitude of the diagonal elements. Entries that are one bit-flip away are present in (b) but slowly disappear as the depth decreases in (d). This behaviour suggests that the output tends to become random and thus equally distributed as depth is increased.

In Fig. 6.4(e-f), we show the results for  $N = 3$  and depths 6 to 10 and 47 to 56, where 100 different experiments are repeated 10 and 3 times, respectively. The one bit-flip errors are more clearly visible in the plot (e), having relatively large amplitudes than other off-diagonal entries. Interestingly, some single bit-flip errors are more dominant than others, e.g. 011 vs 101, although both are one bit-flip away from 111. This shows the presence of other kind of errors. As the circuit depths increase for the  $N = 3$  case, the same behaviour is observed where the diagonal elements decrease in amplitude as was observed for  $N = 2$ .

In Fig. 6.4(g-h), we show the results for  $N = 4$  and depths 6 to 10 and 53 to 66, where 100 different experiments are repeated 10 and 3 times, respectively. Similar to Fig. 6.4(e), the single bit-flip errors are dominant. Although the depths for Fig. 6.4(g) are lower than the corresponding  $N = 3$  case (Fig. 6.4(e)), the amplitude of the diagonal entries is still smaller. For the last plot (h), there is a dramatic decrease in the amplitudes of the diagonal elements, suggesting a large presence of errors in the IBM Q quantum computer. In summary, the matrix  $M'$  gives a good pictorial representation for the presence of some type of errors in a quantum computer, at least for small systems.

### 6.6.1 Detecting bias

In the previous section, we used the off diagonal entries of the matrix to interpret errors in a quantum computer. In this section, we use the diagonal entries for yet another purpose. As a common practice in quantum computing, the state of all qubits is set to  $|0\rangle$  before any computation is carried out. Very often, the first step of variational algorithms is to change this initial state to some other initial state that offers an accelerated performance for the variational optimisation. For example, single Hadamard gates can be applied to all the qubits to set the quantum computer into a uniform superposition of all basis states. For the algorithms used in this work, initial states often represent the number of spins on a lattice or the occupational number of an atom or molecule. Such initial states are seldom a uniform superposition but often some combination of the computational states  $|0\rangle$  or  $|1\rangle$ . Benchmarking experiments on initial states which are a uniform superposition of computational (or basis) states, have led to the conjecture that some IBM quantum computers have a bias towards being in a particular state compared to others [Mic+19]. It is, therefore, interesting to question if such a bias also exists for initial states relevant for variational algorithms used in this work.

The experiments performed in [Mic+19] used a single layer of Hadamard gates to prepare the quantum computer in a uniform superposition of computational states before measurement. The analysis consisted of observing the weighted average of the relative frequency of

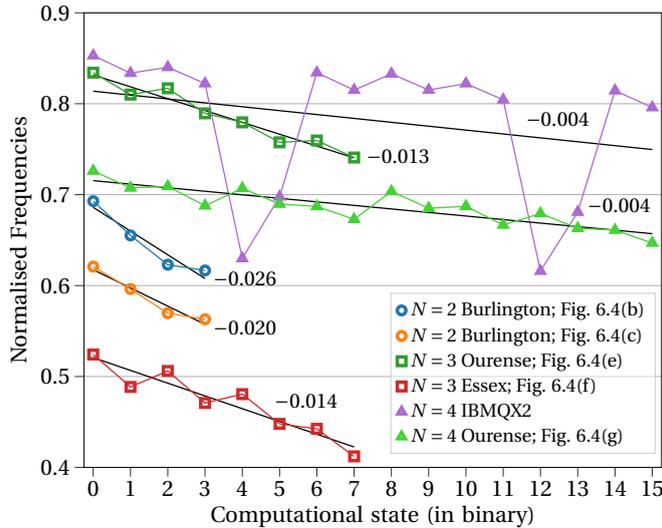


FIGURE 6.5: Normalised frequencies of obtaining a final state that is the same as the initial state as a function of the computational states for circuits consisting of several layers of randomly generated gates that perform an identity operation. The horizontal lines indicate the ideal outcome. The tilted lines and corresponding negative numbers are the linear fittings to the data points indicating the negative slope. Problems sizes and devices are listed in the legend along with the connection to Fig. 6.4.

the computational states as a function of the number of ones in the computational states. The experiments showed that the tested quantum computers had a preference (or bias) towards qubits being in the state  $|0\rangle$ . The benchmarking performed in this chapter differs as follows. Circuits consist of several layers of randomly generated gate operations that ultimately perform an identity operation, thus (ideally) matching the final and initial states. The total number of benchmarking circuits evaluated per experiment is  $2^N$ . The benchmark is an average over several different experiments. Our benchmarking techniques are similar in the sense that one seeks to quantify any bias in the quantum computer when observing the number of ones in the measured states. To achieve this objective, it is sufficient to observe the diagonal entries of the matrices.

The diagonal entries reflect the frequency of obtaining a final state that is the same as the initial state. The normalised frequency of all diagonal entries should be unity on an ideal quantum computer. However, an error prone quantum computer can deviate from the ideal outcome. In Fig. 6.5, we plot diagonal entries of the matrices considered for benchmarking. The legend shows the names and number of qubits of the different quantum computers used and connection to a previous figure (if existing). The architectures of the quantum computers are shown in Fig. 6.6. The data shows the normalised frequencies of finding the same final state initialised as a function of the computational states, labelled by a number corresponding to their binary representation. We observe that the outcomes are not ideal. The ideal outcome is a flat line  $y = 1$ . The frequency to obtain the same initial state decreases in proportion to the increasing binary number representation of the computational states across the different number of qubits, circuit depths, and quantum computers. A linear fitting to the

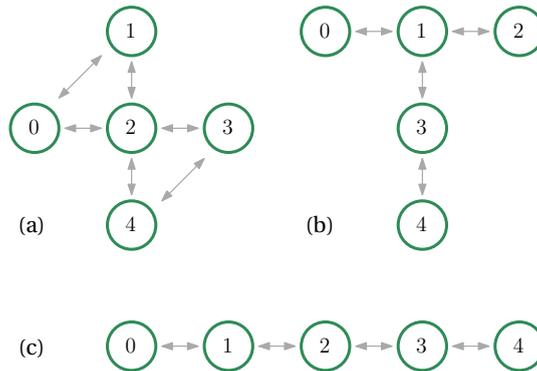


FIGURE 6.6: Different architectures of IBM Q quantum computers with 5 qubits used to produce results shown in this chapter. (a) IBMQX2 [5-qd]. (b) Burlington [5-qb], Ourense [5-qb], Essex [5-qc]. (c) Manila [5-qe]. The circles represent the qubits and the arrows the connections between them.

data points having negative slopes is suggestive to the naked eye of a consistent bias. The effect of increasing circuit depths is also visible as the data points for each  $N$  are lower for the larger circuit depths. These results imply that there are different error rates in terms of the appearance of unwanted bitstrings when initialising from different computational states. While the quantitative values differ slightly, all IBM Q devices show a bias towards states that contain more zeros than ones. This is consistent with the findings of [Mic+19].

We found an exceptional case of  $N = 4$  on the computer IBMQX2. We observed something more than an expected decline in the normalised frequencies as the binary number representation of the computational states increases. There is a sudden and significant dip in the normalised frequencies in a total of four states corresponding to binary representations 4, 5, 12 and 13. Given that the data is averaged over a large number of experiments shows that this behaviour appears systematic in the computer. IBMQX2 is no longer accessible so this dip's cause remains unknown.

## 6.6.2 Mean field model

Quantum computing benchmarks often use random circuits. One reason for that is the absence of interesting problems that can be solved using a very low number of gates supported by current quantum computers. In this section we develop a benchmarking problem that is based on a physical problem. We use the mean field model defined by the Hamiltonian given in Eq. (5.7) with  $J_{ij}^{\alpha} = 1$  for all  $\alpha \in \{x, y, z\}$ . This is essentially a solvable problem whose ground state energy is easy to compute. The ground state energy is given by the expression  $3(a - N)/2$  where  $a = 1$  for odd  $N$  and  $a = 0$  for even  $N$  [HD00].

After several trials we found an ansatz which is able to solve this problem using a circuit of only depth five [Jat+22b]. The reason why the ansatz has a low circuit depth is that it can be constructed from a set of operators instead of a sum of operators. The latter is difficult to implement on a quantum computer due to the large circuit depths that may be required. In contrast, the former is implementable with significantly lower depths. Specifically, the ansatz is given by the set of operators

$$A = \{\sigma_k^x \sigma_{k+1}^y; k = 1, 3, 5, \dots, N\}, \quad (6.7)$$

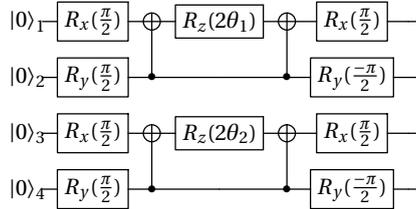


FIGURE 6.7: Circuit implementation for finding the ground state energy of the mean field Hamiltonian (see Eq. (5.7)) when  $N = 4$ .

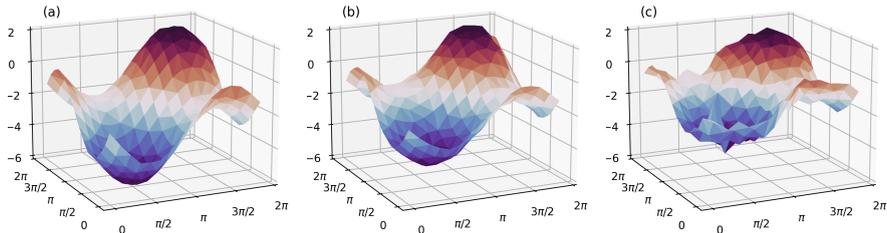


FIGURE 6.8: Energy landscapes for the mean field Hamiltonian (see Eq. (5.7)). The  $x$  and  $y$  axes show the parameter values and the  $z$  axis shows the energy scale. The data is obtained from (a) an ideal emulator, (b) IBM Q *Manilla* [5-qe], (c) IBM Q *Belem* [5-qa].

which are exponentiated to give the product

$$U(\theta) = \prod_k e^{-i\theta_k \sigma_k^x \sigma_{k+1}^y}. \quad (6.8)$$

The corresponding initial state is

$$|\Psi_0\rangle = |\dots 0101\rangle, \quad (6.9)$$

where the odd and even indexed qubits need to be initialized in the state  $|0\rangle$  and  $|1\rangle$ , respectively. The circuit implementation for  $N = 4$  is shown in Fig. 6.7. Clearly, entanglement of all the qubits of a quantum computer is not required. Furthermore, the benchmark is scalable since the circuit depth is independent of  $N$ . The ground state is located at  $\theta = \pi/2$ . Note that such an ansatz also finds the ground state energy of the Majumdar-Ghosh Hamiltonian [Maj70] where the ground state contains pairs of nearest-neighbour singlets [CEM84].

We performed the benchmarking on IBM Q quantum computers for  $N = 5$ . Since only two parameters were involved we scanned the entire energy landscape on a  $16 \times 16$  grid. The results are shown in Fig. 6.8(b-c). For comparison, results of an ideal emulator are shown in Fig. 6.8(a). The first benchmark is performed on IBM Q *Manilla* which has a linear structure (see Fig. 6.6(c)). This device has a *quantum volume* of 32 and is qualitatively able to reproduce the energy landscape obtained from the emulator. The landscape is smooth, as should be expected. The shape of the landscape deteriorates when using the IBM Q *Belem*, which has a quantum volume of 16, as shown in Fig. 6.8(b). The landscape is no longer smooth and has irregularities that may make the traversing of an optimisation algorithm difficult on

the landscape. The change in the shapes can be ascribed to errors in the quantum computers. It appears that computers that offer a higher quantum volume perform better on our benchmark.

## 6.7 Scalable mitigation

The standard and the improved protocols, as presented in this chapter until now, cannot be used for mitigating errors in quantum computers with a large number of qubits. The reason for the non-scalability of both protocols is the use of the matrix that requires an exponentially increasing number of entries to store the mitigation information. It is due to this that both the protocols do not satisfy requirement four from section 6.2. The protocols are not practically realisable in two aspects: (1) The storage space for the matrix grows exponentially with the number of qubits and, therefore, the protocols are not suitable for medium and large scale quantum computers of the near future, (2) a disproportionately large number of calibration circuit evaluations, in comparison to the circuit whose errors are to be mitigated, are required to fill all the elements of the matrix. Therefore, it is of interest to make the protocols scalable enough to go beyond small scale proof of concept demonstrations.

After the completion of the work presented until now in this chapter, a new technique was recently published in [Nat+21] that proposes a solution based on some reasonable assumptions which might be fulfilled for most practical circuits. This technique allows the standard protocol to be scalable. The technique can also be used for the improved protocol since the improved protocol builds upon the standard protocol. The technique is described in what follows.

### 6.7.1 Subspace reduction

The idea is to reduce the working subspace of the error mitigation protocol. Since the total number of states grows exponentially with the number of qubits, the protocol becomes intractable beyond a small number of qubits. In an error free quantum computer, only the intended states will appear as we sample the final state at the end of the computation. However, in error prone quantum computers, these states will be accompanied by some other states that may not be relevant for the computation and, therefore, render the computation erroneous. The idea is to restrict the focus on a limited number of states rather than all of them. In this section, we discuss techniques to achieve that aim.

#### Sparse matrix

The idea of subspace reduction essentially refers to reducing the size of the matrix by reducing the non-zero entries or setting several of its entries to zero. The resulting matrix will then be sparse and could be easily stored and used for mitigation. Such a sparse matrix overcomes the scaling problem of both the standard and the improved protocols. The guiding principle for reducing the entries of the mitigation matrix is to observe the noisy output of the quantum computer and create calibration circuits only for those bitstrings that appear in the noisy output.

For example, consider a circuit whose (error free) output state is 0110. We expect only this bitstring when sampling an ideal quantum computer. However, a noisy quantum computer, which the bit flip error model can describe, say for Hamming distance one, will also produce 0010, 0100, 0111, and 1110. Previously, without the subspace reduction, one would generate  $2^{4+1}$  calibration circuits to fill each column of the matrix  $M$ . Using the error model, one

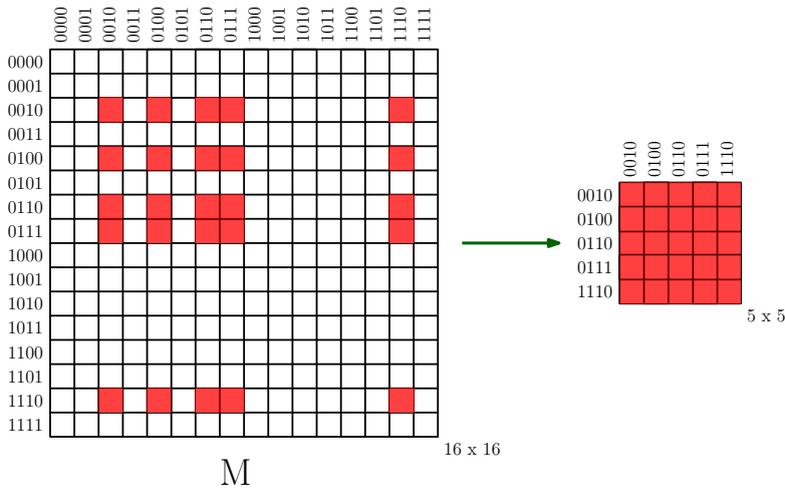


FIGURE 6.9: Visual representation of the sparse matrix created using the subspace reduction technique.

would need to use only  $2 \times (4 + 1)$  circuits, as shown in Fig. 6.9. The sparsity of the matrix due to the reduction is visible.

An essential question is if error models can be used for quantum computers to help expect a certain kind of behaviour in the output. We discuss the following error model that fulfils the criterion.

### Bit flip error model

An error model consisting of *bit flip* errors can be used to describe the error prone quantum computer. A bit flip error occurs when one of the bits in a bitstring undergoes a **NOT** transformation (flips). The concept of bit flips can be generalised in terms of *Hamming distance*. The Hamming distance between two bitstrings is the number of positions at which the corresponding bits are different. For example, the Hamming distance for a single bit flip is 1, for two bit flips is 2, and so on.

The evidence that the bit flip error model is a reasonable model for the IBM quantum computers is discussed in the benchmarking results of section 6.6. As expected, we observed that single bit flips are more frequent than double bit flips, which are in turn more frequent than triple bit flips, and so on. Therefore, we can expect that the noisy bitstrings from a quantum computer described by the bit flip error model are within a certain Hamming distance from the original error free bitstring. The number of possible bitstrings within a finite Hamming distance to a bitstring scale polynomially.

It is possible that bitstrings appear in the output of the calibration circuits that are not within a certain Hamming distance from the original bitstring. This is due to the fact that the new bitstrings are within (not within) a certain Hamming distance from the output of the calibration (original) circuits. For example, in Fig. 6.9, 1010 can appear when using the calibration circuit intended to output 0010. We will ignore such bitstrings and set their frequencies as zero in the matrix, even if they appear in a small but finite number of samples. This step will be undertaken using a threshold value that decides what entries will be set to zero. Each matrix column represents the sum of all the (normalised) frequencies and should sum

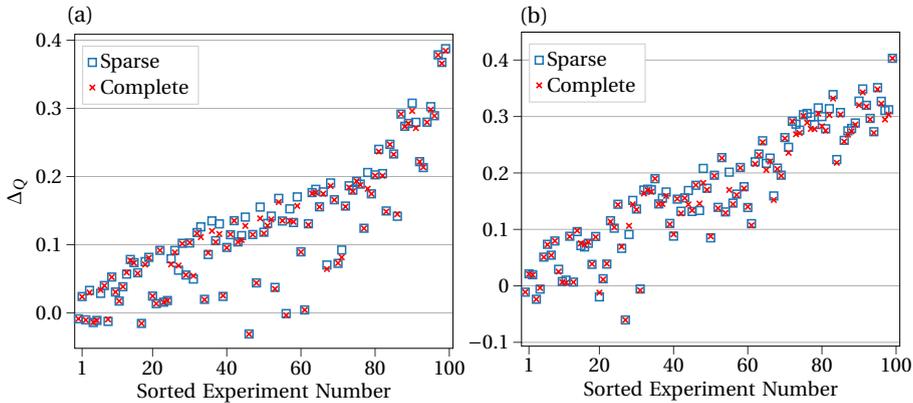


FIGURE 6.10: Comparison of mitigation as a function of sorted experiment numbers using sparse (squares) or complete (crosses) matrices. (a)  $N = 3$ , depth in range  $[6, 10]$ , and  $s = 4$ . (b)  $N = 4$ , depth in range  $[6, 10]$ , and  $s = 4$ .

to unity. By selectively removing some entries or setting them to zero, the columns should then be renormalised for consistency of Eq. 2.5. From an error mitigation perspective, renormalisation is unimportant, and we do not perform it. However, it would be interesting to investigate to what extent the renormalised or the not renormalised matrices would change the mitigation.

### 6.7.2 Demonstration

We demonstrate the subspace reduction technique for the improved mitigation protocol. We apply the technique to the data on which the improved protocol was used. By reusing the available data, a better comparison is possible. The subspace reduction technique can be called successful if the error mitigation using the sparse matrix offers similar results compared to the complete matrix. The results consider the subspace reduction technique when no renormalisation of the sparse matrix column vectors is used.

We apply the technique to the available data (see Fig. 6.4). We only show two plots as examples, but the subspace reduction works for all cases. The technique generates a sparse matrix with substantially fewer memory requirements than the complete matrix. In Fig. 6.10 we compare the strength of mitigation, as measured by  $\Delta_Q$  when either the complete matrix (crosses) or the sparse matrix (crosses) is used. In Fig. 6.10(a-b), we observe that the crosses and the squares overlap in a majority of the experiments. This overlap suggests no significant difference in the error mitigation outcomes using either the complete or the sparse matrices. The difference is minimal when there is no complete overlap. Overall, both plots in Fig. 6.10 show that the sparse matrices produced indistinguishable error mitigation results compared to the mitigation using complete matrices. This suggests that the subspace reduction technique was able to reproduce the results of the improved protocol. We thus obtain the benefit of having reduced memory storage that no longer scales exponentially with  $N$  with no substantial difference in error mitigation. It is important to note that since the improved protocol remains a heuristic, there is no guarantee that the sparse matrix will always produce positive error mitigation.

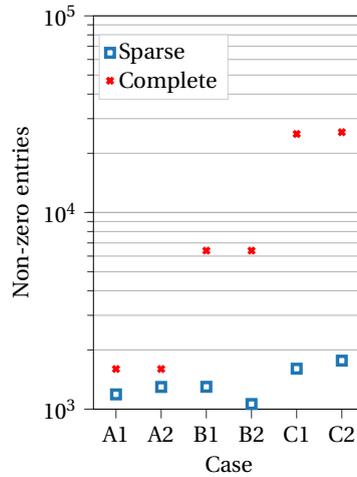


FIGURE 6.11: Scaling behaviour of the positive outcome of error mitigation in terms of the number of non zero entries in complete (crosses) or sparse (squares) matrices for 100 different experiments per case (A1,...,C2, for details see text).

### Threshold selection

A vital element of this quantitative assessment is the appropriate choice of the threshold value which determines the values that appear in the sparse matrix. Values below the threshold will be set to zero when filling the columns of the sparse matrix. It is essential to select an appropriate value. The threshold should be chosen by taking into account the magnitude of noise in a quantum computer.

We choose the threshold of our experiments as follows. Prepare a circuit such that only one bitstring (state) should ideally be output during measurement. Other bitstrings in a noisy quantum computer will also appear along with the ideal output due to errors. The threshold should then be such that all other noisy bitstrings not intended to appear in the output are set to zero. Note that setting the threshold arbitrarily high would be disadvantageous because it may then include those bitstrings that are part of the output and not noise. We choose the threshold depending on the actual performance of the quantum computer by multiplying the standard deviation associated with the measurement process with an appropriate natural number. The measurements for our data were performed using  $2^{13}$  samples. The threshold was then set to  $s/\sqrt{8192}$ , where  $s$  is the natural number.

### Scaling behaviour

In this section we quantify the improvement using the subspace reduction technique. We analyse the available data from the experiments performed in the previous sections. To study the improvement, we compare the number of non-zero entries in the matrices corresponding to either using the complete matrix or using the sparse matrix.

After setting an appropriate value of  $s$ , we compare the number of non-zero entries in the complete and sparse matrices that would be required for all the experiments. For example, in theory, if 100 experiments are performed for  $N = 2$ , the maximum number of non-zero entries required is  $16 \times 100$ . In practice, it is possible that some bitstrings do not appear and therefore appear as zeros in the complete matrix. In the data we present, we count the actual number

of non-zero entries that appear in the complete matrix and not the theoretical value. The data is illustrated in Fig. 6.11. Cases A1 and A2 correspond to data collected on 100 random experiments for  $N = 2$ , averaged over 10 repetitions for depth ranges [16,20] and [35,40], with the choices of  $s = 10$  and  $s = 12$ , respectively. Since the underlying matrices are of size  $4 \times 4$ , one should not expect a large difference in the number of non-zero entries per experiment between the sparse and complete matrices. We observed only a small difference in the data. However, the difference starts to widen as  $N$  increases. Cases B1 and B2 correspond to data collected on 100 random experiments for  $N = 3$ , averaged over 10 and 3 repetitions for depth ranges [6,10] and [47,56], with the choices of  $s = 4$  and  $s = 8$ , respectively. Similarly, cases C1 and C2 correspond to data collected on 100 random experiments for  $N = 4$ , averaged over 5 and 10 repetitions for depth ranges [4,6] and [6,10], with the choices of  $s = 2$  and  $s = 4$ , respectively. The appropriateness of different choices of  $s$  for each case was justified by verifying that there was no distinguishable difference in the mitigation outcomes using either the complete or the sparse matrices. The number of non-zero entries in the complete matrices, as shown by the logarithmic scale in Fig. 6.11, grows exponentially. In contrast, the number of non-zero entries in the corresponding sparse matrices appears to grow only polynomially.

### 6.7.3 Discussion

In this section, we demonstrated a scalable error mitigation protocol. It overcomes the drawbacks of the improved protocol (section 6.3) in two ways: (1) The storage space for the matrix no longer grows exponentially with the number of qubits and, therefore, the protocols that employ subspace reduction are suitable for medium and large scale quantum computers of the near future, (2) a proportionate number of calibration circuits evaluations are sufficient in practical cases to fill all the relevant elements of the matrix.

It turns out that the threshold is an important element of the subspace reduction technique. For a relatively small threshold, almost all values in the complete matrix will be used and the new matrix will not be sufficiently sparse. Without sufficient sparsity, the protocol will not be scalable. On the other hand, for relatively larger threshold, even those values which are part of the intended output will be neglected and the computation will exclude bitstrings relevant for the circuit being mitigated. It is difficult to determine a priori a fixed value of the threshold that works for a large number of cases. It is more practical to determine the threshold by performing benchmarking experiments, e.g. that output a single state.

The usage of the subspace reduction can be envisioned for three cases. First, consider the case of a circuit which produces a single output state. Error mitigation protocols presented in this chapter show that there is a near perfect mitigation in such cases. However, such cases are often not of practical interest since entanglement is essential to many quantum algorithms. Second, consider circuits which produce states polynomial in number compared to the total number of possible states. The scalable mitigation idea rests on the premise that the total number of sampled states relevant in computation for a given circuit  $C_g$  do not scale exponentially. It is assumed that only a polynomial number of states are sufficient for implementing an algorithm. These cases are of practical interest, and our results show that the subspace reduction technique offers indistinguishable performance using a much sparser matrix allowing scalability. Third, consider circuits which produce states distributed over a large majority of all possible states. We observe that error mitigation protocols fail in this case. However, this case is not of practical interest because if an algorithm produces such a distribution, then such an algorithm will require an impractically large number of samples, making them useless for quantum computing.

## 6.8 Summary

In this chapter, we introduced, demonstrated, and assessed the quality of a scalable error mitigation protocol. A set of seven criteria was introduced for error mitigation protocols. The demonstrated protocol was developed in view of the criteria. The requirement 1 was fulfilled numerically in the sense that several different experiments were considered to infer efficacy of the protocol inductively. No theoretical guarantees were argued in favour of the protocol. We also did not show that such a guarantee does not or cannot exist. Thus, the protocol remains a heuristic. A useful side product of the introduced protocol was its ability to benchmark devices, which is of independent interest. We also demonstrated benchmarking results using a mean field model. Given the various benefits of the error mitigation protocol introduced in this chapter, it can be concluded that it is a useful heuristic.

## Chapter 7

# Conclusion and outlook

### Hybrid variational methods

The aim of the work presented in this dissertation was to develop methods that can make use of current and near-future quantum computers. These computers are noisy and cannot accommodate large depth circuits common to conventional quantum algorithms. Building on the framework of variational methods, we developed hybrid algorithms that are noise tolerant and can be run on current quantum computers. Successful completion of these objectives required finding problem-specific ansätze, initial states, and optimisation algorithms. Briefly summarised below are the efforts undertaken, and lessons learnt.

#### The efforts

We outlined a systematic way to build circuits given an ansatz in the Pauli basis. Using that, we investigated an adaptive ansatz construction method applied to the Heisenberg model to predict a set of unitary operators that is likely to reasonably approximate the model's ground state energy as the problems are scaled up. This adaptive method can be used for other problems as well. We numerically demonstrated the ability of the ansatz to find the ground state energy of one-, two-, and three-dimensional antiferromagnetic Heisenberg models.

We were able to set up and use an ansatz to estimate the ground state energy of some molecules and the Hubbard model. This was achieved with the help of open-source packages. We demonstrated a prototype case of finding the dissociation energy of the Li-Li molecule. For a few molecules we also timed the amount of classical computation required to prepare the problem before it can be solved by the quantum computer. Afterwards, we determined the time taken by a quantum computer emulator to calculate the energy once for relatively large molecules.

Another aim of this dissertation was to study the effect of noise when using variational methods. We introduced noise to the simulations to study its effect on the ability of variational methods to find the ground state energy. The relevant questions addressed were regarding the accuracy of the final energy obtained and the effect of noise on the ability of the optimisation algorithms to navigate the landscape. We tested the former and the latter when finding the ground state energy of the water molecule and the Heisenberg model, respectively.

#### The lessons

The flexibility of using any ansatz offered by the variational principle must be balanced by a proper choice that fulfils the following two criteria; (a) the ansatz should be able to express the true ground state of the model, and (b) the energy landscape should be navigable enough

for an optimisation algorithm to converge close to the global minimum. For (a), the adaptive method we employed does not guarantee that a sufficiently expressive ansatz can always be found. Therefore, clever and intuitive choices are highly relevant for the performance of variational methods. For (b), the energy landscape becomes fixed when a decision for the ansatz and problem Hamiltonian has been made. It is known that classical optimisation is a hard problem in itself. Therefore, since the landscape cannot be changed, it is important that a good starting point is found from for which a navigable path exists toward the global minimum. Preferably, such a path should not have local minima in which the optimisation algorithms get trapped. We learnt that obtaining performance guarantees for (a) and (b) in general is difficult.

We learnt that the optimisation algorithm plays a crucial role in the ability to find the ground state energy. In this regard, the choice of quasi-Newton methods is suitable when gradient computation is feasible. Additionally, the choice of an ansatz is highly relevant. Not only should the total number of gates be as low as possible, but the connectivities between the qubits should take into account the physical connectivities in the hardware.

As expected, the presence of noise makes the energy landscape harder to navigate, leading to poor approximations of the ground state energy. We found that although variational methods are resistant to noise, the loss in accuracy of the results may be beyond what can be tolerated in practical applications. For example, in the case of the water molecule, we found that the final energy was no longer within chemical accuracy when realistic gate errors were introduced. For practical applications, the ground state energy will be unknown, and there is no way to ascertain a priori that the minimum energy found is within chemical accuracy. The tolerance for the loss of accuracy is problem-dependent.

In summary, we learnt the lesson that variational methods will not always be successful without sufficient background knowledge about the problem. Specific settings and tuning from the user are helpful but do not guarantee the accurate estimation of ground state energies at the first attempt.

## The drawbacks

The general benefits obtained from variational methods are accompanied by the corresponding drawbacks. While they give the benefit of lower circuit depths and can potentially be run on near-term noisy hardware, the drawback is the loss of guarantee of finding the ground state energy. As mentioned above, the loss of guarantee arises from the inability to find a sufficiently expressive ansatz and the inability to navigate through a rugged energy landscape. To avoid the problem for the Heisenberg model, we prepared the initial state according to the Néel state and found that the path towards a minimum appears navigable for the cases we tested. Similarly, classical approximations to the ground state for the water molecule and the Hubbard model are helpful.

In line with the proposed aims, we identified two major problems faced by variational algorithms; (a) we numerically demonstrated the existence of local minima for the Heisenberg model and the water molecule, and (b) we found evidence of Barren plateaus in the energy landscape of a large three-dimensional isotropic antiferromagnetic lattice. In both cases, the optimisation algorithm becomes trapped in the energy landscape and cannot make downhill progress, a necessity to improve the ground state energy estimate in the context of variational methods.

## The improvements

The main contribution reported in this dissertation is the development of the *quasi-dynamical evolution*, which systematically improves the ground state energy estimate of our problems. To overcome the deficiencies mentioned above, the new approach helps making downhill progress even after standard variational algorithms get stuck in local minima. Quasi-dynamical evolution is inspired by quantum annealing. The quasi-dynamics helps to build a 'trajectory' towards the ground state by adding new operators to an ansatz. There is no restriction on the number or type of unitary operators allowed to be added for each cycle. We call this technique a heuristic because we currently do not have a method to find the most suitable set of unitary operators for each evolution cycle.

We applied the heuristic to improve the previous ground state energy estimates obtained for the antiferromagnetic Heisenberg model. We found that even a simple choice of the same set of unitary operators for each new cycle helps improve the ground state energy estimate. We compared the time taken by the emulator to find the ground state energy to the time taken by a hypothetical fully functional quantum computer. We considered realistic gate execution times and the number of samples. Although it is clear that the hypothetical computer will perform the task faster than any emulator, the former does not overtake an emulator for up to 36 qubits.

## Error mitigation and benchmarking

Another aim of the work presented in this dissertation was to develop error mitigation protocols for current and near-future quantum computers. In the current NISQ era of the hardware, it makes more sense to implement error mitigation than error correction. To develop effective mitigation protocols, we outlined seven criteria that any proper protocol to recover the ideal results should fulfil. The standard matrix-based protocol mitigates state preparation and measurement errors only. We proposed, implemented, and tested an improved protocol capable of capturing most other errors as well. These tests were performed on actual quantum computers through cloud-based access. Our tests indicate that error mitigation is not guaranteed by the protocol. Nevertheless, the mitigation was helpful in most of the randomised circuits tested.

The matrix from the improved protocol also serves the purpose of benchmarking the quantum computers. Different biases of the computer can be understood by looking at both diagonal and off-diagonal entries of the matrix. We also developed and implemented a mean-field model based benchmarking test, a problem whose ground state energy is easy to calculate by a circuit that is small enough to be implementable on current computers.

After others developed a scalable way to implement matrix-based protocols independently, we incorporated it into our protocol, making it scalable. We demonstrated scalability on the same data we collected for the improved protocol. Since our protocol does not completely fulfil all the criteria for an ideal protocol, there is room for future work.

## Outlook

Variational methods make current and near-term quantum computation possible. They offer broad flexibility for solving problems. The flexibility also comes with a responsibility to identify those configurations that will produce reasonable results. It is in this direction that future research should be directed.

The variational principle does not require that the ansatz must be a well-defined circuit made from the Pauli operators, as used in this dissertation. Additionally, cloud accessible superconductivity based quantum computing platforms allow the user to send pulses to the devices. Combining these two aspects, one can imagine defining an ansatz in terms of an optimised set or series of parametrised pulses that act as the ansatz to find the ground state energies.

In its current state of development, quasi-dynamical evolution remains a heuristic. The reason is that it is unknown which set of unitary operators will bring the largest decrease in the energy with each cycle. Future work should find a systematic way to choose sets of operators that guarantee finding the ground state energy in a finite and reasonable number of cycles.

Error correction and mitigation are often contrasted against each other. Instead, they may be made complementary since their workings do not overlap, and they have a common goal of reducing errors or noise.

In Wonderland\*, the Duchess admonished Alice,  
*"Everything's got a moral ..."*;

In Hilbert space, the author can suggest her this,  
*"Noisy quantum computers have purpose ..."*;  
*"if only you can find it".*

---

\*Quoted from [Car65].

## Appendix A

# Sample files

### Circuit preparation algorithm

#### Sample input file

Shown below is a sample input file that is given as input to the algorithm. It consists of three columns, where  $S$  is the string,  $c$  is the coefficient, and  $p$  is the parameter index. Each row represents an operator from Eq. (3.29).

---

	$S$	$c$	$p$
1	IIIIIIIIIXY	1.2	1
2	IIIIIIIIIXY	0.4	1
3	IIIIIXIYYI	1.5	2
4	IIIIIXIYYI	1.0	3
5	IIIIIXYIII	1.5	4
6	...		
7	IIIIXYIIIII	1.0	5
8	YIIIXIIIII	0.5	6
9	YIIIXIIIII	0.7	7
10	YIXIIIIIII	1.3	6
11	YXIIIIIIIII	1.5	4
12			

---

The input file shows that each operator can be assigned an arbitrary coefficient and two operators can share the same parameter.

#### Sample OpenQASM file

Shown below is a sample output file of the automatic circuit preparation algorithm that stores the circuit information in a standard quantum assembly language. The header defines the version of the language and declares the total number of qubits involved. The gate operations start from line 5 in the example shown below.

---

```

1 OPENQASM 2.0;
2 include "qelib1.inc";
3 qreg q[16];
4 creg c[16];
5 u2(-pi/2,pi/2) q[0];
6 u2(pi/2,-pi/2) q[1];
7 u2(-pi/2,pi/2) q[2];

```

```

8 u2(pi/2,-pi/2) q[3];
9 ...

```

---

### Sample Hamiltonian file

Shown below is a sample Hamiltonian input file that looks similar to the input file of the circuit preparation algorithm. The only difference lies in the fact that  $c$  and  $p$  are now the real and imaginary coefficients of the string.

---

```

1 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIXX 1. 0.
2 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIYY 1. 0.
3 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIZZ 1. 0.
4 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIXXI 1. 0.
5 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIYYI 1. 0.
6 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIZZI 1. 0.
7 ...
8 IIIIIIZIIIIIIIIIIIIIIIIIIIIIIIZIIIII 1. 0.
9 IIIIIZIIIIIIIIIIIIIIIIIIIIIIIZIIIII 1. 0.
10 IIIIZIIIIIIIIIIIIIIIIIIIIIIIZIIIII 1. 0.
11 IIZIIIIIIIIIIIIIIIIIIIIIIIZIIIII 1. 0.
12 IIZIIIIIIIIIIIIIIIIIIIIIIIZIIIII 1. 0.
13 IZIIIIIIIIIIIIIIIIIIIIIIIZIIIII 1. 0.
14 XIIIIIIIIIIIIIIIIIIIIIIIXIIIIIIIIIIII 1. 0.

```

---

The shown example is for a 40 qubits Hamiltonian. The emulator based on JUQCS also takes this as an input file.

## Appendix B

# Equivalence to parameter shift rule

The form of the function for each term in the ansatz is given by Eq. (4.26),

$$f(\theta) = a \sin(\theta + \phi) + c. \quad (\text{B.1})$$

The same function gives

$$f\left(\theta + \frac{\pi}{2}\right) = a \sin\left(\theta + \frac{\pi}{2} + \phi\right) + c, \quad (\text{B.2})$$

and

$$f\left(\theta - \frac{\pi}{2}\right) = a \sin\left(\theta - \frac{\pi}{2} + \phi\right) + c. \quad (\text{B.3})$$

By subtracting Eq. (B.3) from Eq. (B.2), we get

$$\begin{aligned} f\left(\theta + \frac{\pi}{2}\right) - f\left(\theta - \frac{\pi}{2}\right) &= a \sin\left(\theta + \frac{\pi}{2} + \phi\right) - a \sin\left(\theta - \frac{\pi}{2} + \phi\right) \\ &= a \left( \sin\left(\left(\theta + \phi\right) + \frac{\pi}{2}\right) - \sin\left(\left(\theta + \phi\right) - \frac{\pi}{2}\right) \right) \\ &= a \left( \sin(\theta + \phi) \cos\left(\frac{\pi}{2}\right) + \cos(\theta + \phi) \sin\left(\frac{\pi}{2}\right) \right. \\ &\quad \left. - \sin(\theta + \phi) \cos\left(\frac{\pi}{2}\right) + \cos(\theta + \phi) \sin\left(\frac{\pi}{2}\right) \right) \\ &= 2a \cos(\theta + \phi) \sin\left(\frac{\pi}{2}\right) \\ &= 2a \cos(\theta + \phi). \end{aligned} \quad (\text{B.4})$$

Dividing by 2 on both sides of Eq. (B.4), we get

$$\frac{f\left(\theta + \frac{\pi}{2}\right) - f\left(\theta - \frac{\pi}{2}\right)}{2} = a \cos(\theta + \phi) = \nabla f. \quad (\text{B.5})$$



# Bibliography

- [1-q] 1-qubit backend: IBM Q team, "IBM Q 1 Armonk backend specification V1.1.0," 2019.
- [5-qa] 5-qubit backend: IBM Q team, "IBM Q 5 Belem backend specification V1.0.30," 2021.
- [5-qb] 5-qubit backend: IBM Q team, "IBM Q 5 Burlington backend specification V1.1.4," 2019.
- [5-qc] 5-qubit backend: IBM Q team, "IBM Q 5 Essex backend specification V1.0.1," 2019.
- [5-qd] 5-qubit backend: IBM Q team, "IBM Q 5 Ibmqx2 backend specification V1.3.0," 2019.
- [5-qe] 5-qubit backend: IBM Q team, "IBM Q 5 Manila backend specification V1.0.26," 2022.
- [5-qa] 5-qubit backend: IBM Q team, "IBM Q 5 Ourense backend specification V1.0.1," 2019.
- [Abd+11] B. Abdo, F. Schackert, M. Hatridge, C. Rigetti, and M. Devoret. "Josephson amplifier for qubit readout". *Applied Physics Letters* **99**, 162506. 2011.  
DOI: 10.1063/1.3653473.
- [ABN19] C. E. Agrapidis, J. van den Brink, and S. Nishimoto. "Ground state and low-energy excitations of the Kitaev-Heisenberg two-leg ladder". *Phys. Rev. B* **99**, 224418. 2019.  
DOI: 10.1103/PhysRevB.99.224418.
- [AG+05] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon. "Simulated Quantum Computation of Molecular Energies". *Science* **309**, 1704–1707. 2005.  
DOI: 10.1126/science.1113479.
- [AGV20] P. Atchade Adelomou, E. Golobardes Ribé, and X. Vilasis Cardona. "Using the Variational-Quantum-Eigensolver (VQE) to Create an Intelligent Social Workers Schedule Problem Solver". *Hybrid Artificial Intelligent Systems*. Ed. by E. A. de la Cal, J. R. Villar Flecha, H. Quintián, and E. Corchado. Cham: Springer International Publishing, 245–260. 2020.  
DOI: 10.1007/978-3-030-61705-9\_21.
- [Ale+] G. Aleksandrowicz et al. Qiskit: An Open-source Framework for Quantum Computing. Version 0.18. 2019.  
DOI: 10.5281/zenodo.2562111.
- [AMM14] M. Amy, D. Maslov, and M. Mosca. "Polynomial-Time T-Depth Optimization of Clifford+T Circuits Via Matroid Partitioning". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **33**, 1476–1489. 2014.  
DOI: 10.1109/TCAD.2014.2341953.

- [Ana+22] A. Anand, P. Schleich, S. Alperin-Lea, P. W. K. Jensen, S. Sim, M. Díaz-Tinoco, J. S. Kottmann, M. Degroote, A. F. Izmaylov, and A. Aspuru-Guzik. “A quantum computing view on unitary coupled cluster theory”. *Chem. Soc. Rev.* **51**, 1659–1684. 2022.  
DOI: 10.1039/D1CS00932J.
- [Ans+19] E. Anschuetz, J. Olson, A. Aspuru-Guzik, and Y. Cao. “Variational Quantum Factoring”. *Quantum Technology and Optimization Problems*. Ed. by S. Feld and C. Linnhoff-Popien. Cham: Springer International Publishing, 74–85. 2019.  
DOI: 10.1007/978-3-030-14082-3\_7.
- [Arc09] Archimedes. *The Works of Archimedes: Edited in Modern Notation with Introductory Chapters*. Ed. by T. L. Heath. Cambridge Library Collection - Mathematics. Cambridge University Press. 2009.  
DOI: 10.1017/CBO9780511695124.
- [Arr+20] A. Arrasmith, L. Cincio, R. D. Somma, and P. J. Coles. “Operator Sampling for Shot-frugal Optimization in Variational Algorithms”. *arXiv:2004.06252*. 2020.  
DOI: 10.48550/ARXIV.2004.06252.
- [Arr+21] A. Arrasmith, M. Cerezo, P. Czarnik, L. Cincio, and P. J. Coles. “Effect of barren plateaus on gradient-free optimization”. *Quantum* **5**, 558. 2021.  
DOI: 10.22331/q-2021-10-05-558.
- [Aru+19] F. Arute et al. “Quantum supremacy using a programmable superconducting processor”. *Nature* **574**, 505–510. 2019.  
DOI: 10.1038/s41586-019-1666-5.
- [Aru+20] F. Arute et al. “Hartree-Fock on a superconducting qubit quantum computer”. *Science* **369**, 1084–1089. 2020.  
DOI: 10.1126/science.abb9811.
- [AS21] S. Abel and M. Spannowsky. “Quantum-Field-Theoretic Simulation Platform for Observing the Fate of the False Vacuum”. *PRX Quantum* **2**, 010349. 2021.  
DOI: 10.1103/prxquantum.2.010349.
- [BA04] J. S. Bell and A. Aspect. *Speakable and Unspeakable in Quantum Mechanics: Collected Papers on Quantum Philosophy*. 2nd ed. Cambridge University Press. 2004.  
DOI: 10.1017/CBO9780511815676.
- [Bab37] C. Babbage. “On the Mathematical Powers of the Calculating Engine”. *The Origins of Digital Computers*, 19–54. 1837.  
DOI: 10.1007/978-3-642-61812-3\_2.
- [Bae+20] J. H. Bae, P. M. Alsing, D. Ahn, and W. A. Miller. “Quantum circuit optimization using quantum Karnaugh map”. *Scientific Reports* **10**, 15651. 2020.  
DOI: 10.1038/s41598-020-72469-7.
- [Bal14] L. E. Ballentine. *Quantum Mechanics*. 2nd. World Scientific. 2014.  
DOI: 10.1142/9038.
- [Bar+95] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. “Elementary gates for quantum computation”. *Phys. Rev. A* **52**, 3457–3467. 1995.  
DOI: 10.1103/PhysRevA.52.3457.

- [BD88] T. Barnes and G. J. Daniell. “Numerical solution of spin systems and the  $S=1/2$  Heisenberg antiferromagnet using guided random walks”. *Phys. Rev. B* **37**, 3637–3651. 1988.  
DOI: 10.1103/PhysRevB.37.3637.
- [BDN12] I. Bloch, J. Dalibard, and S. Nascimbène. “Quantum simulations with ultracold quantum gases”. *Nature Physics* **8**, 267–276. 2012.  
DOI: 10.1038/nphys2259.
- [Bel64] J. S. Bell. “On the Einstein Podolsky Rosen paradox”. *Physics Physique Fizika* **1**, 195–200. 1964.  
DOI: 10.1103/PhysicsPhysiqueFizika.1.195.
- [Ben+19] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini. “Parameterized quantum circuits as machine learning models”. *Quantum Science and Technology* **4**, 043001. 2019.  
DOI: 10.1088/2058-9565/ab4eb5.
- [Ben73] C. H. Bennett. “Logical Reversibility of Computation”. *IBM Journal of Research and Development* **17**, 525–532. 1973.  
DOI: 10.1147/rd.176.0525.
- [Ben80] P. Benioff. “The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines”. *Journal of Statistical Physics* **22**, 563–591. 1980.  
DOI: 10.1007/BF01011339.
- [Ben82] P. Benioff. “Quantum mechanical hamiltonian models of turing machines”. *Journal of Statistical Physics* **29**, 515–546. 1982.  
DOI: 10.1007/BF01342185.
- [Ben93] G. L. Bendazzoli. “The variational principle illustrated by simple examples”. *Journal of Chemical Education* **70**, 912. 1993.  
DOI: 10.1021/ed070p912.
- [Ber+18] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri, K. McKiernan, J. J. Meyer, Z. Niu, A. Száva, and N. Killoran. “PennyLane: Automatic differentiation of hybrid quantum-classical computations”. *arXiv:1811.04968*. 2018.  
DOI: 10.48550/ARXIV.1811.04968.
- [Ber+22] N. F. Berthussen, T. V. Trevisan, T. Iadecola, and P. P. Orth. “Quantum dynamics simulations beyond the coherence time on noisy intermediate-scale quantum hardware by variational Trotter compression”. *Phys. Rev. Research* **4**, 023097. 2022.  
DOI: 10.1103/PhysRevResearch.4.023097.
- [Bet31] H. Bethe. “Zur Theorie der Metalle - I. Eigenwerte und Eigenfunktionen der linearen Atomkette”. *Zeitschrift für Physik* **71**, 205–226. 1931.  
DOI: 10.1007/BF01341708.
- [BF28] M. Born and V. Fock. “Beweis des Adiabatenatzes”. *Zeitschrift für Physik* **51**, 165–180. 1928.  
DOI: 10.1007/BF01343193.
- [BHT98] G. Brassard, P. Høyer, and A. Tapp. “Quantum counting”. *Automata, Languages and Programming*. Ed. by K. G. Larsen, S. Skyum, and G. Winskel. Berlin, Heidelberg: Springer Berlin Heidelberg, 820–831. 1998.

- [Bia+17] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. “Quantum machine learning”. *Nature* **549**, 195–202. 2017.  
DOI: 10.1038/nature23474.
- [BK02] S. B. Bravyi and A. Y. Kitaev. “Fermionic Quantum Computation”. *Annals of Physics* **298**, 210–226. 2002.  
DOI: 10.1006/aphy.2002.6254.
- [BK21] L. Bittel and M. Kliesch. “Training Variational Quantum Algorithms Is NP-Hard”. *Phys. Rev. Lett.* **127**, 120502. 2021.  
DOI: 10.1103/PhysRevLett.127.120502.
- [Blu01] S. Blundell. *Magnetism in Condensed Matter*. Oxford Master Series in Condensed Matter Physics. Oxford University Press. 2001.
- [BM+18] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. E. O’Brien. “Low-cost error mitigation by symmetry verification”. *Phys. Rev. A* **98**, 062339. 2018.  
DOI: 10.1103/PhysRevA.98.062339.
- [BM22] J. L. Bosse and A. Montanaro. “Probing ground-state properties of the kagome antiferromagnetic Heisenberg model using the variational quantum eigensolver”. *Phys. Rev. B* **105**, 094409. 2022.  
DOI: 10.1103/PhysRevB.105.094409.
- [BO27] M. Born and R. Oppenheimer. “Zur Quantentheorie der Molekeln”. *Annalen der Physik* **389**, 457–484. 1927.  
DOI: 10.1002/andp.19273892002.
- [BP+19] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles. “Variational Quantum Linear Solver”. *arXiv:1909.05820*. 2019.  
DOI: 10.48550/ARXIV.1909.05820.
- [BR12] R. Blatt and C. F. Roos. “Quantum simulations with trapped ions”. *Nature Physics* **8**, 277–284. 2012.  
DOI: 10.1038/nphys2252.
- [Bra+17] S. Bravyi, J. M. Gambetta, A. Mezzacapo, and K. Temme. “Tapering off qubits to simulate fermionic Hamiltonians”. *arXiv:1701.08213*. 2017.  
DOI: 10.48550/ARXIV.1701.08213.
- [Bra19] A. I. Braginski. “Superconductor Electronics: Status and Outlook”. *Journal of Superconductivity and Novel Magnetism* **32**, 23–44. 2019.  
DOI: 10.1007/s10948-018-4884-4.
- [Bra+21] S. Bravyi, S. Sheldon, A. Kandala, D. C. McKay, and J. M. Gambetta. “Mitigating measurement errors in multiqubit experiments”. *Phys. Rev. A* **103**, 042605. 2021.  
DOI: 10.1103/PhysRevA.103.042605.
- [BS88] T. Barnes and E. S. Swanson. “Two-dimensional Heisenberg antiferromagnet: A numerical study”. *Phys. Rev. B* **37**, 9405–9409. 1988.  
DOI: 10.1103/PhysRevB.37.9405.
- [Cad+20] C. Cade, L. Mineh, A. Montanaro, and S. Stanisic. “Strategies for solving the Fermi-Hubbard model on near-term quantum computers”. *Phys. Rev. B* **102**, 235122. 2020.  
DOI: 10.1103/PhysRevB.102.235122.

- [Cal+21] A. Callison, M. Festenstein, J. Chen, L. Nita, V. Kendon, and N. Chancellor. “Energetic Perspective on Rapid Quenches in Quantum Annealing”. *PRX Quantum* **2**, 010338. 2021.  
DOI: 10.1103/prxquantum.2.010338.
- [Cao+19] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik. “Quantum Chemistry in the Age of Quantum Computing”. *Chemical Reviews* **119**, 10856–10915. 2019.  
DOI: 10.1021/acs.chemrev.8b00803.
- [Car+20] J. Carolan, M. Mohseni, J. P. Olson, M. Prabhu, C. Chen, D. Bunandar, M. Y. Niu, N. C. Harris, F. N. C. Wong, M. Hochberg, S. Lloyd, and D. Englund. “Variational quantum unsampling on a quantum photonic processor”. *Nature Physics* **16**, 322–327. 2020.  
DOI: 10.1038/s41567-019-0747-6.
- [Car65] L. Carroll. *Alice’s Adventures in Wonderland*. Macmillan. 1865.
- [CBG21] A. Cornelissen, J. Bausch, and A. Gilyén. “Scalable Benchmarks for Gate-Based Quantum Computers”. *arXiv:2104.10698*. 2021.  
DOI: 10.48550/ARXIV.2104.10698.
- [CD00] J. I. Casaubon and G. Doggett. “Variational Principle for a Particle in a Box”. *Journal of Chemical Education* **77**, 1221. 2000.  
DOI: 10.1021/ed077p1221.
- [CEM84] W. J. Caspers, K. M. Emmett, and W. Magnus. “The Majumdar-Ghosh chain. Twofold ground state and elementary excitations”. *Journal of Physics A: Mathematical and General* **17**, 2687–2696. 1984.  
DOI: 10.1088/0305-4470/17/13/020.
- [Cer+21] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles. “Variational quantum algorithms”. *Nature Reviews Physics* **3**, 625–644. 2021.  
DOI: 10.1038/s42254-021-00348-9.
- [Cha19] N. Chancellor. “Domain wall encoding of discrete variables for quantum annealing and QAOA”. *Quantum Science and Technology* **4**, 045004. 2019.  
DOI: 10.1088/2058-9565/ab33c2.
- [Chi+20] D. Chivilikhin, A. Samarin, V. Ulyantsev, I. Iorsh, A. R. Oganov, and O. Kyriienko. “MoG-VQE: Multiobjective genetic variational quantum eigensolver”. *arXiv:2007.04424*. 2020.  
DOI: 10.48550/ARXIV.2007.04424.
- [CL95] P. M. Chaikin and T. C. Lubensky. *Principles of Condensed Matter Physics*. Cambridge University Press. 1995.  
DOI: 10.1017/CBO9780511813467.
- [Cla+20] D. Claudino, J. Wright, A. J. McCaskey, and T. S. Humble. “Benchmarking Adaptive Variational Quantum Eigensolvers”. *Frontiers in Chemistry* **8**, 606863. 2020.  
DOI: 10.3389/fchem.2020.606863.
- [CLLG19] A. Cervera-Lierta, J. I. Latorre, and D. Goyeneche. “Quantum circuits for maximally entangled states”. *Phys. Rev. A* **100**, 022342. 2019.  
DOI: 10.1103/PhysRevA.100.022342.

- [CNB21] E. Campos, A. Nasrallah, and J. Biamonte. “Abrupt transitions in variational quantum circuit training”. *Phys. Rev. A* **103**, 032607. 2021.  
DOI: 10.1103/PhysRevA.103.032607.
- [Col+18] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi. “Computation of Molecular Spectra on a Quantum Processor with an Error-Resilient Algorithm”. *Phys. Rev. X* **8**, 011021. 2018.  
DOI: 10.1103/PhysRevX.8.011021.
- [Con+22] M. Consiglio, W. J. Chetcuti, C. Bravo-Prieto, S. Ramos-Calderer, A. Minguzzi, J. I. Latorre, L. Amico, and T. J. G. Apollaro. “Variational quantum eigensolver for SU(N) fermions”. *Journal of Physics A: Mathematical and Theoretical* **55**, 265301. 2022.  
DOI: 10.1088/1751-8121/ac7016.
- [Cro+17] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta. “Open Quantum Assembly Language”. *arXiv:1707.03429*. 2017.  
DOI: 10.48550/ARXIV.1707.03429.
- [Cro19] G. E. Crooks. “Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition”. *arXiv:1905.13311*. 2019.  
DOI: 10.48550/ARXIV.1905.13311.
- [Cro+19] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta. “Validating quantum computers using randomized model circuits”. *Phys. Rev. A* **100**, 032328. 2019.  
DOI: 10.1103/PhysRevA.100.032328.
- [CW02] J. K. Cullum and R. A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*. Society for Industrial and Applied Mathematics. 2002.  
DOI: 10.1137/1.9780898719192.
- [Dav75] E. R. Davidson. “The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices”. *Journal of Computational Physics* **17**, 87–94. 1975.  
DOI: 10.1016/0021-9991(75)90065-0.
- [DBE95] D. Deutsch, A. Barenco, and A. Ekert. “Universality in Quantum Computation”. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* **449**, 669–677. 1995.  
DOI: 10.1098/rspa.1995.0065.
- [DDL84] H. De Raedt, B. De Raedt, and A. Legendijk. “Thermodynamics of the two-dimensional spin-1/2 XY model”. *Zeitschrift für Physik B Condensed Matter* **57**, 209–220. 1984.  
DOI: 10.1007/BF01318413.
- [De +07] K. De Raedt, K. Michielsen, H. De Raedt, B. Trieu, G. Arnold, M. Richter, Th. Lipfert, H. Watanabe, and N. Ito. “Massively parallel quantum computer simulator”. *Computer Physics Communications* **176**, 121–136. 2007.  
DOI: 10.1016/j.cpc.2006.08.007.
- [De +19] H. De Raedt, F. Jin, D. Willsch, M. Willsch, N. Yoshioka, N. Ito, S. Yuan, and K. Michielsen. “Massively parallel quantum computer simulator, eleven years later”. *Computer Physics Communications* **237**, 47–61. 2019.  
DOI: 10.1016/j.cpc.2018.11.005.

- [De +20] H. De Raedt, M. S. Jattana, D. Willsch, M. Willsch, F. Jin, and K. Michielsen. “Discrete-Event Simulation of an Extended Einstein-Podolsky-Rosen-Bohm Experiment”. *Frontiers in Physics* **8**, 160. 2020.  
DOI: 10.3389/fphy.2020.00160.
- [De 93] H. De Raedt. “Stochastic diagonalization”. *Physics Reports* **231**, 107–149. 1993.  
DOI: 10.1016/0370-1573(93)90015-6.
- [D’E+12] F. D’Errico, L. Backwell, P. Villa, I. Degano, J. J. Lucejko, M. K. Bamford, T. F. Higham, M. P. Colombini, and P. B. Beaumont. “Early evidence of San material culture represented by organic artifacts from Border Cave, South Africa”. *Proceedings of the National Academy of Sciences of the United States of America* **109**, 13214–13219. 2012.  
DOI: 10.1073/pnas.1204213109.
- [Del+22] Y. Deller, S. Schmitt, M. Lewenstein, S. Lenk, M. Federer, F. Jendrzejewski, P. Hauke, and V. Kasper. “Quantum approximate optimization algorithm for qudit systems with long-range interactions”. *arXiv:2204.00340*. 2022.  
DOI: 10.48550/ARXIV.2204.00340.
- [Den+20] Z. Deng, I. Tutunnikov, I. S. Averbukh, M. Thachuk, and R. V. Krems. “Bayesian optimization for inverse problems in time-dependent quantum dynamics”. *The Journal of Chemical Physics* **153**, 164111. 2020.  
DOI: 10.1063/5.0015896.
- [DiV+00] D. P. DiVincenzo, D. Bacon, J. Kempe, G. Burkard, and K. B. Whaley. “Universal quantum computation with the exchange interaction”. *Nature* **408**, 339–342. 2000.  
DOI: 10.1038/35042541.
- [DiV00] D. P. DiVincenzo. “The Physical Implementation of Quantum Computation”. *Fortschritte der Physik* **48**, 771783. 2000.  
DOI: 10.1002/1521-3978(200009)48:9/11<771::aid-prop771>3.0.co;2-e.
- [DiV09] D. P. DiVincenzo. “Fault-tolerant architectures for superconducting qubits”. *Physica Scripta* **T137**, 014020. 2009.  
DOI: 10.1088/0031-8949/2009/t137/014020.
- [DiV95] D. P. DiVincenzo. “Two-bit gates are universal for quantum computation”. *Phys. Rev. A* **51**, 1015–1022. 1995.  
DOI: 10.1103/PhysRevA.51.1015.
- [DM98] L. Dagum and R. Menon. “OpenMP: an industry standard API for shared-memory programming”. *IEEE Computational Science and Engineering* **5**, 46–55. 1998.  
DOI: 10.1109/99.660313.
- [DR87] H. De Raedt. “Product formula algorithms for solving the time dependent Schrödinger equation”. *Computer Physics Reports* **7**, 1–72. 1987.  
DOI: 10.1016/0167-7977(87)90002-5.
- [DS96] D. P. DiVincenzo and P. W. Shor. “Fault-Tolerant Error Correction with Efficient Quantum Codes”. *Phys. Rev. Lett.* **77**, 3260–3263. 1996.  
DOI: 10.1103/PhysRevLett.77.3260.

- [Eba+22] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuleti, and M. D. Lukin. “Quantum optimization of maximum independent set using Rydberg atom arrays”. *Science* **376**, 1209–1215. 2022.  
DOI: 10.1126/science.abo6587.
- [EJ96] A. Ekert and R. Jozsa. “Quantum computation and Shor’s factoring algorithm”. *Reviews of Modern Physics* **68**, 733–753. 1996.  
DOI: 10.1103/RevModPhys.68.733.
- [Elf+20] V. E. Elfving, B. W. Broer, M. Webber, J. Gavartin, M. D. Halls, K. P. Lorton, and A. Bochevarov. “How will quantum computers provide an industrially relevant computational advantage in quantum chemistry?” *arXiv:2009.12472*. 2020.  
DOI: 10.48550/ARXIV.2009.12472.
- [EN07] S. Ejima and S. Nishimoto. “Phase Diagram of the One-Dimensional Half-Filled Extended Hubbard Model”. *Phys. Rev. Lett.* **99**, 216403. 2007.  
DOI: 10.1103/PhysRevLett.99.216403.
- [Far+00] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. “Quantum Computation by Adiabatic Evolution”. *arXiv:quant-ph/0001106*. 2000.  
DOI: 10.48550/ARXIV.QUANT-PH/0001106.
- [Far+01] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda. “A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem”. *Science* **292**, 472–475. 2001.  
DOI: 10.1126/science.1057726.
- [Fen] Ground state energies using the Lanczos algorithm provided by Fengping Jin using in-house software.
- [Fey82] R. P. Feynman. “Simulating physics with computers”. *International Journal of Theoretical Physics* **21**, 467–488. 1982.  
DOI: 10.1007/BF02650179.
- [FGG14] E. Farhi, J. Goldstone, and S. Gutmann. “A Quantum Approximate Optimization Algorithm”. *arXiv:1411.4028*. 2014.  
DOI: 10.48550/ARXIV.1411.4028.
- [Fin+94] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll. “Quantum annealing: A new method for minimizing multidimensional functions”. *Chemical Physics Letters* **219**, 343–348. 1994.  
DOI: 10.1016/0009-2614(94)00117-0.
- [FN18] E. Farhi and H. Neven. “Classification with Quantum Neural Networks on Near Term Processors”. *arXiv:1802.06002*. 2018.  
DOI: 10.48550/ARXIV.1802.06002.
- [For94] M. P. Forum. MPI: A Message-Passing Interface Standard. Tech. rep. USA. 1994.
- [Fri+16] M. J. Frisch et al. Gaussian 16 Rev. C.01. Wallingford, CT. 2016.
- [Fö+21] T. Fösel, M. Y. Niu, F. Marquardt, and L. Li. “Quantum circuit optimization with deep reinforcement learning”. *arXiv:2103.07585*. 2021.  
DOI: 10.48550/ARXIV.2103.07585.

- [Gar+20] B. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes. “Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm”. *npj Quantum Information* **6**, 10. 2020.  
DOI: 10.1038/s41534-019-0240-1.
- [GCS17] J. M. Gambetta, J. M. Chow, and M. Steffen. “Building logical qubits in a superconducting quantum computing system”. *npj Quantum Information* **3**, 2. 2017.  
DOI: 10.1038/s41534-016-0004-0.
- [Gel20] M. R. Geller. “Rigorous measurement error correction”. *Quantum Science and Technology* **5**, 03LT01. 2020.  
DOI: 10.1088/2058-9565/ab9591.
- [GF19] F. Gerber and R. Furrer. “optimParallel: An R Package Providing a Parallel Version of the L-BFGS-B Optimization Method”. *The R Journal* **11**, 352–358. 2019.  
DOI: 10.32614/RJ-2019-030.
- [Gho+21] S. Ghosh, H. Mishra, B. K. Behera, and P. K. Panigrahi. “Experimental realization of BB84 protocol with different phase gates and SARG04 protocol”. *arXiv:2110.00308*. 2021.  
DOI: 10.48550/ARXIV.2110.00308.
- [Gil60] W Gilbert. *De Magnete, Magneticisque Corporibus, et de Magno Magnete Tellure* (Translated by P. Fleury Mottelay). 1660.
- [Gok+20] P. Gokhale, O. Angiuli, Y. Ding, K. Gui, T. Tomesh, M. Suchara, M. Martonosi, and F. T. Chong. “ $O(N^3)$  Measurement Cost for Variational Quantum Eigensolver on Molecular Hamiltonians”. *IEEE Transactions on Quantum Engineering* **1**, 1–24. 2020.  
DOI: 10.1109/TQE.2020.3035814.
- [Goo77] J. Goodisman. *Contemporary Quantum Chemistry*. Springer US. 1977.  
DOI: 10.1007/978-1-4684-2268-9.
- [Got98] D. Gottesman. “Theory of fault-tolerant quantum computation”. *Phys. Rev. A* **57**, 127–137. 1998.  
DOI: 10.1103/PhysRevA.57.127.
- [Gra+19] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti. “An initialization strategy for addressing barren plateaus in parametrized quantum circuits”. *Quantum* **3**, 214. 2019.  
DOI: 10.22331/q-2019-12-09-214.
- [Gre15] D. Greenbaum. “Introduction to Quantum Gate Set Tomography”. *arXiv:1509.02921*. 2015.  
DOI: 10.48550/ARXIV.1509.02921.
- [Gri+19] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall. “An adaptive variational algorithm for exact molecular simulations on a quantum computer”. *Nature Communications* **10**, 3007. 2019.  
DOI: 10.1038/s41467-019-10988-2.
- [Gri+20] H. R. Grimsley, D. Claudino, S. E. Economou, E. Barnes, and N. J. Mayhall. “Is the Trotterized UCCSD Ansatz Chemically Well-Defined?” *Journal of Chemical Theory and Computation* **16**, 1–6. 2020.  
DOI: 10.1021/acs.jctc.9b01083.

- [Gro96] L. K. Grover. “A Fast Quantum Mechanical Algorithm for Database Search”. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 212219. 1996.  
DOI: 10.1145/237814.237866.
- [Gut63] M. C. Gutzwiller. “Effect of Correlation on the Ferromagnetism of Transition Metals”. *Phys. Rev. Lett.* **10**, 159–162. 1963.  
DOI: 10.1103/PhysRevLett.10.159.
- [HA18] L. M. L. Hilario and E. A. Arroyo. “Improved variational method that solves the energy eigenvalue problem of the hydrogen atom”. *Journal of Physics: Conference Series* **1143**, 012031. 2018.  
DOI: 10.1088/1742-6596/1143/1/012031.
- [Had+22] C. Hadfield, S. Bravyi, R. Raymond, and A. Mezzacapo. “Measurements of Quantum Hamiltonians with Locally-Biased Classical Shadows”. *Communications in Mathematical Physics* **391**, 951–967. 2022.  
DOI: 10.1007/s00220-022-04343-8.
- [Han] Data provided by Hans De Raedt using in-house software.
- [Har+10] R. Harris, M. W. Johnson, T. Lanting, A. J. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, F. Cioata, I. Perminov, P. Spear, C. Enderud, C. Rich, S. Uchaikin, M. C. Thom, E. M. Chapple, J. Wang, B. Wilson, M. H. S. Amin, N. Dickson, K. Karimi, B. Macready, C. J. S. Truncik, and G. Rose. “Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor”. *Phys. Rev. B* **82**, 024511. 2010.  
DOI: 10.1103/PhysRevB.82.024511.
- [Har+18] R. Harris et al. “Phase transitions in a programmable quantum spin glass simulator”. *Science* **361**, 162–165. 2018.  
DOI: 10.1126/science.aat2025.
- [Hav+19] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta. “Supervised learning with quantum-enhanced feature spaces”. *Nature* **567**, 209–212. 2019.  
DOI: 10.1038/s41586-019-0980-2.
- [HBR21] H.-Y. Huang, K. Bharti, and P. Rebentrost. “Near-term quantum algorithms for linear systems of equations with regression loss functions”. *New Journal of Physics* **23**, 113021. 2021.  
DOI: 10.1088/1367-2630/ac325f.
- [HC18] L. E. Heyfron and E. T. Campbell. “An efficient quantum compiler that reduces T count”. *Quantum Science and Technology* **4**, 015004. 2018.  
DOI: 10.1088/2058-9565/aad604.
- [HD00] A. Hams and H. De Raedt. “Fast algorithm for finding the eigenvalue distribution of very large matrices”. *Phys. Rev. E* **62**, 4365–4377. 2000.  
DOI: 10.1103/PhysRevE.62.4365.
- [Hem+18] C. Hempel, C. Maier, J. Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. P. Lanyon, P. Love, R. Babbush, A. Aspuru-Guzik, R. Blatt, and C. F. Roos. “Quantum Chemistry Calculations on a Trapped-Ion Quantum Simulator”. *Phys. Rev. X* **8**, 031022. 2018.  
DOI: 10.1103/PhysRevX.8.031022.

- [HHH96] M. Horodecki, P. Horodecki, and R. Horodecki. “Separability of mixed states: necessary and sufficient conditions”. *Physics Letters A* **223**, 1–8. 1996.  
DOI: 10.1016/S0375-9601(96)00706-2.
- [HHL09] A. W. Harrow, A. Hassidim, and S. Lloyd. “Quantum Algorithm for Linear Systems of Equations”. *Phys. Rev. Lett.* **103**, 150502. 2009.  
DOI: 10.1103/PhysRevLett.103.150502.
- [HKP20] H.-Y. Huang, R. Kueng, and J. Preskill. “Predicting many properties of a quantum system from very few measurements”. *Nature Physics* **16**, 1050–1057. 2020.  
DOI: 10.1038/s41567-020-0932-7.
- [HMK03] N. Hansen, S. D. Müller, and P. Koumoutsakos. “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES).” *Evolutionary computation* **11**, 1–18. 2003.  
DOI: 10.1162/106365603321828970.
- [Hol+22] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles. “Connecting Ansatz Expressibility to Gradient Magnitudes and Barren Plateaus”. *PRX Quantum* **3**, 010313. 2022.  
DOI: 10.1103/PRXQuantum.3.010313.
- [Hov+14] D. Hover, S. Zhu, T. Thorbeck, G. J. Ribeill, D. Sank, J. Kelly, R. Barends, J. M. Martinis, and R. McDermott. “High fidelity qubit readout with the superconducting low-inductance undulatory galvanometer microwave amplifier”. *Applied Physics Letters* **104**, 152601. 2014.  
DOI: 10.1063/1.4871088.
- [HSP69] W. J. Hehre, R. E. Stewart, and J. A. Pople. “Self-Consistent Molecular-Orbital Methods. I. Use of Gaussian Expansions of Slater-Type Atomic Orbitals”. *The Journal of Chemical Physics* **51**, 2657–2664. 1969.  
DOI: 10.1063/1.1672392.
- [Hsu+19] T. J. Hsu, F. Jin, C. Seidel, F. Neukart, H. De Raedt, and K. Michielsen. “Quantum annealing with anneal path control: Application to 2-SAT problems with known energy landscapes”. *Communications in Computational Physics* **26**, 928–946. 2019.  
DOI: 10.4208/cicp.OA-2018-0257.
- [HTW17] V. Havlicek, M. Troyer, and J. D. Whitfield. “Operator locality in the quantum simulation of fermionic models”. *Phys. Rev. A* **95**, 032332. 2017.  
DOI: 10.1103/PhysRevA.95.032332.
- [Hub63] J. Hubbard. “Electron correlations in narrow energy bands”. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **276**, 238–257. 1963.  
DOI: 10.1098/rspa.1963.0204.
- [Hub64a] J. Hubbard. “Electron correlations in narrow energy bands. II. The degenerate band case”. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **277**, 237–259. 1964.  
DOI: 10.1098/rspa.1964.0019.
- [Hub64b] J. Hubbard. “Electron correlations in narrow energy bands III. An improved solution”. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **281**, 401–419. 1964.  
DOI: 10.1098/rspa.1964.0190.

- [Hug+19] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire. “Towards quantum machine learning with tensor networks”. *Quantum Science and Technology* **4**, 024001. 2019.  
DOI: 10.1088/2058-9565/aaea94.
- [Hug+21] W. J. Huggins, J. R. McClean, N. C. Rubin, Z. Jiang, N. Wiebe, K. B. Whaley, and R. Babbush. “Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers”. *npj Quantum Information* **7**, 23. 2021.  
DOI: 10.1038/s41534-020-00341-7.
- [HUL93a] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Vol. 305. Grundlehren der mathematischen Wissenschaften. Berlin, Heidelberg: Springer Berlin Heidelberg. 1993.  
DOI: 10.1007/978-3-662-02796-7.
- [HUL93b] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. Vol. 306. Grundlehren der mathematischen Wissenschaften. Berlin, Heidelberg: Springer Berlin Heidelberg. 1993.  
DOI: 10.1007/978-3-662-06409-2.
- [HWB19] O. Higgott, D. Wang, and S. Brierley. “Variational Quantum Computation of Excited States”. *Quantum* **3**, 156. 2019.  
DOI: 10.22331/q-2019-07-01-156.
- [IH01] G. Ifrah and E. F. Harding. *The Universal History of Computing: From the Abacus to the Quantum Computer*. 1st. USA: John Wiley & Sons, Inc. 2001.
- [Izm+20] A. F. Izmaylov, T.-C. Yen, R. A. Lang, and V. Verteletskiy. “Unitary Partitioning Approach to the Measurement Problem in the Variational Quantum Eigensolver Method”. *Journal of Chemical Theory and Computation* **16**, 190–195. 2020.  
DOI: 10.1021/acs.jctc.9b00791.
- [Jac46] C. G. Jacobi. “Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen.” *Journal für die reine und angewandte Mathematik* **1846**, 51–94. 1846.  
DOI: 10.1515/crll.1846.30.51.
- [Jat19] M. S. Jattana. “Simulating Extended Einstein-Podolsky-Rosen-Bohm (EPRB) Experiment”. Masterarbeit. RWTH Aachen. 2019.
- [Jat+20] M. S. Jattana, F. Jin, H. De Raedt, and K. Michielsen. “General error mitigation for quantum circuits”. *Quantum Information Processing* **19**, 414. 2020.  
DOI: 10.1007/s11128-020-02913-0.
- [Jat+22a] M. S. Jattana, F. Jin, H. De Raedt, and K. Michielsen. “Assessment of the Variational Quantum Eigensolver: Application to the Heisenberg Model”. *Frontiers in Physics* **10**, 907160. 2022.  
DOI: 10.3389/fphy.2022.907160.
- [Jat+22b] M. S. Jattana, F. Jin, H. De Raedt, and K. Michielsen. “Improved variational quantum eigensolver via quasi-dynamical evolution”. *arXiv:2202.10130*. 2022.  
DOI: 10.48550/ARXIV.2202.10130.
- [Jay03] E. T. Jaynes. *Probability Theory: The Logic of Science*. Ed. by G. L. Bretthorst. Cambridge University Press. 2003.  
DOI: 10.1017/CBO9780511790423.

- [JB22] T. Jones and S. C. Benjamin. “Robust quantum compilation and circuit optimisation via energy minimisation”. *Quantum* **6**, 628. 2022.  
DOI: 10.22331/q-2022-01-24-628.
- [JGM19] A. Jena, S. Genin, and M. Mosca. “Pauli Partitioning with Respect to Gate Sets”. *arXiv:1907.07859*. 2019.  
DOI: 10.48550/ARXIV.1907.07859.
- [Jia+18] Z. Jiang, K. J. Sung, K. Kechedzhi, V. N. Smelyanskiy, and S. Boixo. “Quantum Algorithms to Simulate Many-Body Physics of Correlated Fermions”. *Phys. Rev. Applied* **9**, 044036. 2018.  
DOI: 10.1103/PhysRevApplied.9.044036.
- [JLM22] M. S. Jattana, H. Lagemann, and K. Michielsen. Final report on HPC simulation of QC: Conclusions for quantum computing without error correction. OpenSuperQ — An Open Superconducting Quantum Computer (Grant 820363). 2022.  
DOI: 10.3030/820363.
- [JM19] M. S. Jattana and K. Michielsen. “Quantencomputer - Steht die Revolution vor der Türe?” *Technik in Bayern* **06/2019**, 18–19. 2019.
- [Joh+11] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose. “Quantum annealing with manufactured spins”. *Nature* **473**, 194–198. 2011.  
DOI: 10.1038/nature10012.
- [Jon+19] T. Jones, S. Endo, S. McArdle, X. Yuan, and S. C. Benjamin. “Variational quantum algorithms for discovering Hamiltonian spectra”. *Phys. Rev. A* **99**, 062304. 2019.  
DOI: 10.1103/PhysRevA.99.062304.
- [JOP+01] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python. 2001.
- [Jur] Jülich Supercomputing Centre. *JURECA: Data Centric and Booster Modules implementing the Modular Supercomputing Architecture at Jülich Supercomputing Centre*. Journal of large-scale research facilities 7, A182 (2021).
- [Juw] Jülich Supercomputing Centre. *JUWELS: Modular Tier-0/1 Supercomputer at the Jülich Supercomputing Centre*. Journal of large-scale research facilities 5, A135 (2019).
- [JW28] P. Jordan and E. Wigner. “Über das Paulische Äquivalenzverbot”. *Zeitschrift für Physik* **47**, 631–651. 1928.  
DOI: 10.1007/BF01331938.
- [Kal62] M. H. Kalos. “Monte Carlo Calculations of the Ground State of Three- and Four-Body Nuclei”. *Phys. Rev.* **128**, 1791–1795. 1962.  
DOI: 10.1103/PhysRev.128.1791.
- [Kan+17] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta. “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets”. *Nature* **549**, 242–246. 2017.  
DOI: 10.1038/nature23879.
- [Kan63] J. Kanamori. “Electron Correlation and Ferromagnetism of Transition Metals”. *Progress of Theoretical Physics* **30**, 275–289. 1963.  
DOI: 10.1143/ptp.30.275.

- [Kar+20] P. J. Karalekas, N. A. Tezak, E. C. Peterson, C. A. Ryan, M. P. da Silva, and R. S. Smith. “A quantum-classical cloud platform optimized for variational hybrid algorithms”. *Quantum Science and Technology* **5**, 024003. 2020.  
DOI: 10.1088/2058-9565/ab7559.
- [Kar+21] A. H. Karamlou, W. A. Simon, A. Katabarwa, T. L. Scholten, B. Peropadre, and Y. Cao. “Analyzing the performance of variational quantum factoring on a superconducting quantum processor”. *npj Quantum Information* **7**, 156. 2021.  
DOI: 10.1038/s41534-021-00478-z.
- [Kat50] T. Kato. “On the Adiabatic Theorem of Quantum Mechanics”. *Journal of the Physical Society of Japan* **5**, 435–439. 1950.  
DOI: 10.1143/JPSJ.5.435.
- [KB22] B. Koczor and S. C. Benjamin. “Quantum analytic descent”. *Phys. Rev. Research* **4**, 023017. 2022.  
DOI: 10.1103/PhysRevResearch.4.023017.
- [Kel11] C. T. Kelley. *Implicit Filtering*. Society for Industrial and Applied Mathematics. 2011.  
DOI: 10.1137/1.9781611971903.
- [Kha+19] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles. “Quantum-assisted quantum compiling”. *Quantum* **3**, 140. 2019.  
DOI: 10.22331/q-2019-05-13-140.
- [Kin+18] A. D. King, J. Carrasquilla, J. Raymond, I. Ozfidan, E. Andriyash, A. Berkley, M. Reis, T. Lanting, R. Harris, F. Altomare, K. Boothby, P. I. Bunyk, C. Enderud, A. Fréchet, E. Hoskinson, N. Ladizinsky, T. Oh, G. Poulin-Lamarre, C. Rich, Y. Sato, A. Y. Smirnov, L. J. Swenson, M. H. Volkmann, J. Whittaker, J. Yao, E. Ladizinsky, M. W. Johnson, J. Hilton, and M. H. Amin. “Observation of topological phenomena in a programmable lattice of 1,800 qubits”. *Nature* **560**, 456–460. 2018.  
DOI: 10.1038/s41586-018-0410-x.
- [Kin+21] A. D. King, C. Nisoli, E. D. Dahl, G. Poulin-Lamarre, and A. Lopez-Bezanilla. “Qubit spin ice”. *Science* **373**. 2021.  
DOI: 10.1126/science.abe2824.
- [Kin+22a] A. D. King, J. Raymond, T. Lanting, R. Harris, A. Zucca, F. Altomare, A. J. Berkley, K. Boothby, S. Ejtemaee, C. Enderud, E. Hoskinson, S. Huang, E. Ladizinsky, A. J. R. MacDonald, G. Marsden, R. Molavi, T. Oh, G. Poulin-Lamarre, M. Reis, C. Rich, Y. Sato, N. Tsai, M. Volkmann, J. D. Whittaker, J. Yao, A. W. Sandvik, and M. H. Amin. “Quantum critical dynamics in a 5000-qubit programmable spin glass”. *arXiv:2207.13800*. 2022.  
DOI: 10.48550/ARXIV.2207.13800.
- [Kin+22b] A. D. King, S. Suzuki, J. Raymond, A. Zucca, T. Lanting, F. Altomare, A. J. Berkley, S. Ejtemaee, E. Hoskinson, S. Huang, E. Ladizinsky, A. MacDonald, G. Marsden, T. Oh, G. Poulin-Lamarre, M. Reis, C. Rich, Y. Sato, J. D. Whittaker, J. Yao, R. Harris, D. A. Lidar, H. Nishimori, and M. H. Amin. “Coherent quantum annealing in a programmable 2000-qubit Ising chain”. *arXiv:2202.05847*. 2022.  
DOI: 10.48550/ARXIV.2202.05847.
- [Kir+22] W. Kirby, B. Fuller, C. Hadfield, and A. Mezzacapo. “Second-Quantized Fermionic Operators with Polylogarithmic Qubit and Gate Complexity”. *PRX Quantum* **3**, 020351. 2022.  
DOI: 10.1103/PRXQuantum.3.020351.

- [KN98] T. Kadowaki and H. Nishimori. “Quantum annealing in the transverse Ising model”. *Phys. Rev. E* **58**, 5355–5363. 1998.  
DOI: 10.1103/PhysRevE.58.5355.
- [Koc+07] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. “Charge-insensitive qubit design derived from the Cooper pair box”. *Phys. Rev. A* **76**, 042319. 2007.  
DOI: 10.1103/PhysRevA.76.042319.
- [Koh99] W. Kohn. “Nobel Lecture: Electronic structure of matter—wave functions and density functionals”. *Rev. Mod. Phys.* **71**, 1253–1266. 1999.  
DOI: 10.1103/RevModPhys.71.1253.
- [KPE21] O. Kyriienko, A. E. Paine, and V. E. Elfving. “Solving nonlinear differential equations with differentiable quantum circuits”. *Phys. Rev. A* **103**, 052416. 2021.  
DOI: 10.1103/PhysRevA.103.052416.
- [Kra88] D. Kraft. A software package for sequential quadratic programming. DLR German Aerospace Center – Institute for Flight Mechanics. 1988.
- [Kus+21] T. Kusumoto, K. Mitarai, K. Fujii, M. Kitagawa, and M. Negoro. “Experimental quantum kernel trick with nuclear spins in a solid”. *npj Quantum Information* **7**, 94. 2021.  
DOI: 10.1038/s41534-021-00423-0.
- [KW20] A. Kissinger and J. van de Wetering. “Reducing the number of non-Clifford gates in quantum circuits”. *Phys. Rev. A* **102**, 022406. 2020.  
DOI: 10.1103/PhysRevA.102.022406.
- [KW21] J. Kattemölle and J. van Wezel. “Variational quantum eigensolver for the Heisenberg antiferromagnet on the kagome lattice”. *arXiv:2108.02175*. 2021.  
DOI: 10.48550/ARXIV.2108.02175.
- [Lam92] S. K. Lamoreaux. “A review of the experimental tests of quantum mechanics”. *International Journal of Modern Physics A* **07**, 6691–6762. 1992.  
DOI: 10.1142/S0217751X92003082.
- [Lar+21] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo. “Theory of overparametrization in quantum neural networks”. *arXiv:2109.11676*. 2021.  
DOI: 10.48550/ARXIV.2109.11676.
- [Lav+20] W. Lavrijsen, A. Tudor, J. Müller, C. Iancu, and W. de Jong. “Classical Optimizers for Noisy Intermediate-Scale Quantum Devices”. *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 267–277. 2020.  
DOI: 10.1109/QCE49297.2020.00041.
- [LB17] Y. Li and S. C. Benjamin. “Efficient Variational Quantum Simulator Incorporating Active Error Minimization”. *Phys. Rev. X* **7**, 021050. 2017.  
DOI: 10.1103/PhysRevX.7.021050.
- [LD98] D. Loss and D. P. DiVincenzo. “Quantum computation with quantum dots”. *Phys. Rev. A* **57**, 120–126. 1998.  
DOI: 10.1103/PhysRevA.57.120.
- [Lee+21] J. Lee, D. W. Berry, C. Gidney, W. J. Huggins, J. R. McClean, N. Wiebe, and R. Babbush. “Even More Efficient Quantum Computations of Chemistry Through Tensor Hypercontraction”. *PRX Quantum* **2**, 030305. 2021.  
DOI: 10.1103/PRXQuantum.2.030305.

- [LH87] H. Q. Lin and J. E. Hirsch. “Two-dimensional Hubbard model with nearest- and next-nearest-neighbor hopping”. *Phys. Rev. B* **35**, 3359–3368. 1987.  
DOI: 10.1103/PhysRevB.35.3359.
- [Lin+17] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe. “Experimental comparison of two quantum computing architectures”. *Proceedings of the National Academy of Sciences* **114**, 3305–3310. 2017.  
DOI: 10.1073/pnas.1618020114.
- [Liu+19] J.-G. Liu, Y.-H. Zhang, Y. Wan, and L. Wang. “Variational quantum eigensolver with fewer qubits”. *Phys. Rev. Research* **1**, 023025. 2019.  
DOI: 10.1103/PhysRevResearch.1.023025.
- [Liu+22] X. Liu, A. Angone, R. Shaydulin, I. Safro, Y. Alexeev, and L. Cincio. “Layer VQE: A Variational Approach for Combinatorial Optimization on Noisy Quantum Computers”. *IEEE Transactions on Quantum Engineering* **3**, 1–20. 2022.  
DOI: 10.1109/TQE.2021.3140190.
- [LMW19] J. Larson, M. Menickelly, and S. M. Wild. “Derivative-free optimization methods”. *Acta Numerica* **28**, 287404. 2019.  
DOI: 10.1017/S0962492919000060.
- [Lom03] C. Lomont. “Quantum Circuit Identities”. *arXiv:quant-ph/0307111*. 2003.  
DOI: 10.48550/ARXIV.QUANT-PH/0307111.
- [Lub+20] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch. “Variational quantum algorithms for nonlinear problems”. *Phys. Rev. A* **101**, 010301. 2020.  
DOI: 10.1103/PhysRevA.101.010301.
- [Lud09] P. E. Ludgate. “On a Proposed Analytical Machine”. *The Origins of Digital Computers: Selected Papers*. Ed. by B. Randell. Springer Berlin Heidelberg, 73–87. 1909.  
DOI: 10.1007/978-3-642-61812-3\_5.
- [Luo07] Y.-R. Luo. *Comprehensive Handbook of Chemical Bond Energies* (1st ed.) CRC Press. 2007.
- [Mai07] A. K. Maini. *Digital electronics : principles, devices and applications*. John Wiley and Sons. 2007.
- [Maj70] C. K. Majumdar. “Antiferromagnetic model with known ground state”. *Journal of Physics C: Solid State Physics* **3**, 911–915. 1970.  
DOI: 10.1088/0022-3719/3/4/019.
- [Mar+21] M. Maronese, L. Moro, L. Rocutto, and E. Prati. “Quantum Compiling”. *arXiv:2112.00187*. 2021.  
DOI: 10.48550/ARXIV.2112.00187.
- [Mas+08] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negrevergne. “Quantum Circuit Simplification and Level Compaction”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **27**, 436–444. 2008.  
DOI: 10.1109/TCAD.2007.911334.
- [Mat06] D. C. Mattis. *The Theory of Magnetism Made Simple*. World Scientific. 2006.  
DOI: 10.1142/5372.
- [MB00] S. Malin and D. Barraclough. “Gilbert’s De Magnete: An early study of magnetism and electricity”. *Eos, Transactions American Geophysical Union* **81**, 233–234. 2000.  
DOI: 10.1029/00EO00163.

- [McC+16] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. “The theory of variational hybrid quantum-classical algorithms”. *New Journal of Physics* **18**, 23023. 2016.  
DOI: 10.1088/1367-2630/18/2/023023.
- [McC+17a] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong. “Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states”. *Phys. Rev. A* **95**, 042308. 2017.  
DOI: 10.1103/PhysRevA.95.042308.
- [McC+17b] J. R. McClean et al. “OpenFermion: The Electronic Structure Package for Quantum Computers”. *arXiv:1710.07629*. 2017.  
DOI: 10.48550/ARXIV.1710.07629.
- [McC+18] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven. “Barren plateaus in quantum neural network training landscapes”. *Nature Communications* **9**, 1–6. 2018.  
DOI: 10.1038/s41467-018-07090-4.
- [McC+19] A. J. McCaskey, Z. P. Parks, J. Jakowski, S. V. Moore, T. D. Morris, T. S. Humble, and R. C. Pooser. “Quantum chemistry as a benchmark for near-term quantum computers”. *npj Quantum Information* **5**, 99. 2019.  
DOI: 10.1038/s41534-019-0209-0.
- [Mei96] W. N. Mei. “Demonstration of systematic improvements in the application of the variational method to the harmonic oscillator”. *International Journal of Mathematical Education in Science and Technology* **27**, 285–292. 1996.  
DOI: 10.1080/0020739960270215.
- [Mic+17] K. Michielsen, M. Nocon, D. Willsch, F. Jin, Th. Lippert, and H. De Raedt. “Benchmarking gate-based quantum computers”. *Computer Physics Communications* **220**, 44–55. 2017.  
DOI: 10.1016/j.cpc.2017.06.011.
- [Mic+19] K. Michielsen, N. Chebrol, F. Jin, and H. De Raedt. “Simulation on and HPC simulation of quantum computers and quantum annealers”. *Advances in Parallel Computing* **34**, 101–119. 2019.  
DOI: 10.3233/APC190008.
- [Mic20] K. Michielsen. Interim report on HPC simulation of QC. OpenSuperQ — An Open Superconducting Quantum Computer (Grant 820363). 2020.  
DOI: 10.3030/820363.
- [Mil+21] D. Mills, S. Sivarajah, T. L. Scholten, and R. Duncan. “Application-Motivated, Holistic Benchmarking of a Full Quantum Computing Stack”. *Quantum* **5**, 415. 2021.  
DOI: 10.22331/q-2021-03-22-415.
- [MNM20] K. Mitarai, Y. O. Nakagawa, and W. Mizukami. “Theory of analytical energy derivatives for the variational quantum eigensolver”. *Phys. Rev. Res.* **2**, 013129. 2020.  
DOI: 10.1103/PhysRevResearch.2.013129.
- [Mor+21] L. Moro, M. G. A. Paris, M. Restelli, and E. Prati. “Quantum compiling by deep reinforcement learning”. *Communications Physics* **4**, 178. 2021.  
DOI: 10.1038/s42005-021-00684-3.
- [MS20] A. Montanaro and S. Stanisic. “Compressed variational quantum eigensolver for the Fermi-Hubbard model”. *arXiv:2006.01179*. 2020.  
DOI: 10.48550/ARXIV.2006.01179.

- [MYB19] S. McArdle, X. Yuan, and S. Benjamin. “Error-Mitigated Digital Quantum Simulation”. *Phys. Rev. Lett.* **122**, 180501. 2019.  
DOI: 10.1103/PhysRevLett.122.180501.
- [MZ93] F Mila and X Zotos. “Phase Diagram of the One-Dimensional Extended Hubbard Model at Quarter-Filling”. *Europhysics Letters (EPL)* **24**, 133–138. 1993.  
DOI: 10.1209/0295-5075/24/2/010.
- [MZO20] F. B. Maciejewski, Z. Zimborás, and M. Oszmaniec. “Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography”. *Quantum* **4**, 257. 2020.  
DOI: 10.22331/q-2020-04-24-257.
- [Nam+20] Y. Nam, J.-S. Chen, N. C. Pienti, K. Wright, C. Delaney, D. Maslov, K. R. Brown, S. Allen, J. M. Amini, J. Apisdorf, K. M. Beck, A. Blinov, V. Chaplin, M. Chmielewski, C. Collins, S. Debnath, K. M. Hudek, A. M. Ducore, M. Keesan, S. M. Kreike-meier, J. Mizrahi, P. Solomon, M. Williams, J. D. Wong-Campos, D. Moehring, C. Monroe, and J. Kim. “Ground-state energy estimation of the water molecule on a trapped-ion quantum computer”. *npj Quantum Information* **6**, 33. 2020.  
DOI: 10.1038/s41534-020-0259-3.
- [Nat+21] P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta. “Scalable Mitigation of Measurement Errors on Quantum Computers”. *PRX Quantum* **2**, 040326. 2021.  
DOI: 10.1103/PRXQuantum.2.040326.
- [NC10] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press. 2010.  
DOI: 10.1017/CBO9780511976667.
- [Nep20] R. I. Nepomechie. “Bethe ansatz on a quantum computer?” *arXiv:2010.01609*. 2020.  
DOI: 10.48550/ARXIV.2010.01609.
- [NFT20] K. M. Nakanishi, K. Fujii, and S. Todo. “Sequential minimal optimization for quantum-classical hybrid algorithms”. *Phys. Rev. Research* **2**, 043158. 2020.  
DOI: 10.1103/PhysRevResearch.2.043158.
- [NM19] National Academies of Sciences, Engineering and Medicine. *Quantum Computing: Progress and Prospects*. Ed. by E. Grumbling and M. Horowitz. Washington, DC: The National Academies Press. 2019.  
DOI: 10.17226/25196.
- [NMF19] K. M. Nakanishi, K. Mitarai, and K. Fujii. “Subspace-search variational quantum eigensolver for excited states”. *Phys. Rev. Research* **1**, 033062. 2019.  
DOI: 10.1103/PhysRevResearch.1.033062.
- [NW06] J. Nocedal and S. J. Wright. *Numerical Optimization*. 2e. New York, NY, USA: Springer. 2006.
- [O’B+19] T. E. O’Brien, B. Senjean, R. Sagastizabal, X. Bonet-Monroig, A. Dutkiewicz, F. Buda, L. DiCarlo, and L. Visscher. “Calculating energy derivatives for quantum chemistry on a quantum computer”. *npj Quantum Information* **5**, 1–12. 2019.  
DOI: 10.1038/s41534-019-0213-4.
- [OGB21] M. Ostaszewski, E. Grant, and M. Benedetti. “Structure optimization for parameterized quantum circuits”. *Quantum* **5**, 391. 2021.  
DOI: 10.22331/q-2021-01-28-391.

- [Ohz20] M. Ohzeki. "Breaking limitation of quantum annealer in solving optimization problems under constraints". *Scientific Reports* **10**, 1–12. 2020.  
DOI: 10.1038/s41598-020-60022-5.
- [O'M+16] P. J. J. O'Malley et al. "Scalable Quantum Simulation of Molecular Energies". *Phys. Rev. X* **6**, 031007. 2016.  
DOI: 10.1103/PhysRevX.6.031007.
- [Omi+22] K. Omiya, Y. O. Nakagawa, S. Koh, W. Mizukami, Q. Gao, and T. Kobayashi. "Analytical Energy Gradient for State-Averaged Orbital-Optimized Variational Quantum Eigensolvers and Its Application to a Photochemical Reaction". *Journal of Chemical Theory and Computation* **18**, 741–748. 2022.  
DOI: 10.1021/acs.jctc.1c00877.
- [Ost+21] M. Ostaszewski, L. M. Trenkwalder, W. Masarczyk, E. Scerri, and V. Dunjko. "Reinforcement learning for optimization of variational quantum circuit architectures". *arXiv:2103.16089*. 2021.  
DOI: 10.48550/ARXIV.2103.16089.
- [Pae+17] S. Paesani, A. A. Gentile, R. Santagati, J. Wang, N. Wiebe, D. P. Tew, J. L. O'Brien, and M. G. Thompson. "Experimental Bayesian Quantum Phase Estimation on a Silicon Photonic Chip". *Phys. Rev. Lett.* **118**, 100503. 2017.  
DOI: 10.1103/PhysRevLett.118.100503.
- [Par+17] R. M. Parrish, L. A. Burns, D. G. A. Smith, A. C. Simmonett, A. E. DePrince, E. G. Hohenstein, U. Bozkaya, A. Y. Sokolov, R. Di Remigio, R. M. Richard, J. F. Gonthier, A. M. James, H. R. McAlexander, A. Kumar, M. Saitow, X. Wang, B. P. Pritchard, P. Verma, H. F. Schaefer, K. Patkowski, R. A. King, E. F. Valeev, F. A. Evangelista, J. M. Turney, T. D. Crawford, and C. D. Sherrill. "Psi4 1.1: An Open-Source Electronic Structure Program Emphasizing Automation, Advanced Libraries, and Interoperability". *Journal of Chemical Theory and Computation* **13**, 3185–3197. 2017.  
DOI: 10.1021/acs.jctc.7b00174.
- [Pat+21] T. L. Patti, K. Najafi, X. Gao, and S. F. Yelin. "Entanglement devised barren plateau mitigation". *Phys. Rev. Research* **3**, 033090. 2021.  
DOI: 10.1103/PhysRevResearch.3.033090.
- [Per+14] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien. "A variational eigenvalue solver on a photonic quantum processor". *Nature Communications* **5**, 4213. 2014.  
DOI: 10.1038/ncomms5213.
- [PF10] J. B. Parkinson and D. J. Farnell. "Other approximate methods". *Lecture Notes in Physics* **816**, 99–108. 2010.  
DOI: 10.1007/978-3-642-13290-2\_9.
- [PF13] A. Paetzniak and A. G. Fowler. "Quantum circuit optimization by topological compaction in the surface code". *arXiv:1304.2807*. 2013.  
DOI: 10.48550/ARXIV.1304.2807.
- [PL13] K. L. Pudenz and D. A. Lidar. "Quantum adiabatic machine learning". *Quantum Information Processing* **12**, 2027–2070. 2013.  
DOI: 10.1007/s11128-012-0506-4.

- [PO+19] A. Perdomo-Ortiz, A. Feldman, A. Ozaeta, S. V. Isakov, Z. Zhu, B. O’Gorman, H. G. Katzgraber, A. Diedrich, H. Neven, J. de Kleer, B. Lackey, and R. Biswas. “Readiness of Quantum Optimization Machines for Industrial Applications”. *Phys. Rev. Applied* **12**, 014004. 2019.  
DOI: 10.1103/PhysRevApplied.12.014004.
- [Pop99] J. A. Pople. “Nobel Lecture: Quantum chemical models”. *Rev. Mod. Phys.* **71**, 1267–1274. 1999.  
DOI: 10.1103/RevModPhys.71.1267.
- [Pow09] M. J. D. Powell. The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives. Report DAMTP 2009/NA06, Centre for Mathematical Sciences, University of Cambridge. 2009.
- [Pow94] M. J. D. Powell. “A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation”. *Advances in Optimization and Numerical Analysis*. Dordrecht: Springer Netherlands, 51–67. 1994.  
DOI: 10.1007/978-94-015-8330-5\_4.
- [Pre+07] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical Recipes 3rd Edition: The Art of Scientific Computing. Cambridge University Press. 2007.
- [Pre18] J. Preskill. “Quantum Computing in the NISQ era and beyond”. *Quantum* **2**, 79. 2018.  
DOI: 10.22331/q-2018-08-06-79.
- [Pre21] J. Preskill. “Quantum computing 40 years later”. *arXiv:2106.10522*. 2021.  
DOI: 10.48550/ARXIV.2106.10522.
- [Pro+22] T. Proctor, K. Rudinger, K. Young, E. Nielsen, and R. Blume-Kohout. “Measuring the capabilities of quantum computers”. *Nature Physics* **18**, 75–79. 2022.  
DOI: 10.1038/s41567-021-01409-7.
- [PT19] J. F. Pérez-Torres. “Dilemma of the “Best Wavefunction”: Comparing Results of the STO-NG Procedure versus the Linear Variational Method”. *Journal of Chemical Education* **96**, 704–707. 2019.  
DOI: 10.1021/acs.jchemed.8b00959.
- [PWK22] H. Patil, Y. Wang, and P. S. Krstić. “Variational quantum linear solver with a dynamic ansatz”. *Phys. Rev. A* **105**, 012423. 2022.  
DOI: 10.1103/PhysRevA.105.012423.
- [Ran82] B. Randell, ed. The Origins of Digital Computers: Selected Papers. Springer Berlin Heidelberg. 1982.  
DOI: 10.1007/978-3-642-61812-3.
- [Rei+19] J.-M. Reiner, F. Wilhelm-Mauch, G. Schön, and M. Marthaler. “Finding the ground state of the Hubbard model by variational methods on a quantum computer with gate errors”. *Quantum Science and Technology* **4**, 035005. 2019.  
DOI: 10.1088/2058-9565/ab1e85.
- [RGM22] E. Rosenberg, P. Ginsparg, and P. L. McMahon. “Experimental error mitigation using linear rescaling for variational quantum eigensolving with up to 20 qubits”. *Quantum Science and Technology* **7**, 015024. 2022.  
DOI: 10.1088/2058-9565/ac3b37.

- [Ris+12] D. Ristè, C. C. Bultink, K. W. Lehnert, and L. DiCarlo. “Feedback Control of a Solid-State Qubit Using High-Fidelity Projective Measurement”. *Phys. Rev. Lett.* **109**, 240502. 2012.  
DOI: 10.1103/PhysRevLett.109.240502.
- [RM86] L. H. Reed and A. R. Murphy. “An investigation of the quality of approximate wave functions”. *Journal of Chemical Education* **63**, 757. 1986.  
DOI: 10.1021/ed063p757.
- [ROAG17] J. Romero, J. P. Olson, and A. Aspuru-Guzik. “Quantum autoencoders for efficient compression of quantum data”. *Quantum Science and Technology* **2**, 045001. 2017.  
DOI: 10.1088/2058-9565/aa8072.
- [RS10] J. Richter and J. Schulenburg. “The spin-1/2  $J_1$ - $J_2$  Heisenberg antiferromagnet on the square lattice: Exact diagonalization for  $N=40$  spins”. *The European Physical Journal B* **73**, 117–124. 2010.  
DOI: 10.1140/epjb/e2009-00400-4.
- [RS13] L. M. Rios and N. V. Sahinidis. “Derivative-free optimization: a review of algorithms and comparison of software implementations”. *Journal of Global Optimization* **56**, 1247–1293. 2013.  
DOI: 10.1007/s10898-012-9951-y.
- [RSR10] K. V. Rao., K. R. Sudha., and G. M. Rao. Pulse and digital circuits. Dorling Kinderley. 2010.
- [Run92a] K. J. Runge. “Finite-size study of the ground-state energy, susceptibility, and spin-wave velocity for the Heisenberg antiferromagnet”. *Phys. Rev. B* **45**, 12292–12296. 1992.  
DOI: 10.1103/PhysRevB.45.12292.
- [Run92b] K. J. Runge. “Quantum Monte Carlo calculation of the long-range order in the Heisenberg antiferromagnet”. *Phys. Rev. B* **45**, 7229–7236. 1992.  
DOI: 10.1103/PhysRevB.45.7229.
- [RWS12] S. Raeisi, N. Wiebe, and B. C. Sanders. “Quantum-circuit design for efficient simulations of many-body quantum dynamics”. *New Journal of Physics* **14**, 103017. 2012.  
DOI: 10.1088/1367-2630/14/10/103017.
- [Sag+19] R. Sagastizabal, X. Bonet-Monroig, M. Singh, M. A. Rol, C. C. Bultink, X. Fu, C. H. Price, V. P. Ostroukh, N. Muthusubramanian, A. Bruno, M. Beekman, N. Haider, T. E. O’Brien, and L. DiCarlo. “Experimental error mitigation via symmetry verification in a variational quantum eigensolver”. *Phys. Rev. A* **100**, 010302. 2019.  
DOI: 10.1103/PhysRevA.100.010302.
- [Sca16] R. T. Scalettar. “An Introduction to the Hubbard Hamiltonian”. *Quantum Materials: Experiments and Theory*. Ed. by E. Pavarini, E. Koch, J. van den Brink, and G. Sawatzky. Vol. 6. Schriften des Forschungszentrums Jülich. Reihe modeling and simulation. Jülich: Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag, 420 S. 2016.
- [Sch+17] A. Scherer, B. Valiron, S.-C. Mau, S. Alexander, E. van den Berg, and T. E. Chapuran. “Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2D target”. *Quantum Information Processing* **16**, 60. 2017.  
DOI: 10.1007/s11128-016-1495-5.

- [Sch+18] V. M. Schäfer, C. J. Ballance, K. Thirumalai, L. J. Stephenson, T. G. Ballance, A. M. Steane, and D. M. Lucas. “Fast quantum logic gates with trapped-ion qubits”. *Nature* **555**, 75–78. 2018.  
DOI: 10.1038/nature25737.
- [Sch+19] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran. “Evaluating analytic gradients on quantum hardware”. *Phys. Rev. A* **99**, 032331. 2019.  
DOI: 10.1103/PhysRevA.99.032331.
- [Sch+20] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe. “Circuit-centric quantum classifiers”. *Phys. Rev. A* **101**, 032308. 2020.  
DOI: 10.1103/PhysRevA.101.032308.
- [Sho94] P. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 124–134. 1994.  
DOI: 10.1109/SFCS.1994.365700.
- [Sho96] P. Shor. “Fault-tolerant quantum computation”. *Proceedings of 37th Conference on Foundations of Computer Science*, 56–65. 1996.  
DOI: 10.1109/SFCS.1996.548464.
- [Sho97] P. W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. *SIAM J. Comput.* **26**, 14841509. 1997.  
DOI: 10.1137/S0097539795293172.
- [Sho99] P. W. Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”. *SIAM Review* **41**, 303–332. 1999.  
DOI: 10.1137/S0036144598347011.
- [Sko+21] A. Skolik, J. R. McClean, M. Mohseni, P. van der Smagt, and M. Leib. “Layerwise learning for quantum neural networks”. *Quantum Machine Intelligence* **3**, 5. 2021.  
DOI: 10.1007/s42484-020-00036-4.
- [Som11] T. Sommerfeld. “Lorentz Trial Function for the Hydrogen Atom: A Simple, Elegant Exercise”. *Journal of Chemical Education* **88**, 1521–1524. 2011.  
DOI: 10.1021/ed200040e.
- [Son+19] C. Song, J. Cui, H. Wang, J. Hao, H. Feng, and Y. Li. “Quantum computation with universal error mitigation on a superconducting quantum processor”. *Science Advances* **5**, eaaw5686. 2019.  
DOI: 10.1126/sciadv.aaw5686.
- [SPK13] R. D. da Silva, E. Pius, and E. Kashefi. “Global Quantum Circuit Optimization”. *arXiv:1301.0351*. 2013.  
DOI: 10.48550/ARXIV.1301.0351.
- [SRL12] J. T. Seeley, M. J. Richard, and P. J. Love. “The Bravyi-Kitaev transformation for quantum computation of electronic structure”. *The Journal of Chemical Physics* **137**, 224109. 2012.  
DOI: 10.1063/1.4768229.
- [SSY20] K. Seki, T. Shirakawa, and S. Yunoki. “Symmetry-adapted variational quantum eigensolver”. *Phys. Rev. A* **101**, 052340. 2020.  
DOI: 10.1103/PhysRevA.101.052340.
- [Sto+20] J. Stokes, J. Izaac, N. Killoran, and G. Carleo. “Quantum Natural Gradient”. *Quantum* **4**, 269. 2020.  
DOI: 10.22331/q-2020-05-25-269.

- [Sto47] E. C. Stoner. "Ferromagnetism". *Reports on Progress in Physics* **11**, 43–112. 1947.  
DOI: 10.1088/0034-4885/11/1/304.
- [Sun+18] Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K.-L. Chan. "PySCF: the Python-based simulations of chemistry framework". *WIREs Computational Molecular Science* **8**, e1340. 2018.  
DOI: 10.1002/wcms.1340.
- [Sut85] B. Sutherland. "An introduction to the Bethe ansatz". *Exactly Solvable Problems in Condensed Matter and Relativistic Field Theory*. Ed. by B. S. Shastry, S. S. Jha, and V. Singh. Berlin, Heidelberg: Springer Berlin Heidelberg, 1–95. 1985.
- [SV00] G. L. G. Sleijpen and H. A. V. der Vorst. "A Jacobi-Davidson Iteration Method for Linear Eigenvalue Problems". *SIAM Review* **42**, 267–293. 2000.
- [SW18] K. Setia and J. D. Whitfield. "Bravyi-Kitaev Superfast simulation of electronic structure on a quantum computer". *The Journal of Chemical Physics* **148**, 164104. 2018.  
DOI: 10.1063/1.5019371.
- [SW95] T. Sleator and H. Weinfurter. "Realizable Universal Quantum Logic Gates". *Phys. Rev. Lett.* **74**, 4087–4090. 1995.  
DOI: 10.1103/PhysRevLett.74.4087.
- [Swe+20] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Faehrmann, B. Meynard-Piganeau, and J. Eisert. "Stochastic gradient descent for hybrid quantum-classical optimization". *Quantum* **4**, 314. 2020.  
DOI: 10.22331/q-2020-08-31-314.
- [SY22] K. Seki and S. Yunoki. "Spatial, spin, and charge symmetry projections for a Fermi-Hubbard model on a quantum computer". *Phys. Rev. A* **105**, 032419. 2022.  
DOI: 10.1103/PhysRevA.105.032419.
- [Tak+17] M. Takita, A. W. Cross, A. D. Córcoles, J. M. Chow, and J. M. Gambetta. "Experimental Demonstration of Fault-Tolerant State Preparation with Superconducting Qubits". *Phys. Rev. Lett.* **119**, 180501. 2017.  
DOI: 10.1103/PhysRevLett.119.180501.
- [Tan+21] H. L. Tang, V. Shkolnikov, G. S. Barron, H. R. Grimsley, N. J. Mayhall, E. Barnes, and S. E. Economou. "Qubit-ADAPT-VQE: An Adaptive Algorithm for Constructing Hardware-Efficient Ansätze on a Quantum Processor". *PRX Quantum* **2**, 020–310. 2021.  
DOI: 10.1103/PRXQuantum.2.020310.
- [TBG17] K. Temme, S. Bravyi, and J. M. Gambetta. "Error Mitigation for Short-Depth Quantum Circuits". *Phys. Rev. Lett.* **119**, 180509. 2017.  
DOI: 10.1103/PhysRevLett.119.180509.
- [Ter00] B. M. Terhal. "Bell inequalities and the separability criterion". *Physics Letters A* **271**, 319–326. 2000.  
DOI: 10.1016/S0375-9601(00)00401-1.
- [TGH22] S. Tornow, W. Gehrke, and U. Helmbrecht. "Non-equilibrium dynamics of a dissipative two-site Hubbard model simulated on IBM quantum computers". *Journal of Physics A: Mathematical and Theoretical* **55**, 245302. 2022.  
DOI: 10.1088/1751-8121/ac6bd0.

- [Tor+20] G. Torlai, G. Mazzola, G. Carleo, and A. Mezzacapo. “Precise measurement of quantum observables with neural-network estimators”. *Phys. Rev. Research* **2**, 022–060. 2020.  
DOI: 10.1103/PhysRevResearch.2.022060.
- [Tra+18] A. Tranter, P. J. Love, F. Mintert, and P. V. Coveney. “A Comparison of the Bravyi-Kitaev and Jordan-Wigner Transformations for the Quantum Simulation of Quantum Chemistry”. *Journal of Chemical Theory and Computation* **14**, 5617–5630. 2018.  
DOI: 10.1021/acs.jctc.8b00450.
- [Tra+19] A. Tranter, P. J. Love, F. Mintert, N. Wiebe, and P. V. Coveney. “Ordering of Trotterization: Impact on Errors in Quantum Simulation of Electronic Structure”. *Entropy* **21**, 1218. 2019.  
DOI: 10.3390/e21121218.
- [Tro+12] S. Trotzky, Y.-A. Chen, A. Flesch, I. P. McCulloch, U. Schollwöck, J. Eisert, and I. Bloch. “Probing the relaxation towards equilibrium in an isolated strongly correlated one-dimensional Bose gas”. *Nature Physics* **8**, 325–330. 2012.  
DOI: 10.1038/nphys2232.
- [Tur37] A. M. Turing. “On computable numbers, with an application to the entscheidungsproblem”. *Proceedings of the London Mathematical Society* **s2-42**, 230–265. 1937.  
DOI: 10.1112/plms/s2-42.1.230.
- [VBG55] J. Von Neumann, R. T. Beyer, and H. H. Goldstine. *Mathematical foundations of quantum mechanics*. Princeton University Press. 1955.
- [VC05] F. Verstraete and J. I. Cirac. “Mapping local Hamiltonians of fermions to local Hamiltonians of spins”. *Journal of Statistical Mechanics*, P09012. 2005.  
DOI: 10.1088/1742-5468/2005/09/p09012.
- [VCL09] F. Verstraete, J. I. Cirac, and J. I. Latorre. “Quantum circuits for strongly correlated quantum systems”. *Phys. Rev. A* **79**, 032316. 2009.  
DOI: 10.1103/PhysRevA.79.032316.
- [VCM13] M. Valko, A. Carpentier, and R. Munos. “Stochastic Simultaneous Optimistic Optimization”. *Proceedings of the 30th International Conference on Machine Learning*. Ed. by S. Dasgupta and D. McAllester. Vol. 28. Proceedings of Machine Learning Research. Atlanta, Georgia, USA: PMLR, 19–27. 2013.
- [VD+21] J. S. Van Dyke, G. S. Barron, N. J. Mayhall, E. Barnes, and S. E. Economou. “Preparing Bethe Ansatz Eigenstates on a Quantum Computer”. *PRX Quantum* **2**, 040329. 2021.  
DOI: 10.1103/PRXQuantum.2.040329.
- [VDN94] L. Visscher, H. De Raedt, and W. Nieuwpoort. “A new configuration selection method for configuration interaction calculations”. *Chemical Physics Letters* **227**, 327–336. 1994.  
DOI: 10.1016/0009-2614(94)00824-8.
- [Ver96] G. L. Verschuur. *Hidden Attraction: The History and Mystery of Magnetism*. Oxford University Press. 1996.
- [VG96] H. van der Vorst and G. Golub. “150 Years old and still alive: eigenproblems”. *The State of the Art in Numerical Analysis*. Ed. by I. S. Duff and G. A. Watson. Oxford University Press. 1996.

- [Vij+12] R. Vijay, C. Macklin, D. H. Slichter, S. J. Weber, K. W. Murch, R. Naik, A. N. Korotkov, and I. Siddiqi. “Stabilizing Rabi oscillations in a superconducting qubit using quantum feedback”. *Nature* **490**, 77–80. 2012.  
DOI: 10.1038/nature11505.
- [VOZ69] R. Velasco, C. Ottinger, and R. N. Zare. “Dissociation Energy of  $\text{Li}_2$  from Laser-Excited Fluorescence”. *The Journal of Chemical Physics* **51**, 5522–5532. 1969.  
DOI: 10.1063/1.1671979.
- [VPB18] G. Verdon, J. Pye, and M. Broughton. “A Universal Training Algorithm for Quantum Deep Learning”. *arXiv:1806.09729*. 2018.  
DOI: 10.48550/ARXIV.1806.09729.
- [VT18] J. G. Vidal and D. O. Theis. “Calculus on parameterized quantum circuits”. *arXiv:1812.06323*. 2018.  
DOI: 10.48550/ARXIV.1812.06323.
- [VYI20] V. Verteletskiy, T.-C. Yen, and A. F. Izmaylov. “Measurement optimization in the variational quantum eigensolver using a minimum clique cover”. *The Journal of Chemical Physics* **152**, 124114. 2020.  
DOI: 10.1063/1.5141458.
- [WBAG11] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik. “Simulation of electronic structure Hamiltonians using quantum computers”. *Molecular Physics* **109**, 735–750. 2011.  
DOI: 10.1080/00268976.2011.552441.
- [Wei+20] P. Weinberg, M. Tylutki, J. M. Rönkkö, J. Westerholm, J. A. Åström, P. Manninen, P. Törmä, and A. W. Sandvik. “Scaling and Diabatic Effects in Quantum Annealing with a D-Wave Device”. *Phys. Rev. Lett.* **124**, 090502. 2020.  
DOI: 10.1103/PhysRevLett.124.090502.
- [Wes+17] S. Wessel, B. Normand, F. Mila, and A. Honecker. “Efficient Quantum Monte Carlo simulations of highly frustrated magnets: the frustrated spin-1/2 ladder”. *SciPost Phys.* **3**, 005. 2017.  
DOI: 10.21468/SciPostPhys.3.1.005.
- [WF08] A. Weiße and H. Fehske. “Exact Diagonalization Techniques”. *Computational Many-Particle Physics*. Ed. by H. Fehske, R. Schneider, and A. Weiße. Berlin, Heidelberg: Springer Berlin Heidelberg, 529–544. 2008.  
DOI: 10.1007/978-3-540-74686-7\_18.
- [WGG20] D. Wierichs, C. Gogolin, and M. Kastoryano. “Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer”. *Phys. Rev. Research* **2**, 043246. 2020.  
DOI: 10.1103/PhysRevResearch.2.043246.
- [WHB19] D. Wang, O. Higgott, and S. Brierley. “Accelerated Variational Quantum Eigensolver”. *Phys. Rev. Lett.* **122**, 140504. 2019.  
DOI: 10.1103/PhysRevLett.122.140504.
- [Whi92] S. R. White. “Density matrix formulation for quantum renormalization groups”. *Phys. Rev. Lett.* **69**, 2863–2866. 1992.  
DOI: 10.1103/PhysRevLett.69.2863.
- [WHT15] D. Wecker, M. B. Hastings, and M. Troyer. “Progress towards practical quantum variational algorithms”. *Phys. Rev. A* **92**, 042303. 2015.  
DOI: 10.1103/PhysRevA.92.042303.

- [Wie14] C.-R. Wie. “Bloch sphere model for two-qubit pure states”. *arXiv:1403.8069*. 2014. DOI: 10.48550/ARXIV.1403.8069.
- [Wie+20] R. Wiersema, C. Zhou, Y. de Sereville, J. F. Carrasquilla, Y. B. Kim, and H. Yuen. “Exploring Entanglement and Optimization within the Hamiltonian Variational Ansatz”. *PRX Quantum* **1**, 020319. 2020. DOI: 10.1103/PRXQuantum.1.020319.
- [Wil+20] D. Willsch, M. Willsch, H. De Raedt, and K. Michielsen. “Support vector machines on the D-Wave quantum annealer”. *Computer Physics Communications* **248**, 107006. 2020. DOI: 10.1016/j.cpc.2019.107006.
- [Wil20] D. Willsch. “Supercomputer simulations of transmon quantum computers”. PhD thesis. RWTH Aachen. 2020.
- [Wil+20a] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen. “Benchmarking the quantum approximate optimization algorithm”. *Quantum Information Processing* **19**, 197. 2020. DOI: 10.1007/s11128-020-02692-8.
- [Wil+20b] M. Willsch, D. Willsch, K. Michielsen, and H. De Raedt. “Discrete-Event Simulation of Quantum Walks”. *Frontiers in Physics* **8**, 145. 2020. DOI: 10.3389/fphy.2020.00145.
- [Woh53] E. P. Wohlfarth. “The theoretical and experimental status of the collective electron theory of ferromagnetism”. *Reviews of Modern Physics* **25**, 211–219. 1953. DOI: 10.1103/RevModPhys.25.211.
- [Wol69] P. Wolfe. “Convergence Conditions for Ascent Methods”. *SIAM Review* **11**, 226–235. 1969. DOI: 10.1137/1011036.
- [Wu+20] X.-C. Wu, M. G. Davis, F. T. Chong, and C. Iancu. “QGo: Scalable Quantum Circuit Optimization Using Automated Synthesis”. *arXiv:2012.09835*. 2020. DOI: 10.48550/ARXIV.2012.09835.
- [Yao+21] Y.-X. Yao, N. Gomes, F. Zhang, C.-Z. Wang, K.-M. Ho, T. Iadecola, and P. P. Orth. “Adaptive Variational Quantum Dynamics Simulations”. *PRX Quantum* **2**, 030307. 2021. DOI: 10.1103/PRXQuantum.2.030307.
- [Yor+21] Y. S. Yordanov, V. Armaos, C. H. W. Barnes, and D. R. M. Arvidsson-Shukur. “Qubit-excitation-based adaptive variational quantum eigensolver”. *Communications Physics* **4**, 228. 2021. DOI: 10.1038/s42005-021-00730-0.
- [You01] D. C. Young. *Computational Chemistry: A Practical Guide for Applying Techniques to Real-World Problems*. John Wiley & Sons. 2001.
- [Zen+21] J. Zeng, Z. Wu, C. Cao, C. Zhang, S.-Y. Hou, P. Xu, and B. Zeng. “Simulating noisy variational quantum eigensolver with local noise models”. *Quantum Engineering* **3**, e77. 2021. DOI: 10.1002/que2.77.
- [ZG21] C. Zhao and X.-S. Gao. “Analyzing the barren plateau phenomenon in training quantum neural networks with the ZX-calculus”. *Quantum* **5**, 466. 2021. DOI: 10.22331/q-2021-06-04-466.

- [Zha+20a] S. Zhang, Y. Lu, K. Zhang, W. Chen, Y. Li, J.-N. Zhang, and K. Kim. “Error-mitigated quantum gates exceeding physical fidelities in a trapped-ion system”. *Nature Communications* **11**, 587. 2020.  
DOI: 10.1038/s41467-020-14376-z.
- [Zha+20b] A. Zhao, A. Tranter, W. M. Kirby, S. F. Ung, A. Miyake, and P. J. Love. “Measurement reduction in variational quantum algorithms”. *Phys. Rev. A* **101**, 062322. 2020.  
DOI: 10.1103/PhysRevA.101.062322.
- [Zha+21] Z.-J. Zhang, T. H. Kyaw, J. S. Kottmann, M. Degroote, and A. Aspuru-Guzik. “Mutual information-assisted adaptive variational quantum eigensolver”. *Quantum Science and Technology* **6**, 035001. 2021.  
DOI: 10.1088/2058-9565/abdca4.
- [Zhu+19] D. Zhu, N. M. Linke, M. Benedetti, K. A. Landsman, N. H. Nguyen, C. H. Alderete, A. Perdomo-Ortiz, N. Korda, A. Garfoot, C. Brecque, L. Egan, O. Perdomo, and C. Monroe. “Training of quantum circuits on a hybrid quantum computer”. *Science Advances* **5**, eaaw9918. 2019.  
DOI: 10.1126/sciadv.aaw9918.
- [Zhu+22] Q. Zhu et al. “Quantum computational advantage via 60-qubit 24-cycle random circuit sampling”. *Science Bulletin* **67**, 240–245. 2022.  
DOI: 10.1016/j.scib.2021.10.017.
- [Zus36] K. Zuse. “Method for Automatic Execution of Calculations with the Aid of Computers”. *The Origins of Digital Computers: Selected Papers*. Ed. by B. Randell. Springer Berlin Heidelberg, 163–170. 1936.  
DOI: 10.1007/978-3-642-61812-3\_12.



# List of publications

1. M. S. Jattana, F. Jin, H. De Raedt, and K. Michielsen. “Assessment of the Variational Quantum Eigensolver: Application to the Heisenberg Model”. *Frontiers in Physics* **10**, 907160. 2022. DOI: 10.3389/fphy.2022.907160
2. M. S. Jattana, H. Lagemann, and K. Michielsen. Final report on HPC simulation of QC: Conclusions for quantum computing without error correction. OpenSuperQ — An Open Superconducting Quantum Computer (Grant 820363). 2022. DOI: 10.3030/820363
3. M. S. Jattana, F. Jin, H. De Raedt, and K. Michielsen. “Improved variational quantum eigensolver via quasi-dynamical evolution”. *arXiv:2202.10130*. 2022. DOI: 10.48550/ARXIV.2202.10130
4. H. De Raedt, M. S. Jattana, D. Willsch, M. Willsch, F. Jin, and K. Michielsen. “Discrete-Event Simulation of an Extended Einstein-Podolsky-Rosen-Bohm Experiment”. *Frontiers in Physics* **8**, 160. 2020. DOI: 10.3389/fphy.2020.00160
5. M. S. Jattana, F. Jin, H. De Raedt, and K. Michielsen. “General error mitigation for quantum circuits”. *Quantum Information Processing* **19**, 414. 2020. DOI: 10.1007/s11128-020-02913-0
6. M. S. Jattana and K. Michielsen. “Quantencomputer - Steht die Revolution vor der Türe?” *Technik in Bayern* **06/2019**, 18–19. 2019



# Eidesstattliche Erklärung

Ich, Manpreet Singh Jattana, erkläre hiermit, dass diese Dissertation und die darin dargelegten Inhalte die eigenen sind und selbstständig, als Ergebnis der eigenen originären Forschung, generiert wurden.

Hiermit erkläre ich an Eides statt

- Diese Arbeit wurde vollständig oder größtenteils in der Phase als Doktorand dieser Fakultät und Universität angefertigt;
- Sofern irgendein Bestandteil dieser Dissertation zuvor für einen akademischen Abschluss oder eine andere Qualifikation an dieser oder einer anderen Institution verwendet wurde, wurde dies klar angezeigt;
- Wenn immer andere eigene- oder Veröffentlichungen Dritter herangezogen wurden, wurden diese klar benannt;
- Wenn aus anderen eigenen- oder Veröffentlichungen Dritter zitiert wurde, wurde stets die Quelle hierfür angegeben. Diese Dissertation ist vollständig meine eigene Arbeit, mit der Ausnahme solcher Zitate;
- Alle wesentlichen Quellen von Unterstützung wurden benannt;
- Wenn immer ein Teil dieser Dissertation auf der Zusammenarbeit mit anderen basiert, wurde von mir klar gekennzeichnet, was von anderen und was von mir selbst erarbeitet wurde;
- Teile dieser Arbeit wurden zuvor veröffentlicht und zwar in:
  - M. S. Jattana, F. Jin, H. De Raedt, and K. Michielsen. "Assessment of the Variational Quantum Eigensolver: Application to the Heisenberg Model". *Frontiers in Physics* **10**, 907160. 2022. DOI: 10.3389/fphy.2022.907160
  - M. S. Jattana, F. Jin, H. De Raedt, and K. Michielsen. "General error mitigation for quantum circuits". *Quantum Information Processing* **19**, 414. 2020. DOI: 10.1007/s11128-020-02913-0
  - M. S. Jattana, H. Lagemann, and K. Michielsen. Final report on HPC simulation of QC: Conclusions for quantum computing without error correction. OpenSuperQ — An Open Superconducting Quantum Computer (Grant 820363). 2022. DOI: 10.3030/820363
  - M. S. Jattana, F. Jin, H. De Raedt, and K. Michielsen. "Improved variational quantum eigensolver via quasi-dynamical evolution". *arXiv:2202.10130*. 2022. DOI: 10.48550/ARXIV.2202.10130

Unterschrift:

---

Datum:

---



# Acknowledgements

The greatest appreciations are due to Kristel Michielsen. Thanks for giving me the opportunity and advice to work on this new topic after my Masters. Thanks for writing several recommendation letters, signing documents, and updating and taking care of a lot of administrative stuff over the years. I have greatly enjoyed the supervision and flexibility of working in your group. I am very thankful to you for helping facilitate my career prospects.

Special thanks to Hans De Raedt. Thanks for helping me better formulate some of the equations and text over the years. Thanks for your spontaneous skype calls to answer some small questions which have led to hour-long discussions of other things often more important than my original question. You are always welcome to continue asking for 'reading resources'.

Gratefulness is due to Fengping Jin. You have guided me for the 4+ years through my Masters and PhD at every stage. Thanks for always pointing out the small details yet keeping the big picture in mind. I am really thankful for your availability and discussions. I am happy to observe that you are not unbeatable at table tennis, with some rare evidence of me winning by a few points. I hope we can continue some of the work we discussed but were not finished in this thesis.

Thanks to David DiVincenzo for the second supervision of the thesis and for being on the doctoral committee.

Thank you Dennis Willsch for proofreading the introductory chapters. Your active participation in the lunch questions helps me come up with new ones! As per tradition, there are (again) some unexplained dips in here somewhere. I am glad we could occasionally do BITB and play volleyball together despite corona. Thanks to Madita Willsch for helpful discussions and for co-tutoring the computational physics course. Thanks to both of you for helpful advice during the last few days with the administrative hurdles of finishing this all up.

Many thanks to Thomas Lippert for the special permission granted during the pandemic to undertake the project in Canada. Thank you also for occasionally dropping by in our group meetings or at lunch at SeeCasino for a discussion. I gratefully appreciate you offering me the position in Frankfurt and the future possibilities and freedoms associated with it.

Thanks to Roman Krems and his team for hosting me at the University of British Columbia to carry out the project "Accelerated quantum variational methods using Bayesian optimization". The secret Santa game was a delightful highlight. I also thank Mitacs Canada and my institute for providing funding for the visit.

Thanks to Göran Wendin and the whole team at Chalmers University of Technology for providing the Python code for quantum chemistry problems and providing help with the Li molecule simulation.

I acknowledge support from the project OpenSuperQ (820363) of the EU Quantum Flagship. I acknowledge support from the project Jülich UNified Infrastructure for Quantum computing (JUNIQU) that has received funding from the German Federal Ministry of Education and Research (BMBF) and the Ministry of Culture and Science of the State of North Rhine-Westphalia. I gratefully acknowledge the Gauss Centre for Supercomputing e.V. ([www.gauss-centre.eu](http://www.gauss-centre.eu)) for funding this project by providing computing time at Jülich Supercomputing Centre. I acknowledge the use of IBM Quantum services for this work. The views expressed

are my own and do not reflect the official policy or position of IBM or the IBM Quantum team.

Thank you Carlos Gonzalez Calaza for your help with Qiskit and Python in general in the early stages of my PhD. Some early tips have helped me avoid programming problems. Thank you Vrinda Mehta for our occasional Hindi conversations. Thank you Hannes Lagemann for being a parallel PhD partner with all kinds of administrative stuff, as well as being a translator for the Abstract. Thanks to all the new members of the group for a nice work atmosphere and great discussions. I am hoping you all will invite me for the next year's football tournament! and the JSC Table Tennis tournament! Thanks to everyone at JSC for their organisation and attendance of CaDS seminars where I have always happily presented a thing or two.

I would like to thank my parents and rest of the family for their encouragement and support throughout my PhD even from abroad. Finally, I would like to thank Moritz, Danitza, Meret, Rebekka, Harsh, Torsten and Gabi. Thank you Aachen; goodbye for now!

Band / Volume 43

**Algorithms for massively parallel generic *hp*-adaptive finite element methods**

M. Fehling (2020), vii, 78 pp

ISBN: 978-3-95806-486-7

URN: urn:nbn:de:0001-2020071402

Band / Volume 44

**The method of fundamental solutions for computing interior transmission eigenvalues**

L. Pieronek (2020), 115 pp

ISBN: 978-3-95806-504-8

Band / Volume 45

**Supercomputer simulations of transmon quantum computers**

D. Willsch (2020), IX, 237 pp

ISBN: 978-3-95806-505-5

Band / Volume 46

**The Influence of Individual Characteristics on Crowd Dynamics**

P. Georg (2021), xiv, 212 pp

ISBN: 978-3-95806-561-1

Band / Volume 47

**Structural plasticity as a connectivity generation and optimization algorithm in neural networks**

S. Diaz Pier (2021), 167 pp

ISBN: 978-3-95806-577-2

Band / Volume 48

**Porting applications to a Modular Supercomputer**

Experiences from the DEEP-EST project

A. Kreuzer, E. Suarez, N. Eicker, Th. Lippert (Eds.) (2021), 209 pp

ISBN: 978-3-95806-590-1

Band / Volume 49

**Operational Navigation of Agents and Self-organization Phenomena in Velocity-based Models for Pedestrian Dynamics**

Q. Xu (2022), xii, 112 pp

ISBN: 978-3-95806-620-5

Band / Volume 50

**Utilizing Inertial Sensors as an Extension of a Camera Tracking System for Gathering Movement Data in Dense Crowds**

J. Schumann (2022), xii, 155 pp

ISBN: 978-3-95806-624-3

Band / Volume 51

**Final report of the DeepRain project  
Abschlußbericht des DeepRain Projektes**

(2022), ca. 70 pp

ISBN: 978-3-95806-675-5

Band / Volume 52

**JSC Guest Student Programme Proceedings 2021**

I. Kabadshow (Ed.) (2023), ii, 82 pp

ISBN: 978-3-95806-684-7

Band / Volume 53

**Applications of variational methods for quantum computers**

M. S. Jattana (2023), vii, 160 pp

ISBN: 978-3-95806-700-4

Weitere **Schriften des Verlags im Forschungszentrum Jülich** unter  
<http://wwwzb1.fz-juelich.de/verlagextern1/index.asp>



IAS Series  
Band / Volume 53  
ISBN 978-3-95806-700-4